

**2022-2023 Academic Year**

**Object Oriented Programming Lesson**

**2. Midterm Exam Project**

**HOSPITAL INFORMATION SYSTEM**

**Project Members**

- **Hilal ÖZSÖZ -1030521037**
- **Gizem GÜRSOY -1030521041**
- **Elif İrem KESKİN -1030510558**
- **Berfin ÇAKMAKTEPE -1030510090**

## **Application Domain**

The hospital information system is a software that allows us to perform hospital-related transactions through the application. It provides access to the polyclinics and doctors in the hospital, making an appointment with the desired doctor, and assigning the examinations they want in a very short time. It also includes tasks such as assigning laboratory results to these examinations and prescribing drugs for the diagnosis made as a result of the findings. All these ensure that the workload of each staff working in the hospital is reduced and their performance is maximized.

The hospital information system saves time, as it allows many transactions to be made after registration through the application. It is also very useful for the patient to be able to access the diagnosis made and the prescription given through the application. It also allows the patient to save on the road. It is an application aimed at reducing the workload of doctors.

Our application was needed in order to carry out the hospital works regularly and quickly. Today, since every sector is digitalized, this type of system was also needed in the hospital sector. In cases where this system is not used, the organization and functioning of the works will slow down considerably and the productivity will decrease.

### **Importance of hospital information system:**

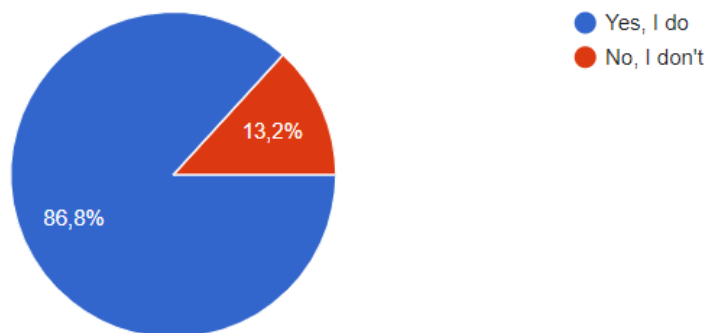
- Hospital information system is the most efficient and easiest application that can be used for a hospital.
- The information given during registration is stored in the most secure way.
- With this system, the physical workload will be reduced, and the whole process will be only through the application.
- Anyone who wants to be examined will be able to get an appointment date and be examined only by registering with their information.
- It will save a lot of time as the whole process will proceed in just a few clicks.

## PRE-APPLICATION SURVEY

The questionnaire was prepared to receive suggestions, feedback and thoughts of those who use the hospital information system. The questionnaire was used to receive pre-project requests and make post-project improvements.

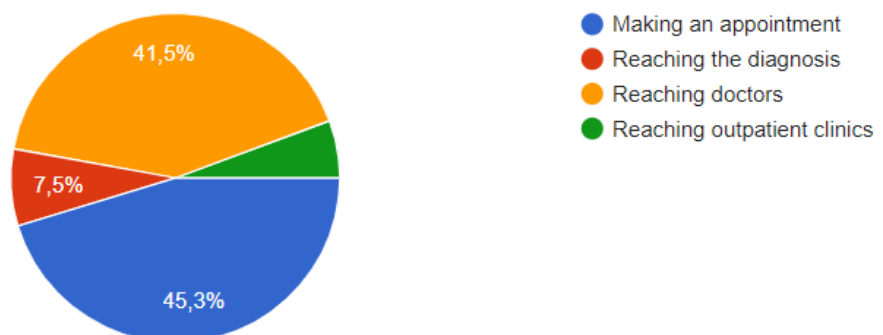
Do you want to make doctor appointments from an app without going to the hospital?

53 yanıt



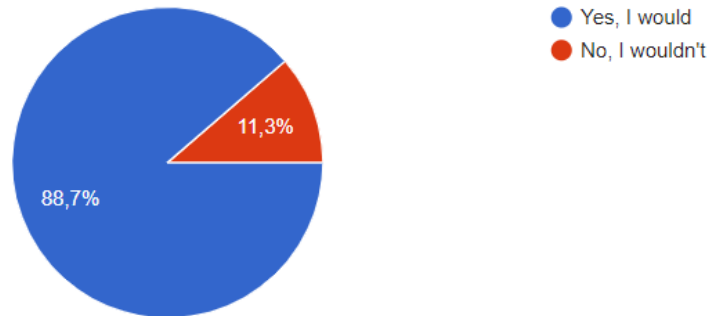
Which phase would you most like to have in the application?

53 yanıt



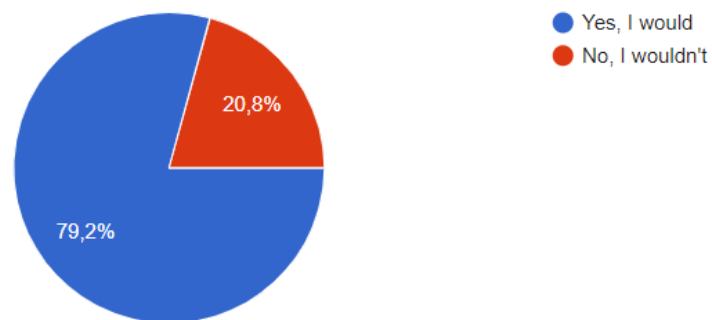
Would you like the hospital to send an information message to your phone number?

53 yanıt



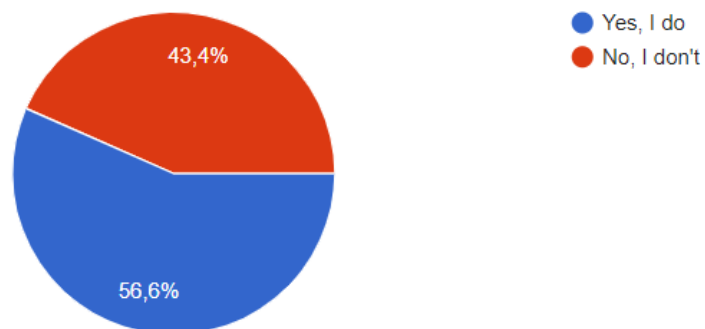
Would you like to reach the diagnosis and the prescription written for you without going to the hospital?

53 yanıt



Do you have any doubts about giving your personal information to the hospital information system?

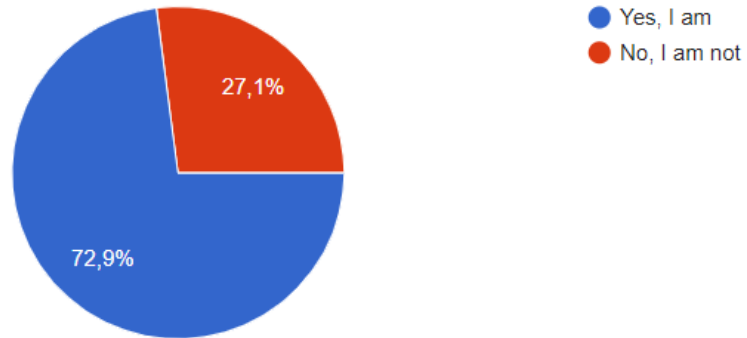
53 yanıt



# HOSPITAL INFORMATION SYSTEM EVALUATION SURVEY RESULT

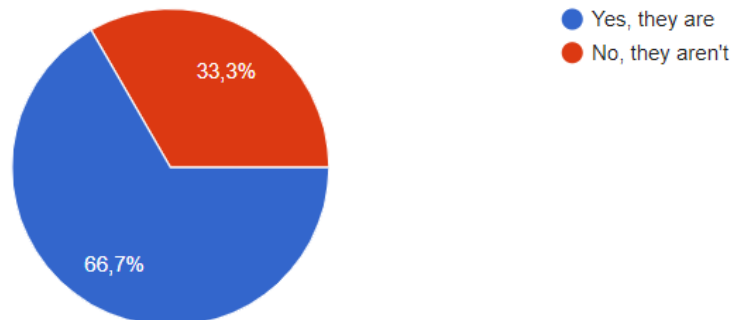
Are you satisfied with the appointment system in the hospital?

48 yanıt



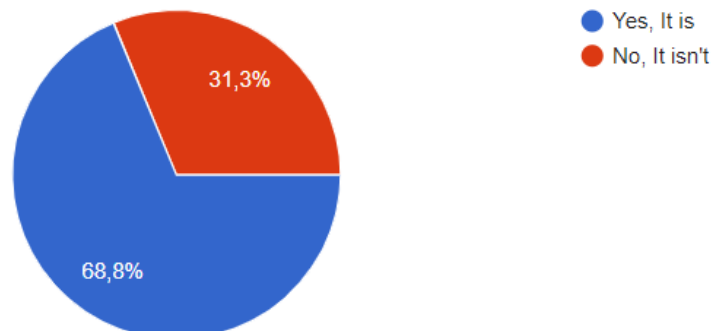
Are the stages of registering in the hospital system easy for you?

48 yanıt



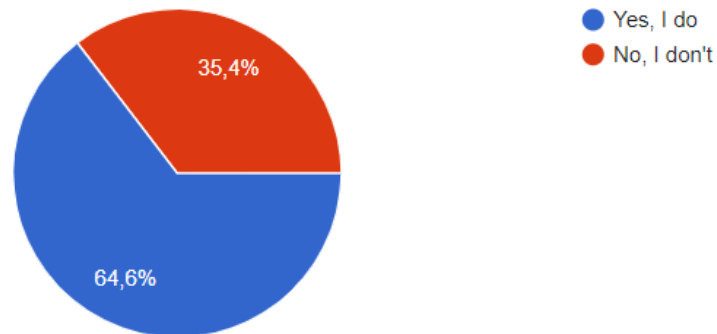
Is the functioning of the hospital system clear enough?

48 yanıt



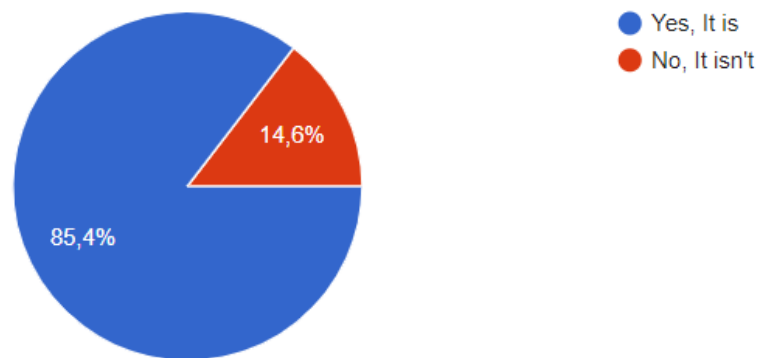
Do you have difficulty in reaching the diagnosis of your disease and the results of the examination through the application?

48 yanıt



Is it a convenience for you to send an information message to your phone number from the hospital?

48 yanıt



# GLOSSARY OF HOSPITAL INFORMATION SYSTEM

The glossary is used within the project to provide a better understanding of the terms specific to the project by the customer and other engineers. Thanks to the glossary, the project becomes more understandable and presentable. Each project is specially extracted according to its content.

**Actor:**

Actors are people who use the system. Actors of the hospital information system; doctor, patient, laboratory manage.

**Appointment:**

A promise the patient has set from the hospital to be examined at a certain time of the day. The patient receives the appropriate appointment according to her disease.

**Appointment Cancellation:**

It is the process that is done when the appointment cannot be made or when the wrong appointment is selected.

**Confirmation Code:**

The string of words sent to register the phone number when registering in the system. It is a 6 digit syntax. The code consists of the first letters of the patient's name and surname and the last four digits of the ID number.

**Control:**

It is a test of whether a situation-event is appropriate, operability, plausibility and logic.

**Diagnosis :**

It is the name given to the doctor's detection of the disease in her patient. This is the stage before the prescription is written.

**Doctor:**

A person who has medical training to prevent and treat diseases. Doctors perform tasks such as examining patients, performing surgery, monitoring treatment, and prescribing. It is the another main character of the system like patient.

**Examination:**

The process of investigating whether someone is sick or where their illness is, carried out by a physician. It is the first condition of disease detection.

**Examination Result:**

Output containing the diseases revealed as a result of the patient's examination.

**Hospital :**

It is the place where the person is examined and treated by doctors or nurses under the name of the patient when there is a problem-deficiency-wound in her body, organs, mind or spirit. In our system, it is the place where the patient makes an appointment, is examined and has tests done.

**Hospital Information System :**

The hospital information system is the platform where the patient is examined by making an appointment at the desired date and time, at the end of the examination, the doctor has the tests he deems necessary, the results of the examinations are recorded, the doctor writes a prescription and records it

**Hospital Number:**

It is a string of integers determined by the admin as the last 6 digits of the ID number to use when logging in for the doctor and lab manager

**Lab:**

A place with tools and equipment for scientific and technical research. Some blood, secretion, etc., which the doctor deems necessary according to the patient's disease. place of inspection.

**Lab Manager:**

Lab manager is the person who performs the examinations in the laboratory and enters the results.

**Message:**

It is a form of communication between the user and the system as a result of a situation.

**Outpatient:**

Another name for the polyclinic.

**Patient:**

She/He is a person who feels a disorder in her/him health, in her/him organs, in her/him mind, in her/him soul. It is the main character of the system.



**User Password :**

Password is a word string consisting of the first letter of the patient's name and surname and the last four digits of the ID number when registering to the system. The patient has to use this password when entering the system to make an appointment.

**Polyclinic:**

It's where various diseases are treated and patients are kept alive. There is a separate outpatient clinic for each disease. These are the hospital departments where there are specialist doctors for diseases.

**Prescriptions :**

The paper on which the doctor writes down the medicines that she deems necessary for her patient and the amounts and forms of use of them. It is the last stage in the system. It is the first stage of the patient's treatment

**Survey:**

Another name for the examination.

**System :**

The system is a platform where patients make an appointment by entering, choosing a day, a time, and a doctor. In addition, patients' examination information, examinations, examination results and prescriptions are also recorded on this platform.

**System Login :**

This is the section where the patient's name, surname and password are entered into the hospital information system. Patient-examination-doctor-disease is the beginning of prescription phases.

**System Admin:**

In other words, the hospital information system or interface. main provider integral part that does everyone's work within the system. System heart.

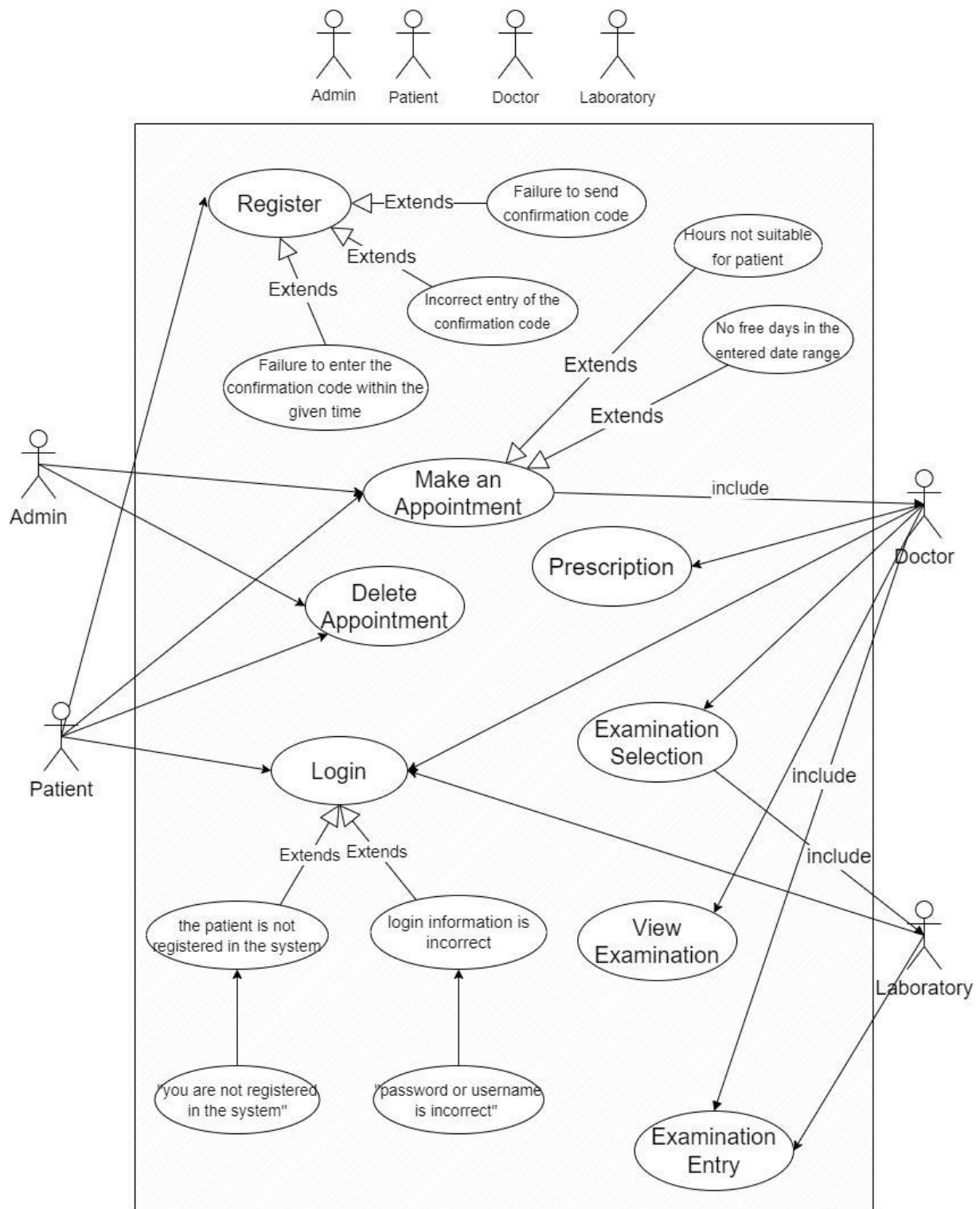
**UserName :**

Username is the patient's first and last name. The patient registers the first and last name while registering to the system. It is used to log in to the system when you want to make an appointment.

## USE CASE

Use Case Diagrams are used to show all the functions needed during the management of business processes, the actors that will trigger these functions, the actors that will be affected by the functions, and the relationships between the functions.

## USE-CASE ACTORS



## USE CASE OF THE USER'S REGISTER(ENROLL) SCENARIO

THE REGISTER USE-CASE
<p><b>Brief Description:</b> Recording the patient's information and verifying it with a confirmation code</p> <p><b>Entry Condition:</b> Obtaining the name, surname, ID number and telephone number from the patient to be registered</p> <p><b>Exit Condition:</b> Successful completion of registration</p> <p><b>Exeptions:</b></p> <ol style="list-style-type: none"><li>1. Failure to send confirmation code</li><li>2. Failure to enter the confirmation code within the given time</li><li>3. Incorrect entry of the confirmation code</li></ol>
<p><b>Step-by-step Description:</b></p> <ol style="list-style-type: none"><li>1. The patient enters their name, surname, ID number and phone number</li><li>2. The last six digits of the ID number become the password of the patient</li><li>3. Confirmation code is generated using the "first letter of name+first letter of last name+last four digits of ID" format</li><li>4. The confirmation code is sent to the phone as an SMS</li><li>5. If no confirmation code was sent, it will be sent again</li><li>6. The patient enters the confirmation code within 2 minutes</li><li>7. Return to 4 if the code is not entered within 2 minutes</li><li>8. The correctness of the code is checked, if it is correct, the patient is registered in the system</li><li>9. If the entered code is incorrect, the text "check the code" appears and return to 4</li><li>10. "patient registration completed" appears</li></ol>

## USE CASE OF THE USER'S LOGIN SCENARIO

THE LOGIN USE-CASE
<p><b>Brief Description:</b> Login of the user registered in the system with</p> <p><b>Entry Condition:</b> Login information is known</p> <p><b>Exit Condition:</b> The user logs in to the system</p> <p><b>Exeptions:</b></p> <ol style="list-style-type: none"><li>1. The patient is not registered in the system</li><li>2. Their information is incorrect</li></ol>
<p><b>Step-by-step Description</b></p> <ol style="list-style-type: none"><li>1. The user enters their login information</li><li>2. If the information entered by the patient is not in the system, "you are not registered in the system" appears</li><li>3. If the login information is incorrect, "login information incorrect" appears</li><li>4. If the information is correct, the user logs in to the system</li></ol>

## USE CASE OF THE USER'S MAKE APPOINTMENT SCENARIO

THE MAKE AN APPOINTMENT USE-CASE
<p><b>Brief Description:</b> Creates an appointment with the outpatient clinic, date and doctor requested by the patient</p> <p><b>Entry Condition:</b> The patient is logged in to the system</p> <p><b>Exit Condition:</b> The appointment is created</p> <p><b>Exeptions:</b></p> <ol style="list-style-type: none"><li>1. No free days in the entered date range</li><li>2. The patient does not find the free hours on the date he chooses suitable for himself</li></ol>
<p><b>Step-by-step Description</b></p> <ol style="list-style-type: none"><li>1. The patient chooses a polyclinic</li><li>2. The doctors of the selected clinic are listed</li><li>3. The patient chooses a doctor</li><li>4. The patient enters the date range</li><li>5. Shows available dates in the entered date range</li><li>6. Go back to 3 if there are no free dates</li><li>7. Show available times on the selected date</li><li>8. If the hours do not suit the patient, the patient returns to 5</li><li>9. The patient chooses the time</li><li>10. An appointment is created for the patient at the selected time</li><li>11. "your appointment has been created" appears</li></ol>

## USE CASE OF THE USER'S APPOINTMENT CANCELLATION SCENARIO

THE DELETE APPOINTMENT USE-CASE
<p><b>Brief Description:</b> Login of the user registered in the system with</p> <p><b>Entry Condition:</b> The patient has an appointment</p> <p><b>Exit Condition:</b> The appointment is deleted from the system</p>
<p><b>Step-by-step Description</b></p> <ol style="list-style-type: none"><li>1. Appointments are listed</li><li>2. The patient chooses an appointment</li><li>3. The patient cancels the appointment</li></ol>

## USE CASE OF THE USER'S DOCTOR SCENARIOS

### THE EXAMINATION SELECTION USE-CASE

**Brief Description:** The doctor selects examination for the patient

**Entry Condition:** The patient has an appointment

**Exit Condition:** The doctor selects an examination

#### Step-by-step Description

1. The doctor logs in to the patient's information system
2. The doctor selects examination for the patient

### THE EXAMINATION ENTRY USE-CASE

**Brief Description:** The laboratory enters the laboratory results into the system

**Entry Condition:** the doctor should have chosen an examination and the patient's examination results should be known

**Exit Condition:** The results of the examination are entered into the system

#### Step-by-step Description

1. The laboratory enters to the patient's information system
2. The laboratory enters the laboratory results into the system

### THE VIEW EXAMINATION USE-CASE

**Brief Description:** The doctor displays the results of the examination

**Entry Condition:** The laboratory uploads the results of the examination to the system

**Exit Condition:** The doctor displays the results of the examination

#### Step-by-step Description

1. The doctor enters to the patient's information system
2. The doctor displays the examination results



## THE PRESCRIPTION USE-CASE

**Brief Description:** The doctor enters the patient's diagnosis and prescription into the system

**Entry Condition:** The doctor logs in to the patient's information system

**Exit Condition:** The diagnosis and the prescription are saved in the system

### Step-by-step Description

1. The doctor enters the patient's diagnosis into the system
2. The doctor enters the patient's prescription into the system

## USE CASE OF THE USER'S LABMANAGER SCENARIO

## THE EXAMINATION ENTRY USE-CASE

**Brief Description:** The laboratory enters the laboratory results into the system

**Entry Condition:** the doctor should have chosen an examination and the patient's examination results should be known

**Exit Condition:** The results of the examination are entered into the system

### Step-by-step Description

1. The laboratory enters to the patient's information system
2. The laboratory enters the laboratory results into the system

## FUNCTIONAL REQUIREMENTS

They define the functional infrastructure of the system with the services to be provided by the system. They reveal structurally and functionally what the system will do. Development-independent are mostly requirements for input, output interfaces, processes and error management. Permission and authorization requirements at system login are also of this type. It determines what the system will do in a detailed way, not abstractly.

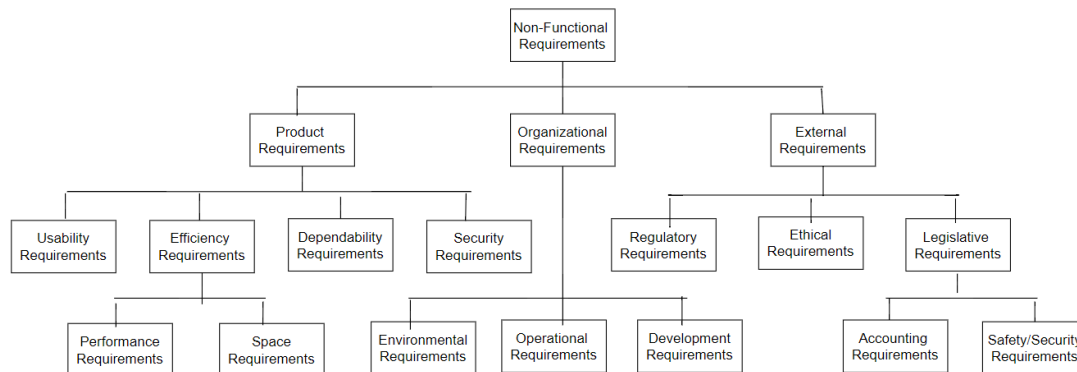
- ❖ Business rules
- ❖ Transaction corrections, edits and cancellations
- ❖ Administrative functions
- ❖ Authentication
- ❖ Authorization levels
- ❖ Audit Tracking
- ❖ External Interfaces
- ❖ Certification Requirements
- ❖ Reporting Requirements
- ❖ Historical data
- ❖ Legal or Regulatory Requirements



FUNCTIONAL REQUIREMENTS NO	REQUIREMENT DEFINITION	WHAT TO DO?
1	Select enroll or login	The hospital information system(HIS) must leads to the user's selected page.
2	Select type of record	The HIS must leads to the user's selected page.
3	Enroll	The HIS must create a place for the user to the user to the enter their name,ID number,hospital number,phone number,password.
4	Enter password	The user must enter the password given by the hospital information system in the password section.
5	Select type of user	The HIS must set the user's password and creates space for the user to enter the password.
6	Confirmation code control system	It must send the HIS confirmation code to the user's number,create a place for him to enter the code,and verify its accuracy.
7	Select type of user	The HIS must leads to the user's selected page.
8	Login	The HIS must create a place for the user to enter ID number,hospital number and password to login.
9	Select make appointment	HIS must view the list of polyclinics and doctors,available dates and available times for the user to make appointment.After the checks are made,it should create and save the appointment.
10	Select interrogete/delete appointment	The HIS must view the appointment list,delete the selected appointment and checks should be made.
11	Select lab results,diagnosis and prescription	The HIS must view lab results,diagnosis and prescription.
12	Patient ID number enter	The HIS must provide space for the medical personnal to enter the patient's ID number.
13	Examination	The HIS must view the examination list and save when user select examination.
14	Desired examination	The HIS must view the saved examination.
15	Desired examination result enter	The HIS must view the examination result list and save the selected examination result list.
16	Diagnosis enter	It must view the results of the HIS examination,enter the date and time of the diagnosis,open the place,give permission to write and control.
17	Prescription enter	The HIS must show the disease results,enter the prescription,open the date and time,give permission to write and control.

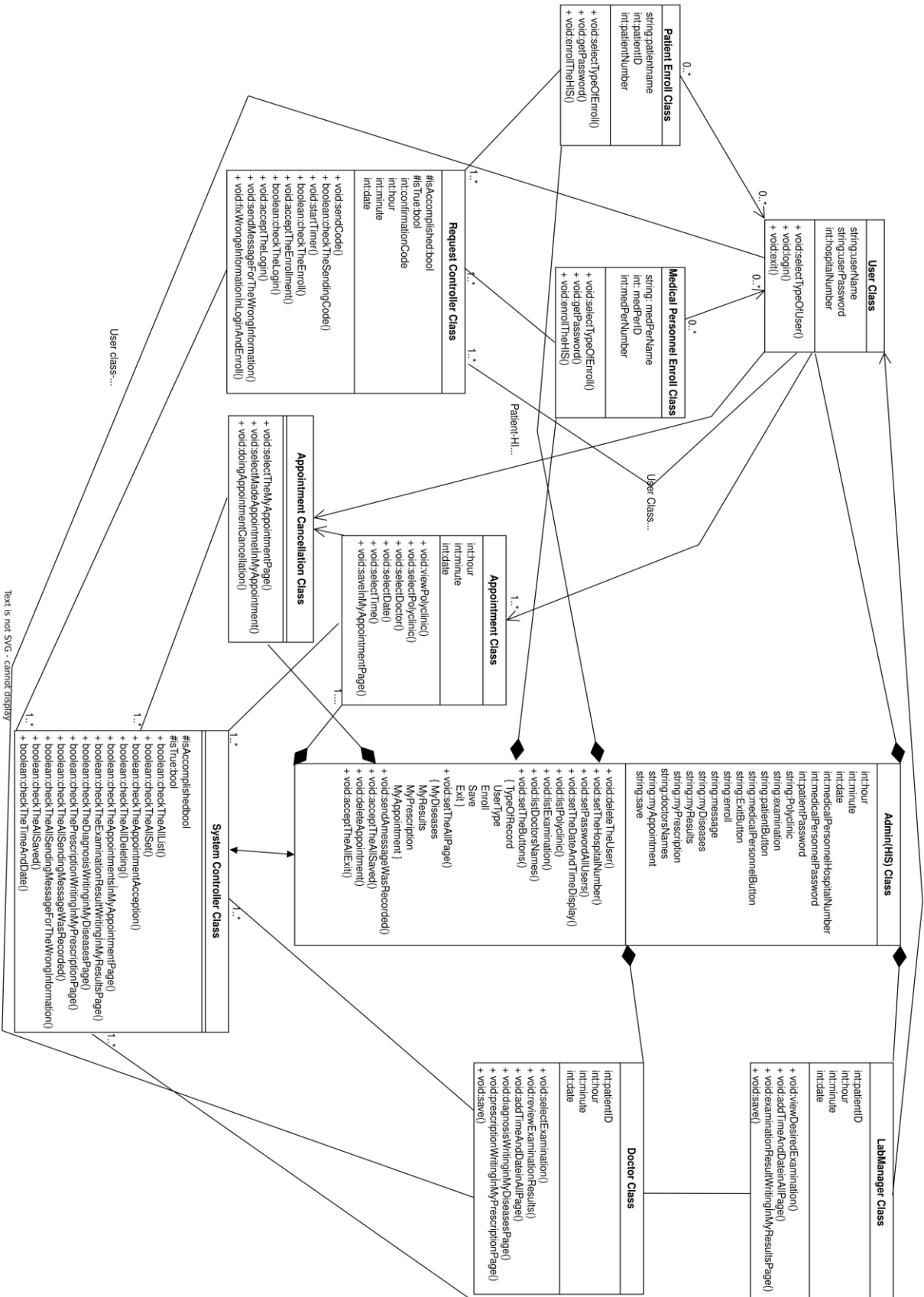
# NON-FUNCTIONAL REQUIREMENTS

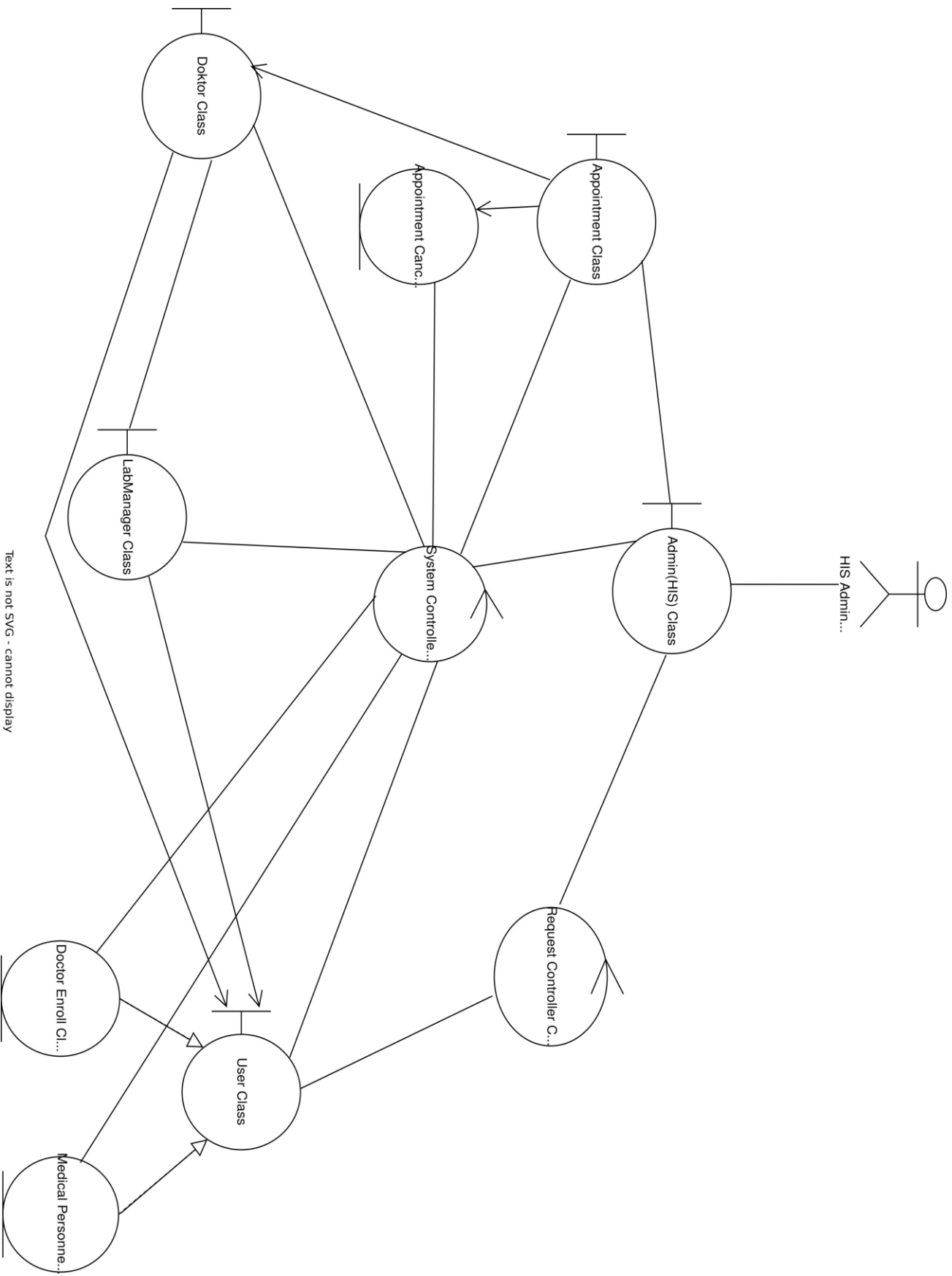
These are the requirements that determine the physical environment, interfaces, being user-oriented, security, reliability, quality assurance, etc. Although they do not add functionality to software, such requirements play a critical role, especially in terms of software quality. Unless these requirements are met in the software, the usability of the software will be insufficient.



NON-FUNCTIONAL REQUIREMENTS NO	QUALITY ATTRIBUTE	REQUIREMENT DEFINITION	HOW DOES IT WORK?
1	Security	Enter password	Password shall the last 6 digits of the ID number,hospital number.
2	Security	Send a confirmation code	Confirmation code shall in capital letters,first letter of name,first and, last four digits of ID number.
3	Security	Enter confirmation code	The user shall enter the confirmation code into the HIS with in 2 minutes.
4	Performance and Usability	Make appointment	Appointments suitable for the doctor's date range are quickly listed in the HIS.
5	Performance and Usability	Make appointment	The hours available for the appointment are quickly listed.
6	Regulatory	Interrogete/Delete appointment	When the time is selected,the HIS shall save it in the appointments section.
7	Performance and Usability	Examination result	Examination results are quickly listed in order of entry time/date.
8	Manageability	Hospital information system	The program shall executed in harmony both HIS and user.
9	Dependability	Hospital information system	The HIS shall secure from hacking for or another threats.
10	Space	Hospital information system	The HIS database's capacity shall enough for all actors,appointment,examination and prescription.
11	Collaboration	Hospital information system	The HIS shall be able to be accessed everywhere like telephone.
12	Maintenance	Hospital information system	Maintenance shall be easy.

## CLASS DIAGRAM





Text is not SVG - cannot display

## CRC(CLASS RESPONSIBILITY COLLABORATION)CARDS OF CLASS DIAGRAM

It is an easy-to-implement and secure error-detection method that is mostly used in digital networks and storage devices and detects erroneous changes in raw data.

<b>CLASS: User Class</b>
<b>RESPONSIBILITY</b> It is the class where people who want to use the hospital information system will enter after enroll. 1- User Class select the Type of user for the login. 2- User Class does the login. 3- User Class does the exit.
<b>COLLABORATION</b> 1-Patient Enroll Class(subclass) 2-Medical Personnel Enroll Class(subclass) 3-Request Controller Class 4-Appointment Class 5-LabManager Class 6-Doctor Class 7-Admin 8-Admin(HIS) Class(superclass)
<b>CLASS: Patient Enroll Class</b>
<b>RESPONSIBILITY</b> It is the class in which the patient registers by giving and receiving some information to the system in order to log in to the system. 1- Patient Enroll Class select the type of enroll for the enrollment. 2- Patient Enroll Class after the entering gets its password from HIS Class. 3- Patient Enroll Class enrollments in HIS
<b>COLLABORATION</b> 1-User Class(superclass) 2-Request Controller Class 3-Admin(HIS) Class(superclass)

**CLASS: Medical Personnel Enroll Class****RESPONSIBILITY**

It is the class in which the medical personnel registers by giving and receiving some information to the system in order to log in to the system.

- 1-Medical Personnel Enroll Class select the type of enroll for the enrollment.
- 2-Medical Personnel Enroll Class after the entering gets its password from HIS Class.
- 3-Medical Personnel Enroll Class enrollments in HIS.

**COLLABORATION**

- 1-User Class(superclass)
- 2-Request Controller Class
- 3-Admin(HIS) Class(superclass)

**CLASS: Appointment Class****RESPONSIBILITY**

It is the class in which the patient makes an appointment to be treated.

- 1-Appointment Class views the all polyclinic.
- 2-Appointment Class select a polyclinic.
- 3-Appointment Class select a doctor.
- 4-Appointment Class select a date.
- 5-Appointment Class select a time.
- 6-Appointment Class saves the Appoint in My Appointment page.

**COLLABORATION**

- 1-Appointment Cancellation Class(subclass)
- 2-User Class
- 3-System Controller Class
- 4-Admin(HIS) Class(superclass)

**CLASS: Appointment Cancellation Class****RESPONSIBILITY**

This is the class used when an appointment is requested to be canceled.

1-Appointment Cancellation Class selects the my appointment page and select made appointment in my appointment page.

2-Appointment Cancellation Class does the cancellation.

**COLLABORATION**

1-User Class

2-Appointment Class

3-System Controller Class

**CLASS: LabManager Class****RESPONSIBILITY**

the class that performs the patient's examinations.

1-LabManager Class views the desired examination.

2-LabManager Class writing examination results in My Results.

3-LabManager Class adds time and date in all page.

4-LabManager saves what his/her do.

**COLLABORATION**

1-User Class

2-LabManager Class

3-System Controller Class

4-Admin(HIS) Class(superclass)

<b>CLASS: Doctor Class</b>
<p><b>RESPONSIBILITY</b></p> <p>It is the class that includes the doctor's relations with the system and the patient.</p> <ol style="list-style-type: none"> <li>1-Doctor Class selects examination.</li> <li>2-Doctor Class reviews the examination res</li> <li>3-Doctor Class writing diagnosis in My dise: page.</li> <li>4-Doctor Class writing prescription in my prescription page.</li> <li>5-Doctor Class adds the time and date in al page.</li> <li>6-Doctor Class saves what his/her do.</li> </ol>
<p><b>COLLABORATION</b></p> <ol style="list-style-type: none"> <li>1-User Class</li> <li>2-LabManager Class</li> <li>3-System Controller Class</li> <li>4-Admin(HIS) Class(superclass)</li> </ol>

<b>CLASS: Admin (HIS) Class</b>
<p><b>RESPOSIBILITY</b></p> <p>System that is the interface-manager-heart of the system.</p> <ol style="list-style-type: none"> <li>1-HIS can delete the user.</li> <li>2-HIS can delete appointment.</li> <li>3-HIS sets hospital number,</li> <li>4-HIS sets password.</li> <li>5-HIS sets date and time display.</li> <li>6-HIS sets butttons.</li> <li>7-HIS sets all page.</li> <li>8-HIS lists polyclinic.</li> <li>9-HIS lists examination.</li> <li>10-HIS lists doctors name.</li> <li>11-HIS sends a message was recorded</li> <li>12-HIS accepts all saved.</li> <li>13-HIS accepts all exit.</li> </ol>
<p><b>COLLABORATION</b></p> <ol style="list-style-type: none"> <li>1-User Class</li> <li>2-Patient Enroll Class</li> <li>3-Medical Personnerl Enroll Class</li> <li>4-Appointment Class</li> <li>5-Appointment Cancellation</li> <li>6-System Controller Class</li> <li>7-LabManager Class</li> <li>8-Doctor Class</li> </ol>

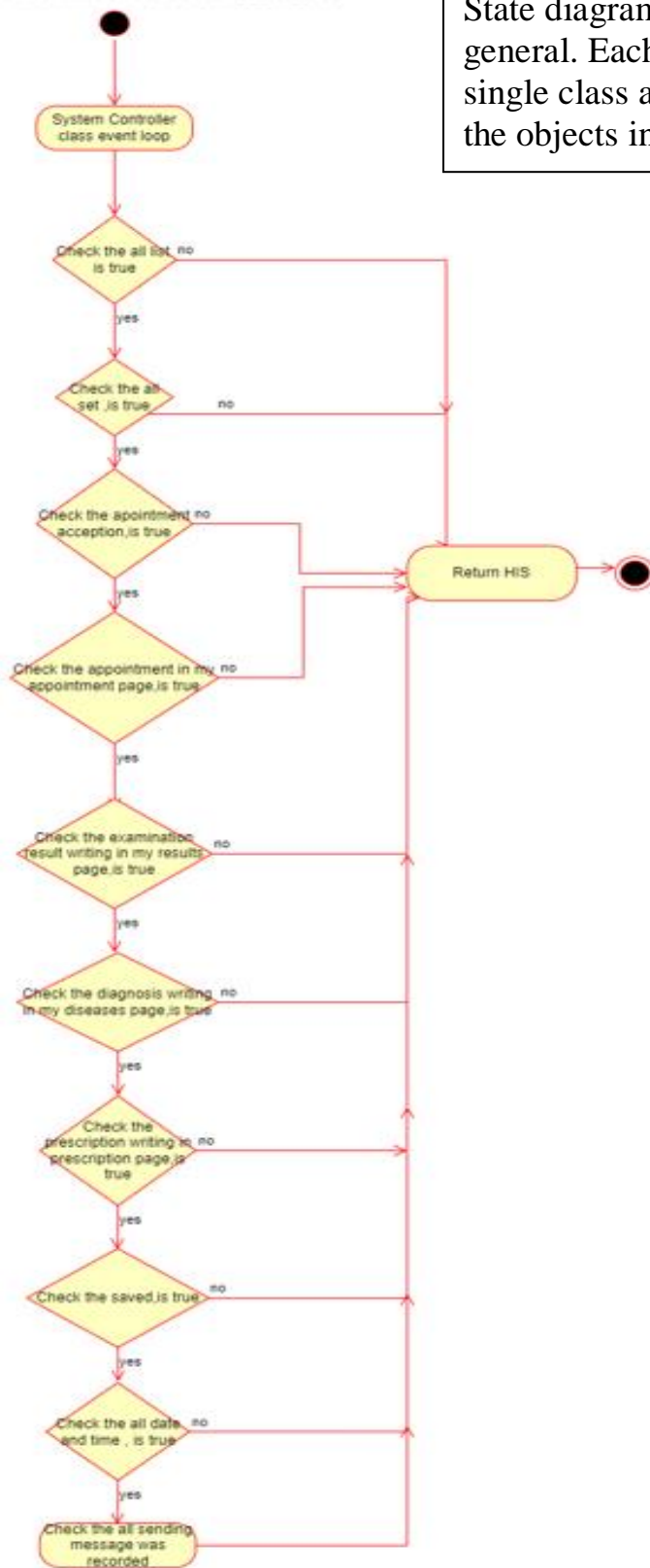


<b>CLASS:Request Controller Class</b>
<p><b>RESPONSIBILITY</b></p> <p>It is the class that provides the controls in the registration and login sections.</p> <ol style="list-style-type: none"> <li>1-Request Controller Class sends a code and checks the sending code.</li> <li>2-Request Controller Class starts timer.</li> <li>3-Request Controller Class checks all enrollment and accepts them.</li> <li>4-Request Controller Control checks the all login and accepts them.</li> <li>5-Request Controller Class if there is wrong information in sending code phase, enrollment phase or login phase, its send a message for the wrong information.</li> <li>6-Request Controller Class fixes wrong information in login and enroll phases.</li> </ol>
<p><b>COLLABORATION</b></p> <ol style="list-style-type: none"> <li>1-User Class</li> <li>2-Patient Enroll Class</li> <li>3-Medical Personnel Enroll Class</li> </ol>

<b>CLASS: System Controller Class</b>
<p><b>RESPONSIBILITY</b></p> <p>It is the class that provides all the controls needed for the system to function properly.</p> <ol style="list-style-type: none"> <li>1-System Controller Class checking all list.</li> <li>2-System Controller Class checking all set.</li> <li>3-System Controller Class checking appointment acceptance.</li> <li>4-System Controller Class checking all deleting.</li> <li>5-System Controller Class checking appointments in my appointment page.</li> <li>6-System Controller Class checking examination results writing in mt results page.</li> <li>7-System Controller Class checking diagnosis writing in my diseases page.</li> <li>8-System Controller Class checking prescription writing in my prescription page.</li> <li>9-System Controller Class checking all sending message was recorded</li> <li>10-System Controller Class checking all saved</li> <li>11-System Controller Class checking time and date</li> </ol>
<p><b>COLLABORATION</b></p> <ol style="list-style-type: none"> <li>1-Request Controller Class</li> <li>2-Appointment Class</li> <li>3-Appointment Cancellation Class</li> <li>4-LabManage Class</li> <li>5-Doctor Class</li> <li>6-Admin(HIS) Class</li> </ol>

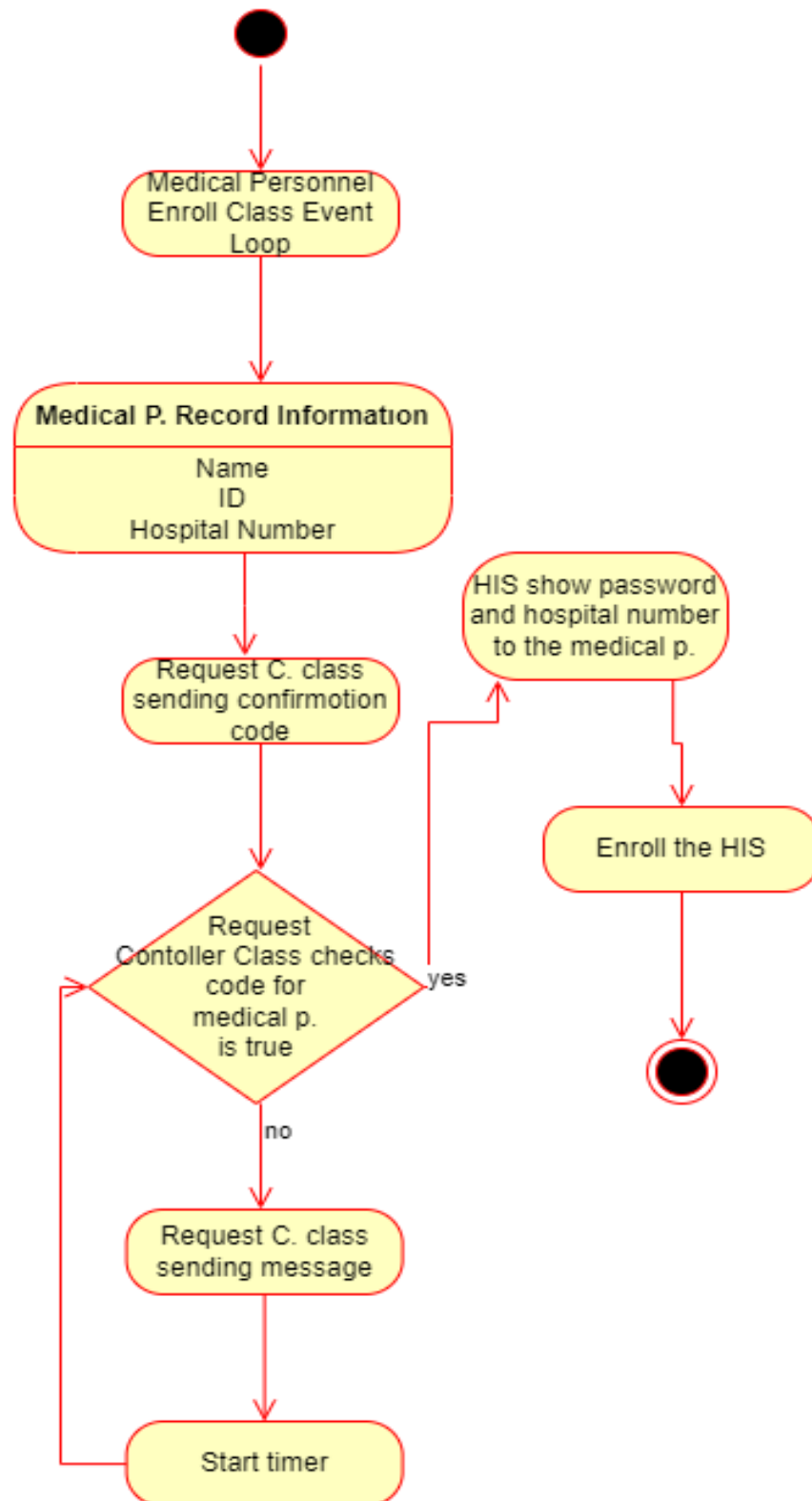
## STATECHART DRAWING OF EACH CLASS

STATECHART DRAWING SYSTEM COTROLLER CLASS

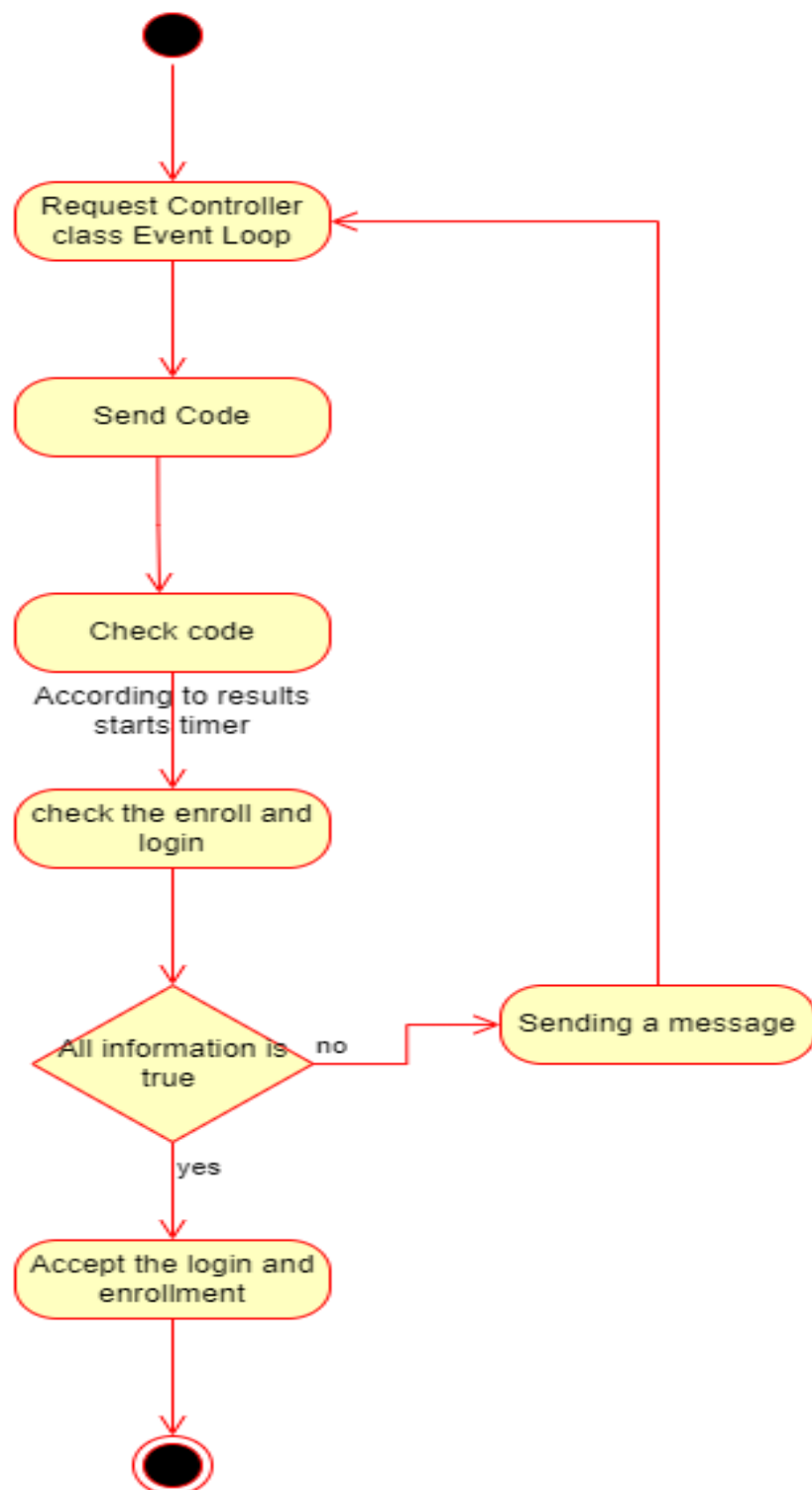


State diagrams show the behavior of systems in general. Each diagram contains the objects of a single class and the states, transitions, events of the objects in the system.

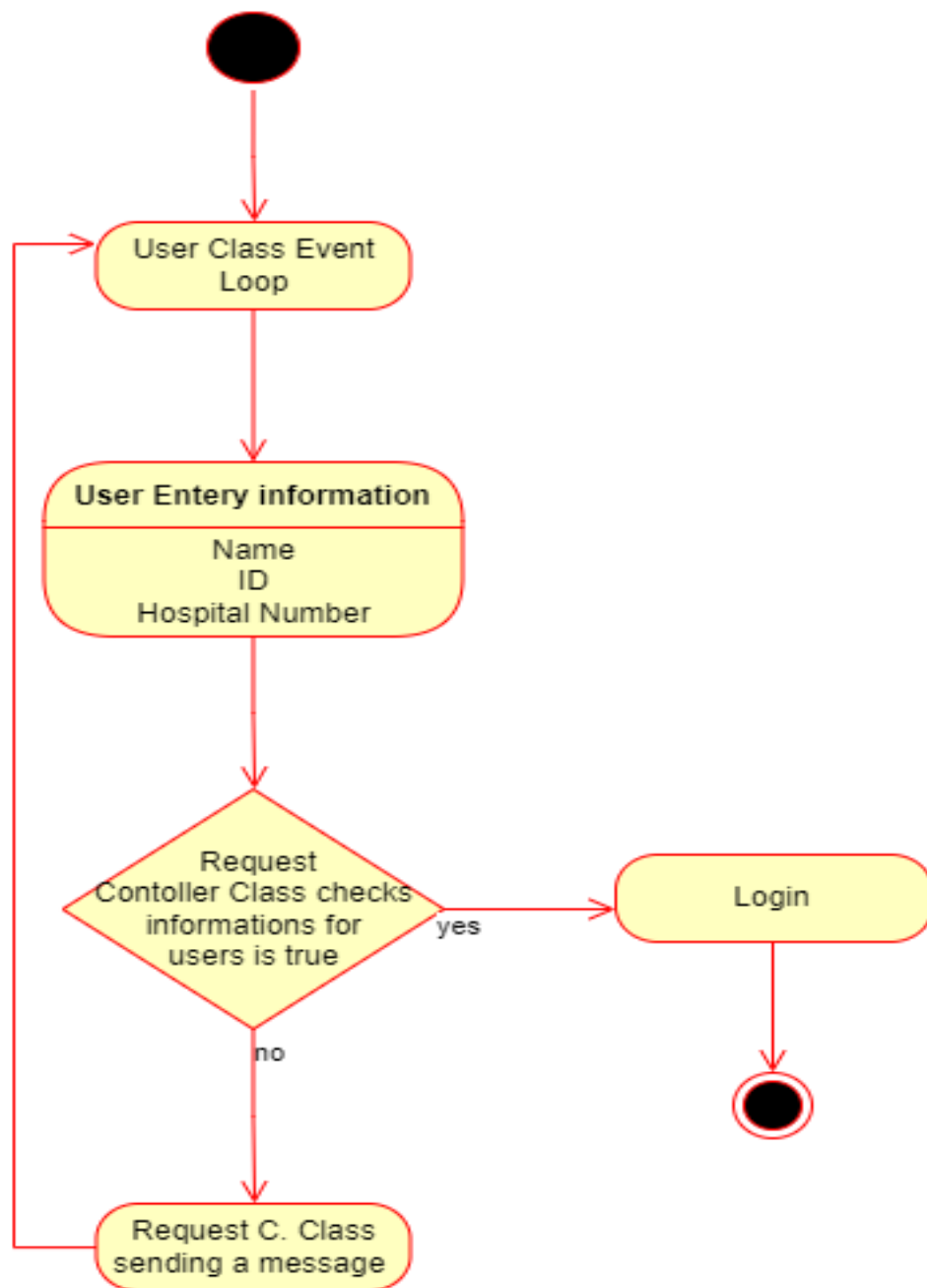
# STATECHART DRAWING FOR MEDICAL PERSONNEL ENROLL CLASS



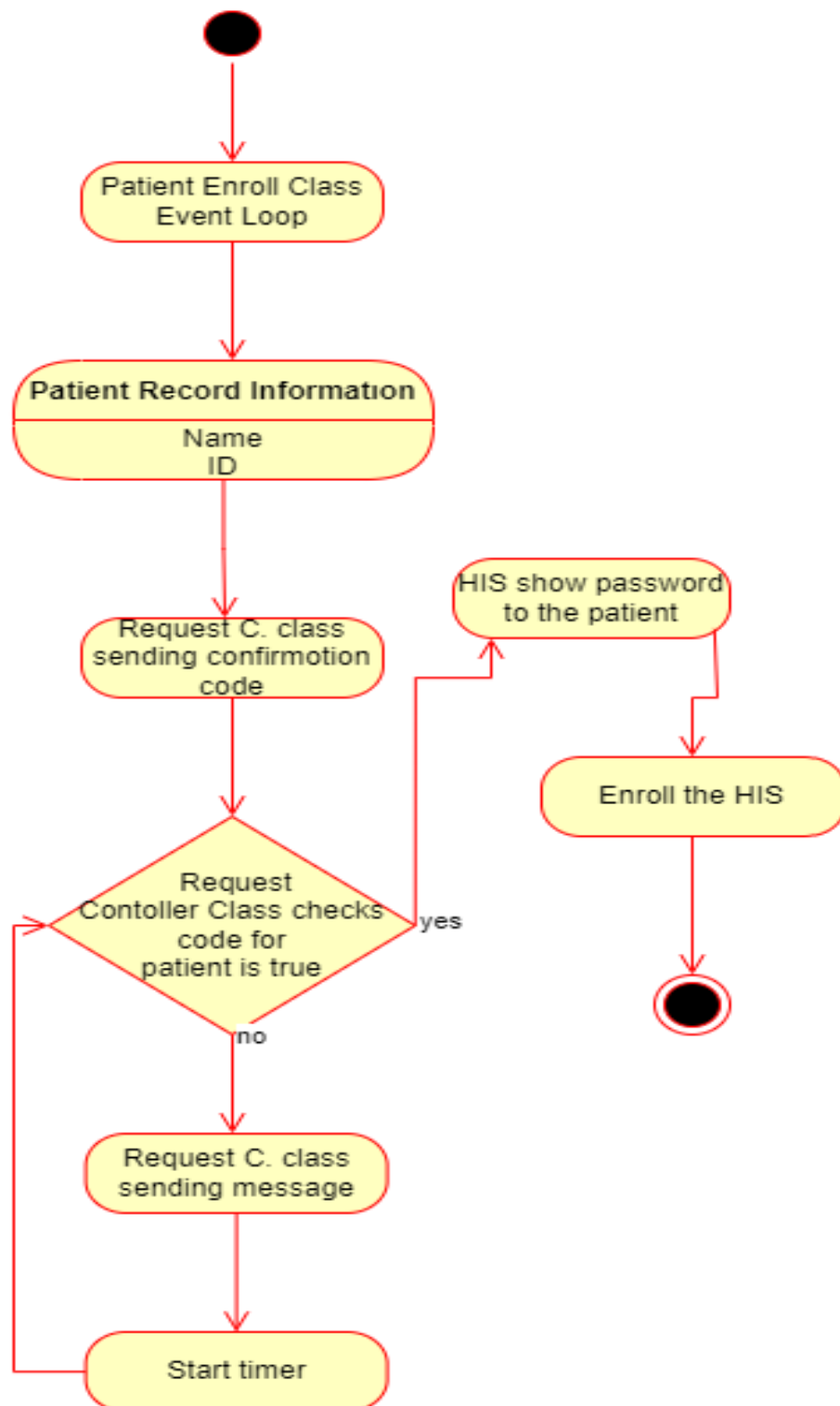
# STATECHART DRAWING FOR REQUEST CONTROLLER CLASS



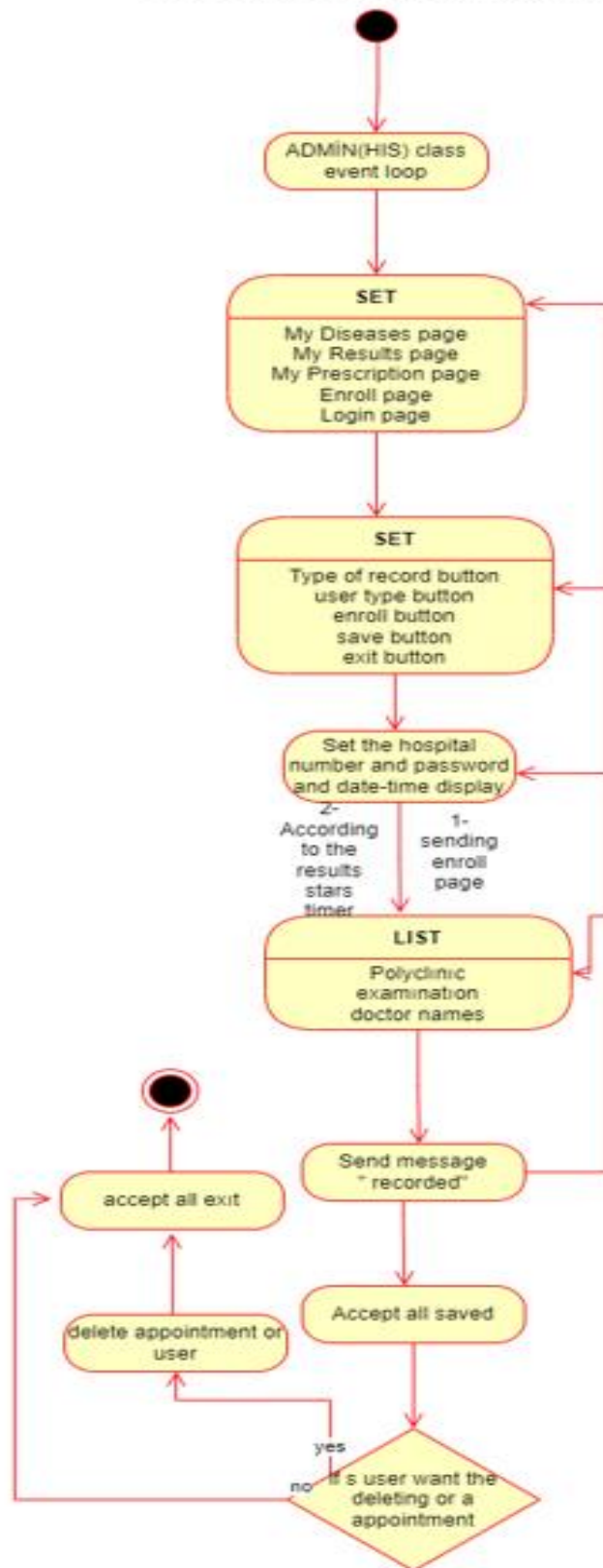
# STATECHARTS DRAWING FOR USER CLASS



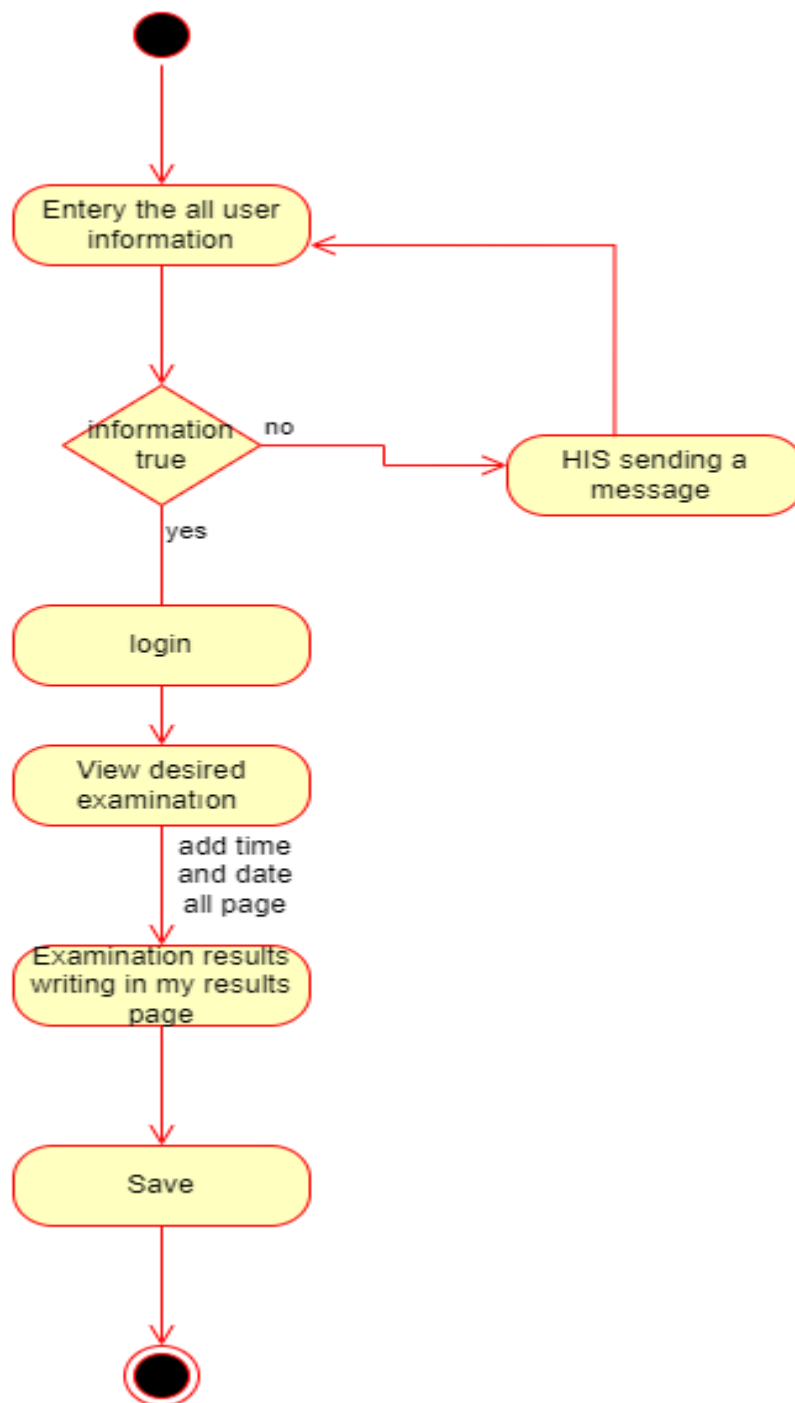
# STATECHART DRAWING FOR PATIENT ENROLL CLASS



# STATECHART DRAWING FOR ADMIN(HIS) CLASS

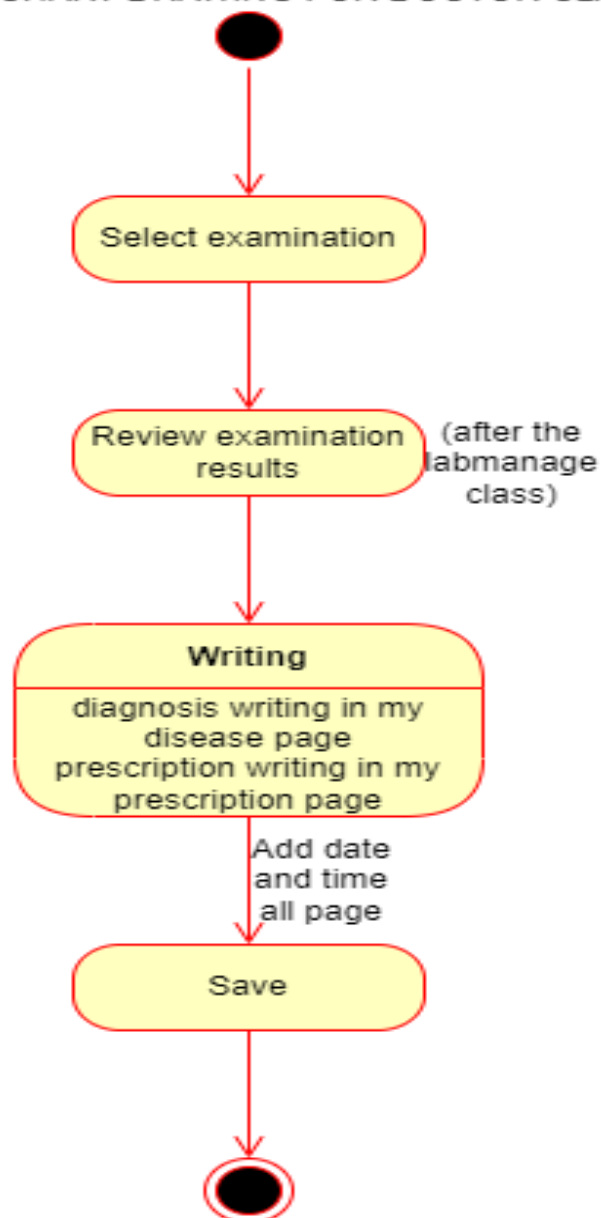


# STATECHART DRAWING FOR LABMANAGER CLASS

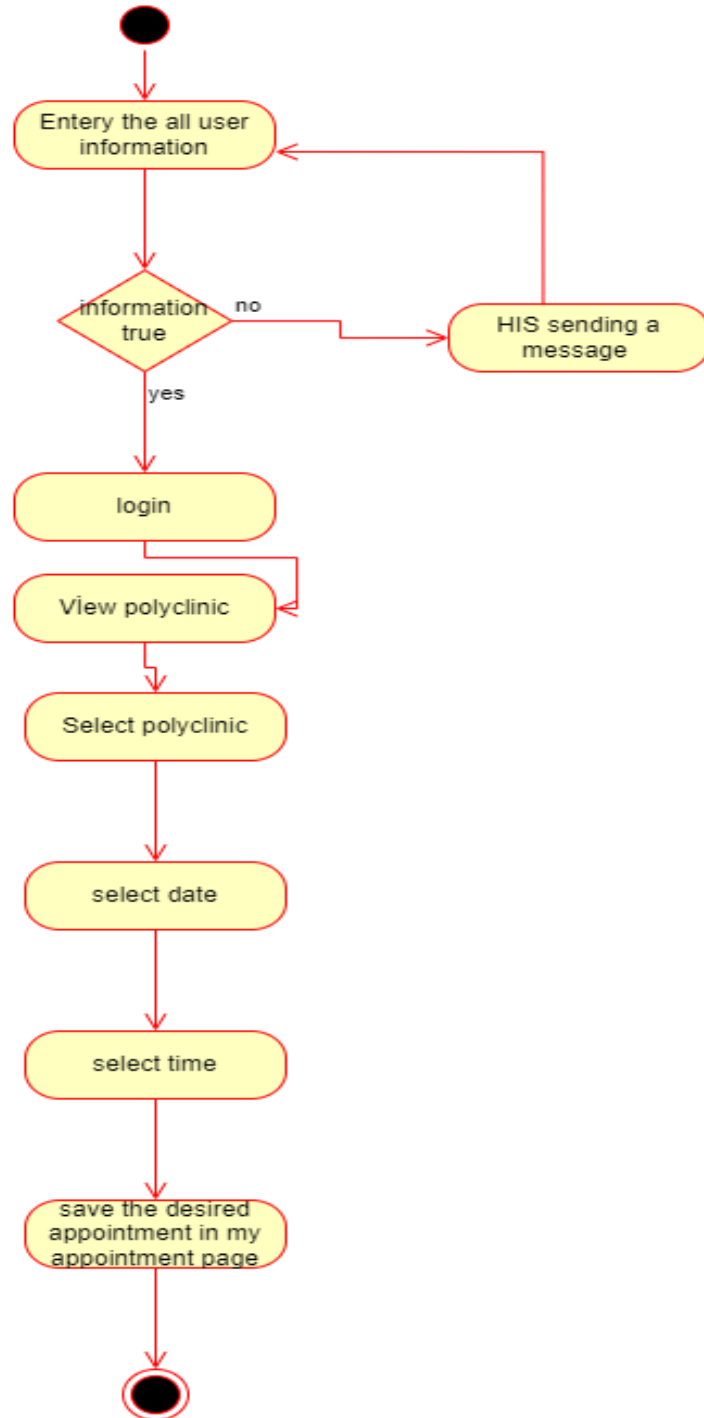




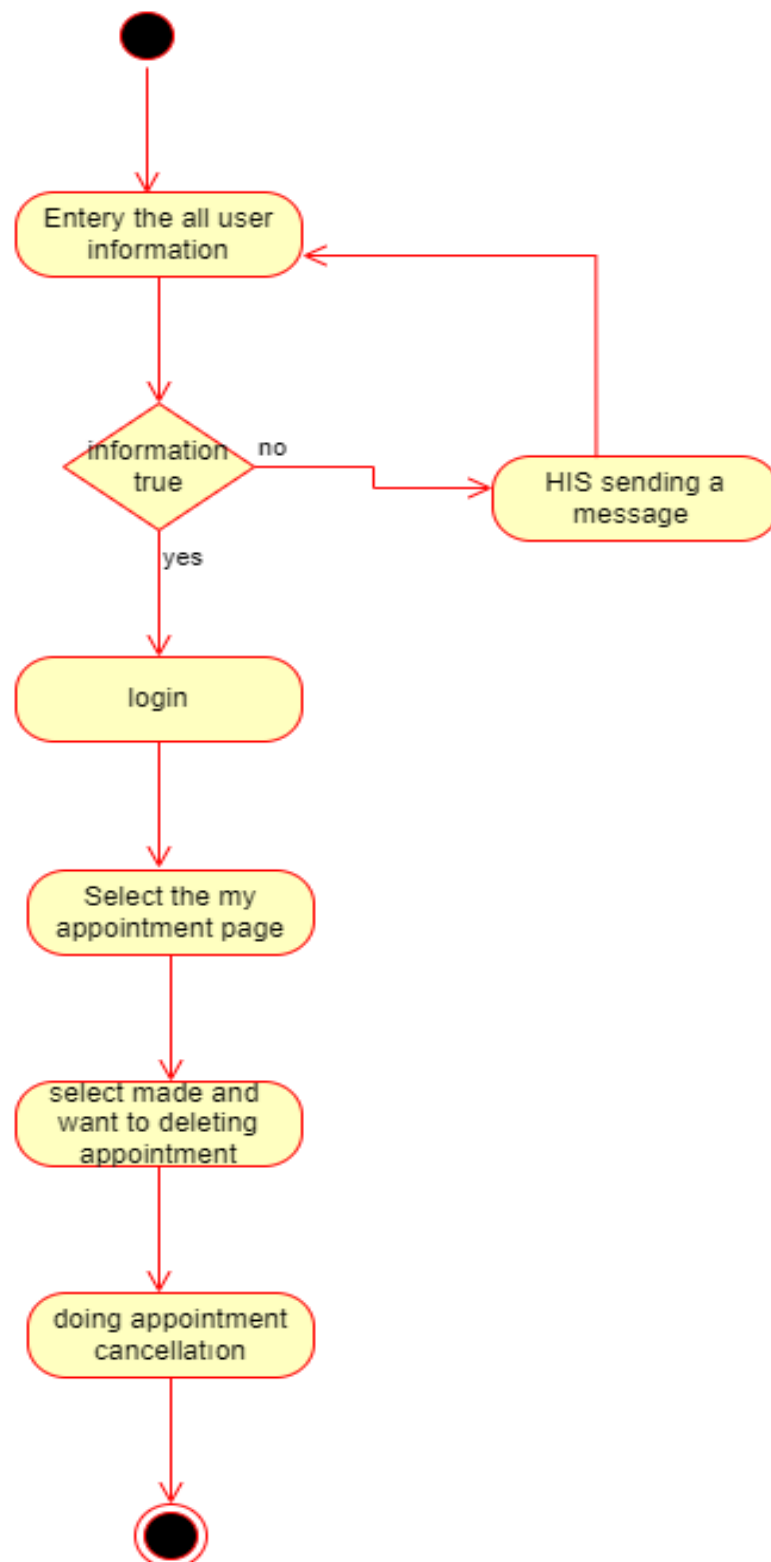
# STATECHART DRAWING FOR DOCTOR CLASS



# STATECHART DRAWING FOR APPOINTMENT CLASS



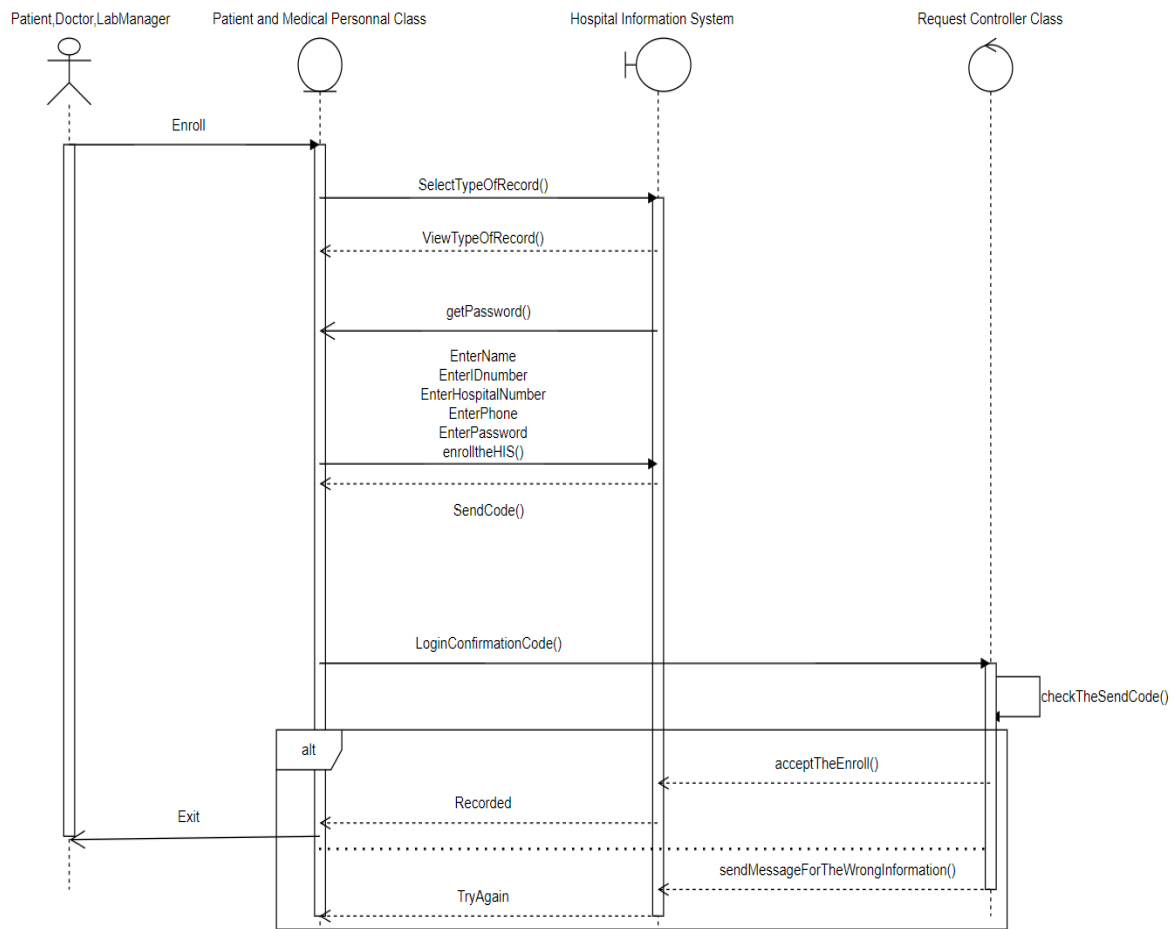
# STATECHART DRAWING APPOINTMENT CANCELLATION CLASS



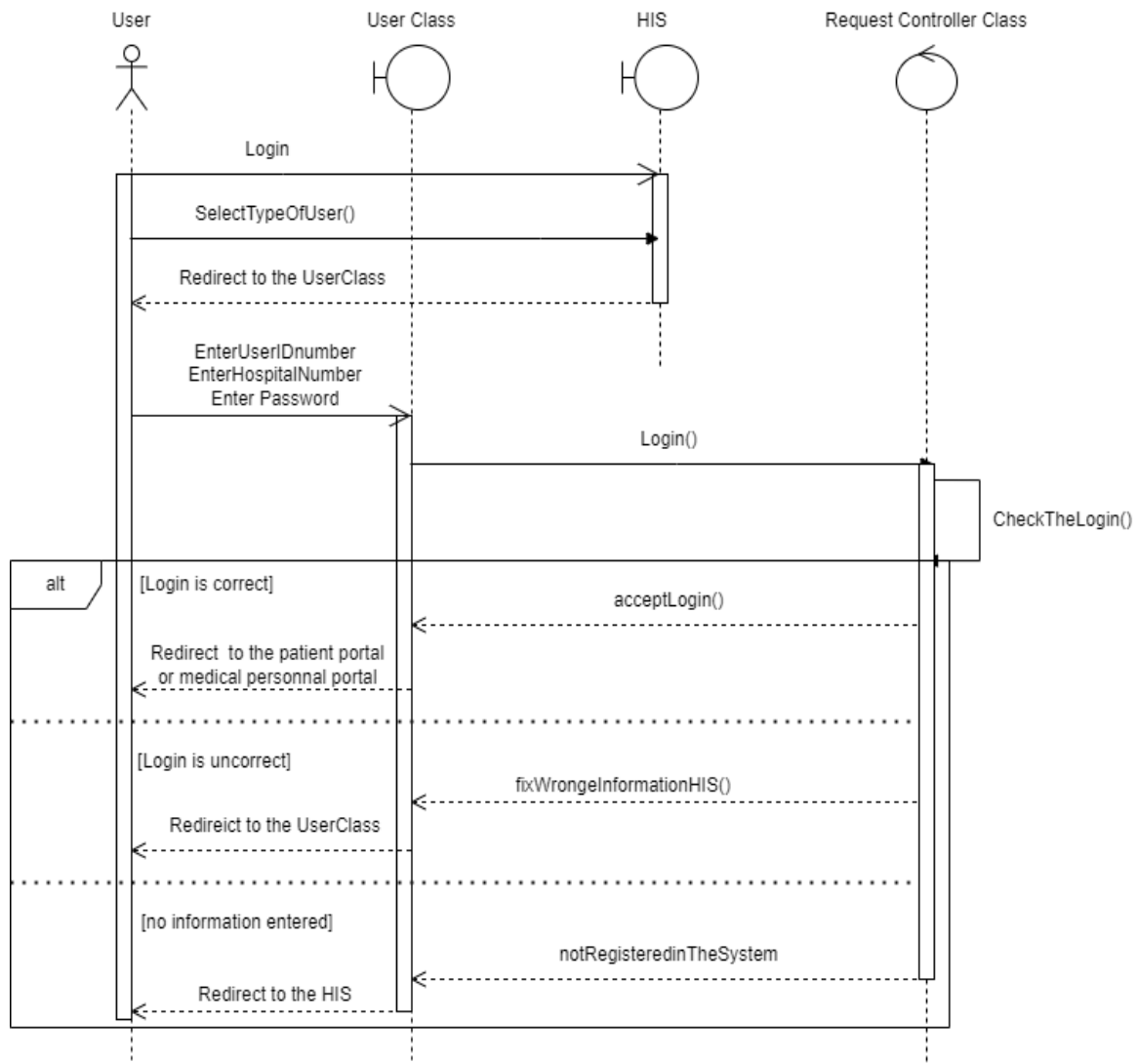
## SEQUENCE DIAGRAMS

Sequence Diagram is mostly used to show the communication of objects with each other in order. Makes connections between objects and tasks clearer. Shows the time-based ranking.

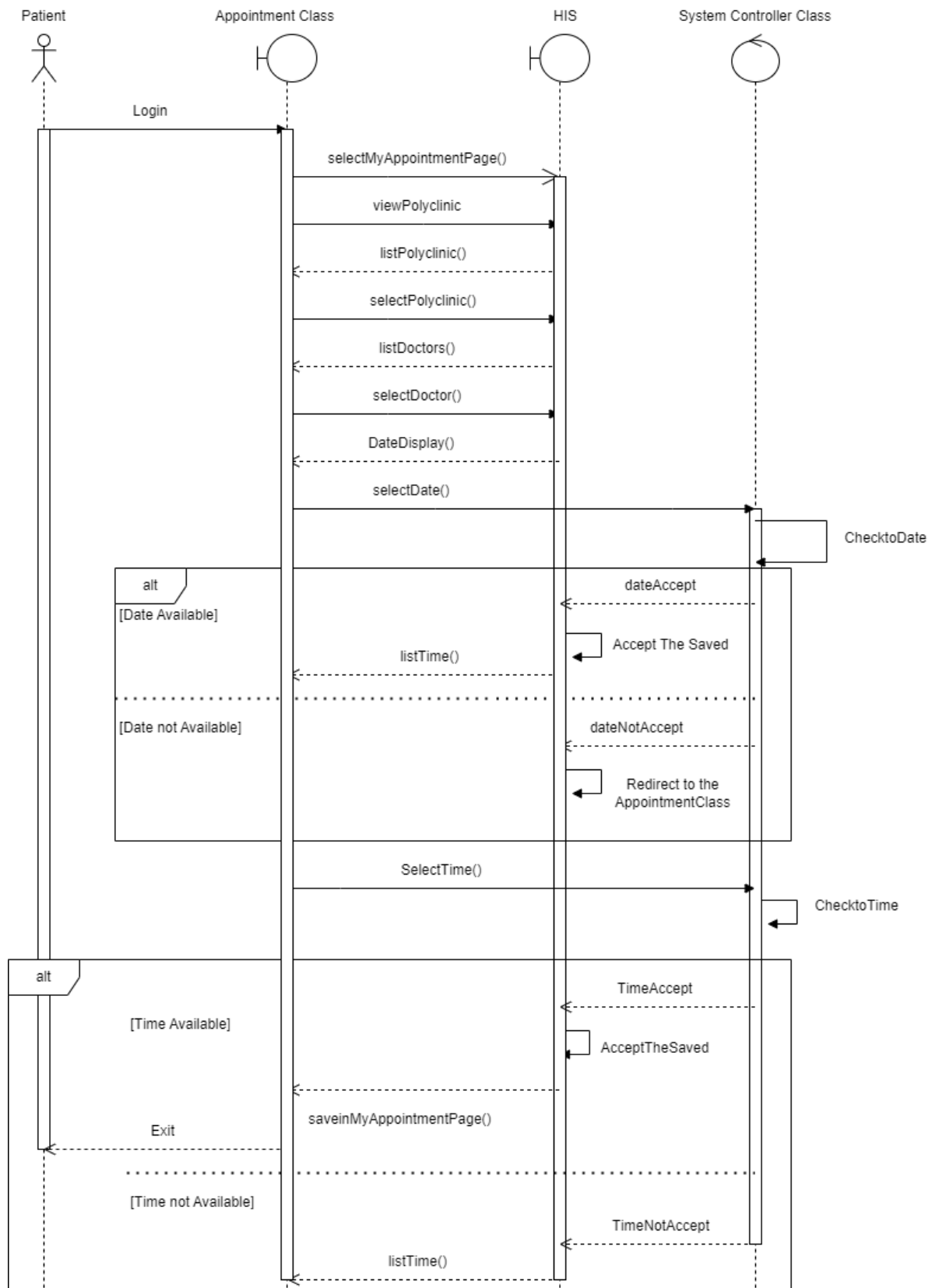
### SEQUENCE DIAGRAM OF THE USER'S REGISTER(ENROLL) SCENARIO



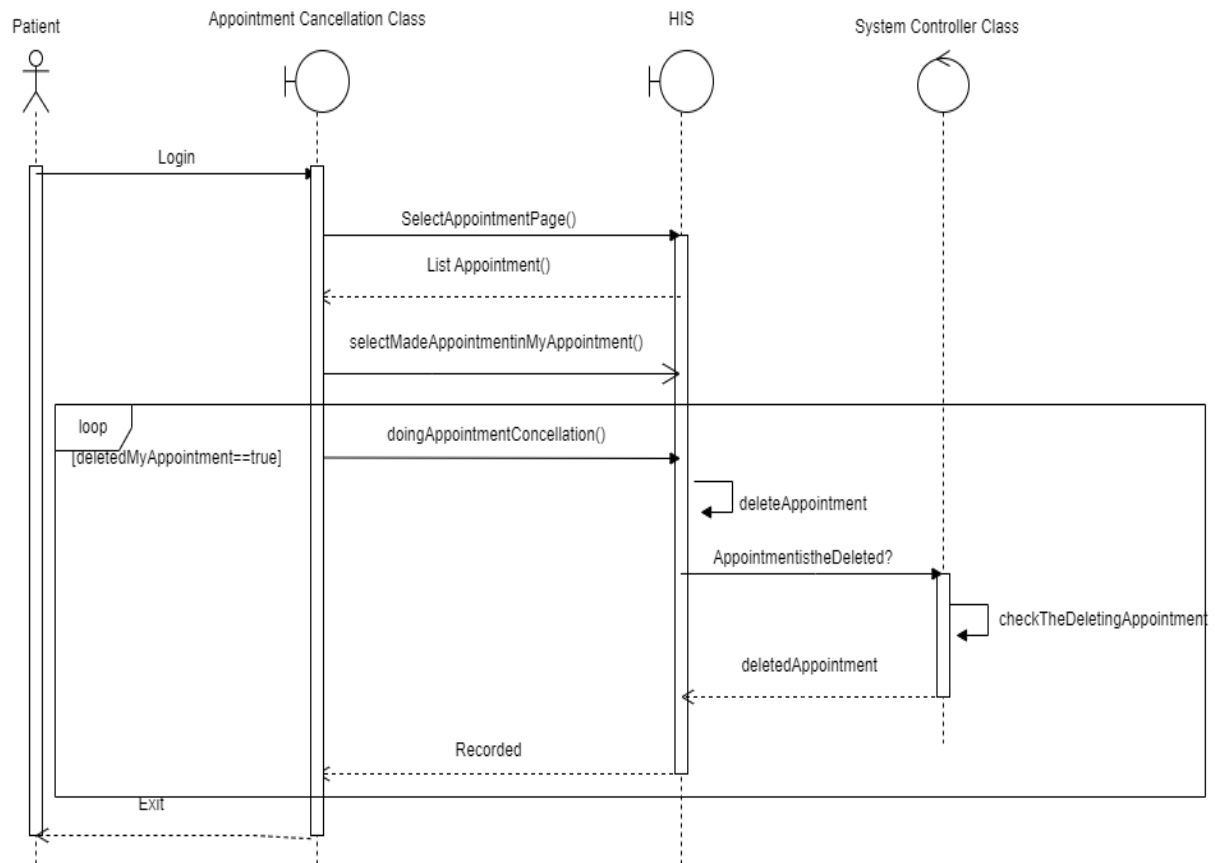
## SEQUENCE DIAGRAM OF THE USER'S LOGIN SCENARIO



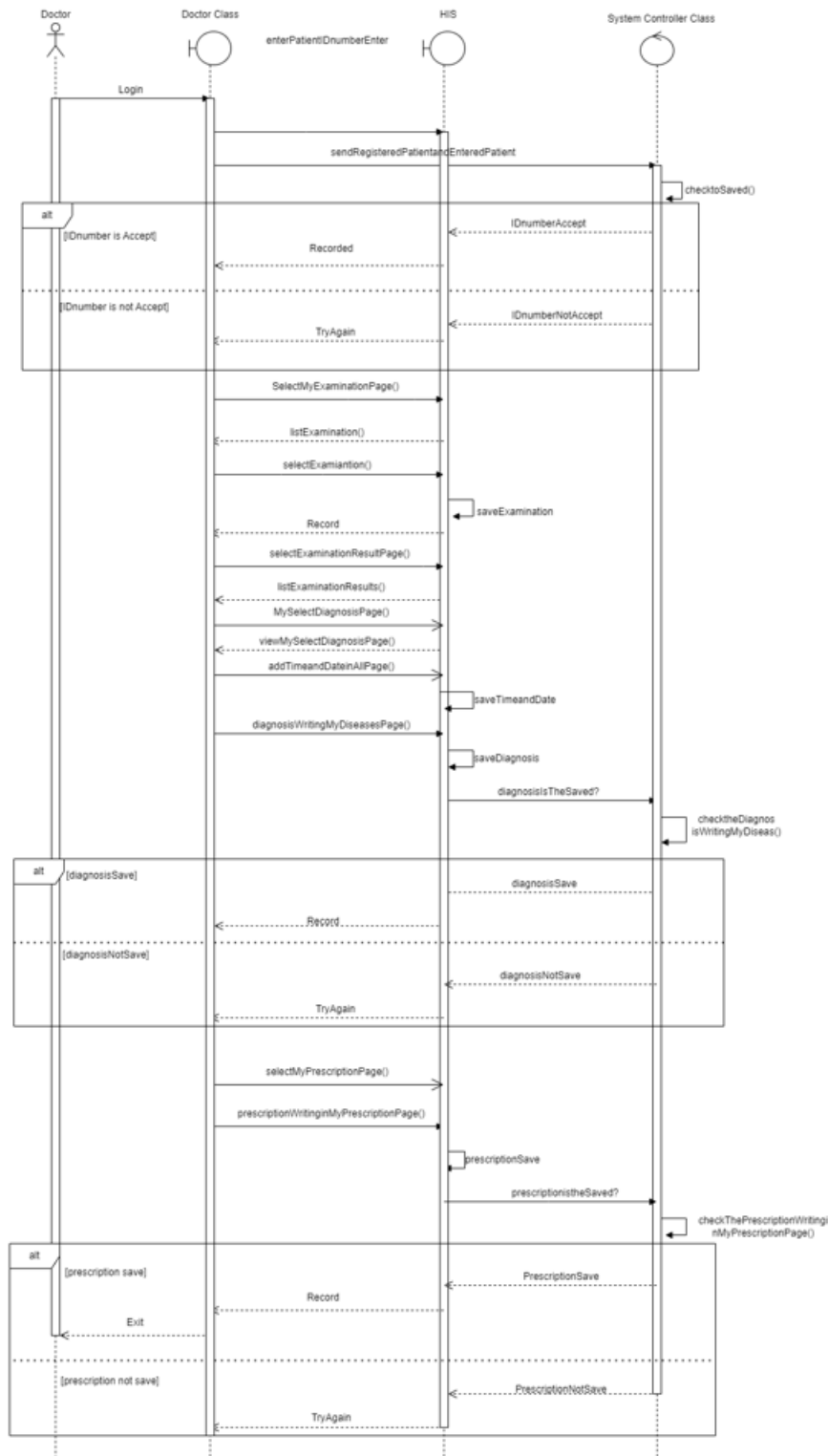
## SEQUENCE DIAGRAM OF THE USER'S MAKE APPOINTMENT SCENARIO



## SEQUENCE DIAGRAM OF THE USER'S APPOINTMENT CANCELLATION SCENARIO

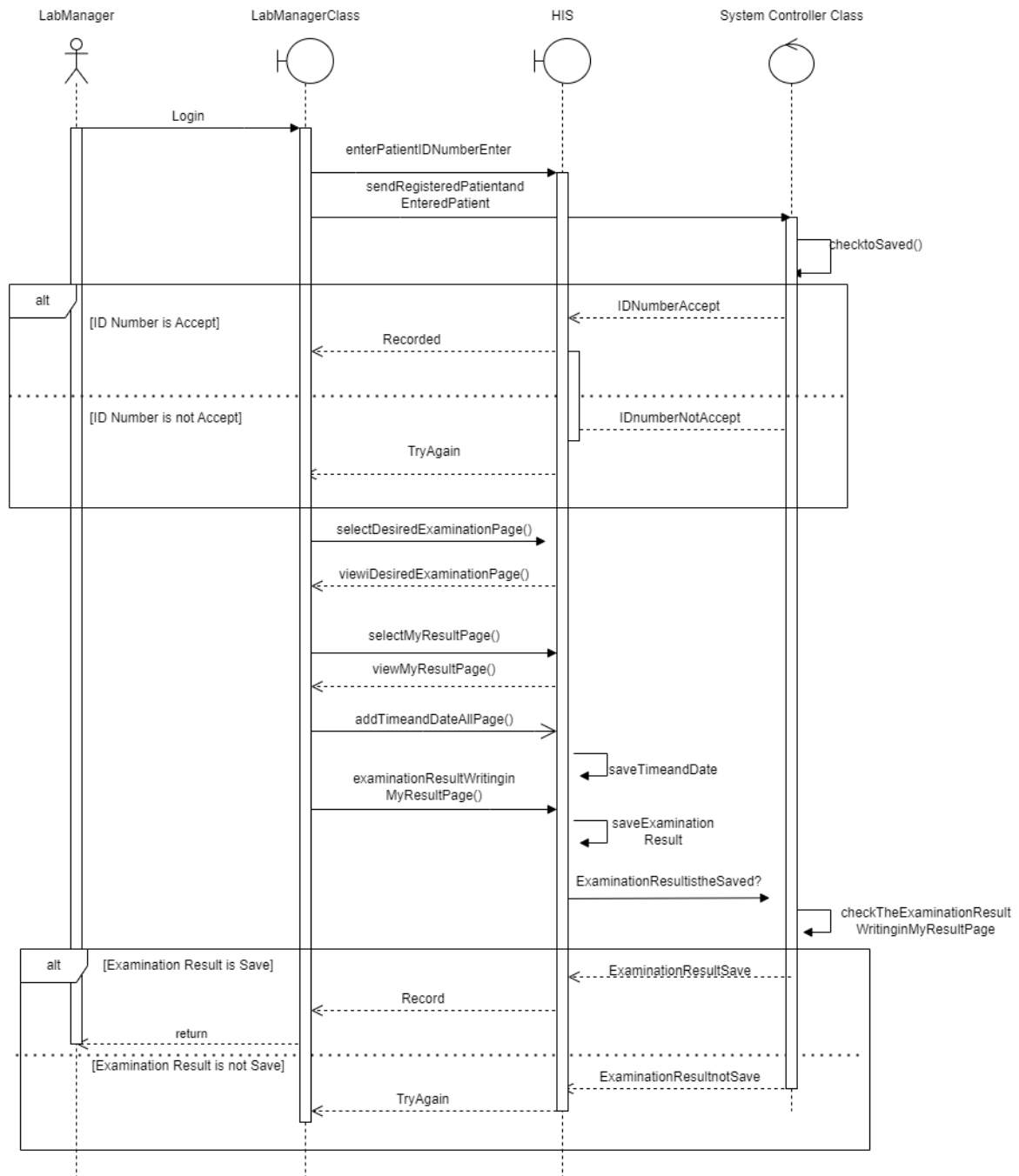


# SEQUENCE DIAGRAM OF THE USER'S DOCTOR SCENARIO





## SEQUENCE DIAGRAM OF THE USER'S LABMANAGER SCENARIO



# SOFTWARE PROJECT MANAGEMENT PLAN (IEEE)

Software project management control document. A software project management plan establishes the policies, procedures, rules, tasks, schedules, and resources needed to complete the project.

## 1. Overview

### 1.1. Project Summary

#### 1.1.1. Purpose, Scope and Objectives

The objective of this project is to develop a hospital information system that patients and hospital employees can use. Patients can make an appointment from the application, information about the patient's diagnosis process can be added to the application.

#### 1.1.2. Assumptions and Constraints

Information security must be ensured.

The deadline must be met.

The budget constraint must be met.

The product must be reliable.

The app must meet the needs of users.

The application must comply with hospital policy.

#### 1.1.3. Project Deliverables

The whole project will be completed and delivered within 3 months, including the test phase.

#### 1.1.4. Schedule and Budget Summary

Requirements workflow – 4 team members – 4 days – 0\$

Analysis workflow – 3 team members – 7 days – 0\$

Design workflow – 3 team members – 5 days – 0\$

Implementation workflow – 4 team members – 4 weeks – 0\$

Testing workflow – 4 team members – 3 weeks (Expandable) – 0\$

## **1.2.Reference Materials**

- IEEE Standard for Software Configuration Management Plans, ANSI/IEEE Std. 828-199.
- IEEE Standard for Software Project Management ANSI/IEEE Std.1058.1-1987.
- IEEE Standard for Developing Software Life Cycle Processes, ANSI/IEEE Std. 1074-1991.
- Test Plan by project team.

## **1.3.Evolution of the SPMP**

Changes to the project at the request of the client will take place if they are discussed by all group members and approved by a majority vote. If the proposed change is to be approved or rejected, the effect on the other modules of the project and the project as a whole is kept. The effects of the change are indicated to the customer and approval is obtained.

## **1.4.Definitions and Acronyms**

HIS – Hospital Information System

SPMP – Software Project Management Plan

CRC - Class Responsibility Collaborator

## **2. Project Organization**

### **2.1.Roles**

Elif İrem – Analysis workflow

Hilal – Requirements workflow

Gizem – Requirements workflow

Berfin – Design workflow

### **2.2.Responsibilities**

Elif prepares use-case, use-case diagrams, test plan and SPMP

Hilal prepares class diagrams, CRC cards, statechart diagram and the dictionary

Gizem determines requirements for system and prepares sequence diagrams and interface design

Berfin determines application domain and class data details, prepares pseudo codes.

### **3. Managerial Process Plan**

#### **3.1. Project Team Training Plan**

No additional student training is needed for this Project.

#### **3.2. Control Plan**

The periods of the project will be followed by the project team members and the client. On a regular basis, the client will be interviewed about the client's requests and suggestions regarding the program. Each member of the group will inform others about the developments in the project and get their opinions. It will be checked whether the project is in parallel with the plan and the approval of the customer will be obtained at each stage.

#### **3.3. Staffing Plan**

Four group members and the client are the staff of the project. Group members will develop the project and the customer will monitor each stage.

#### **3.4. Risk Plan**

- Deliver on time - Project group members will follow the given plan and schedule; meet regularly to report on developments
- The user has no experience using the computer - The user interface will be designed simple considering that users do not have computer experience, the program will be usable.
- Lack of communication, causing lack of clarity and confusion - Project members who do not comply with the schedule without an excuse will be sanctioned.
- New direction, policy, or statute – regression tests will be performed to see if the application is compatible with changes

#### **3.5. Testing Plan**

Test Plan was prepared by Elif İrem and included in the project documents

## **4. Technical Process Plans**

### **4.1.Methods, Tools and Techniques**

Documentation and coding will be done in IEEE standards. Object-oriented design and object-oriented programming will be used. The source code will be written in the JAVA.

### **4.2.Infrastructure Plan**

The workflows have been performed with the Unified Process. Project documents will be prepared using diagrams.net and the Office Word.

### **4.3.Product Acceptance Plan**

The product development process will be approved by negotiating with the customer during the production process. Product acceptance will be made with the customer at the deadline.

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

It's like a rule book to follow. Important aspects such as test estimation, test scope, Test Strategy are documented in the Test Plan so it can be reviewed by the Management Team and reused for other projects.

### **1. Introduction**

#### **I. System Overview**

The Hospital Information System is an easy-to-use program that guarantees information security to its users; where patients can make appointments; doctors view the results of laboratory examinations, enter diagnoses and prescriptions.

#### **II. Test Objectives**

- The system integration test of the Hospital Information System should validate from both the requirements perspective and business perspective .
- The system works well for doctor, laboratory and patient
- Admin has access to registrations, appointments made and cancellation of appointments

## **2. Introduction**

### **III. System Overview**

The Hospital Information System is an easy-to-use program that guarantees information security to its users; where patients can make appointments; doctors view the results of laboratory examinations, enter diagnoses and prescriptions.

### **IV. Test Objectives**

- The system integration test of the Hospital Information System should validate from both the requirements perspective and business perspective .
- The system works well for doctor, laboratory and patient
- Admin has access to registrations, appointments made and cancellation of appointments

## **3. Test Items**

- Appointment
- Examination
- Diagnosis
- Prescription

## **4. Approach**

### **I. Assumptions**

- The first build of the Hospital Information System will be ready for system integration testing on January 1, XXXX.

### **II. Constraints**

- Three weeks might not be enough time to test the entire system and then retest the system to find new defects due to fixes.

### **III. Test Type**

- Functional testing, by performing test cases based on testable requirements.
- Security testing, by testing each end-user's security access levels.
- Controls testing, by testing appointment controls
- Compliance testing, by evaluating system performance against the hospital policies and procedures
- Regression testing, to ensure that a change to the system does not introduce new defects.

## **5. Item Pass/Fail Criteria**

To pass the system integration test, the following criteria must be met:

- The registration function should work correctly.
- The function of logging all users who will use the system should work correctly.
- There should be no information leakage between users.
- The function of choosing a date for the appointment should work correctly
- The function of selecting the time for the appointment should work correctly
- The examination selection function should work correctly
- Verification code functions registration must work correctly to fraudulent registration
- There should be no problem with the examination entry
- There should be no problem with the diagnosis entry
- There should be no problem with the prescription entry

## **6. Test Environment**

### **I. Hardware**

- Asus X415EA-EB573  
Intel Core i5-1135G7  
8GB DDR3 RAM
- iMAC 1.4GHz  
dual-core Intel Core i5 processor with 3MB shared L3 cache  
8GB of 1600MHz LPDDR3 onboard memory

### **II. Software**

- The Hospital Information System application software
- Windows 10 operating system
- Windows 11 operating system

## **7. Responsibilities**

- Hilal - perform functional and control tests
- Berfin – perform compliance tests
- Elif İrem - editing the result and preparing an error fixing plan for the errors found
- Gizem - confirm test results

## 8. Schedule

- 1 January XXXX - system integration testing process will be started
- 22 January XXXX - integration test results will be reported
- 5 February XXXX - test results are checked and confirmed.

## INFORMATION HIDING

Information hiding is a technique used in computer programming to hide certain details of an implementation from the user. It allows developers to design and implement complex systems in a way that makes them easier to understand and use by separating the interface, which is the part that the user interacts with, from the implementation, which is the underlying code that makes the system work.

There are several ways to achieve information hiding in programming, including encapsulation, abstraction, and modularization.

Information hiding is important because it allows developers to design and implement complex systems without exposing the details of their inner workings to the user. This makes it easier for users to understand and use the system, and it also helps to ensure that the system is more robust and maintainable because changes to the implementation are less likely to break the interface.

## PSEUDO CODE

Rough code (pseudo code) is used during the design phase before using the software language for a software project. Rough code is a way of describing how the software will work, what actions to take, and what data to use. The rough code is used to create a "template" for writing the actual code of the software. While writing rough code, syntax and structure suitable for the scripting language are used, but not translated into a real scripting language. Therefore, brute code is usually easier to read and understand. Also, when rough code is written, thought and design is done about how the software will work, which saves time and effort during actual code writing. Rough code writing is done during the design phase of the software project and is done before the actual code writing. In this way, time and effort are saved during the actual code writing, and the software is written in a more organized and understandable way.



## PSEUDO-CODES OF THE PROJECT

All access levels are set to private due to “Information Hiding”. Objects can be accessed by applying the get and set methods.

### USER CLASS

```
import java.util.*;

public class UserClass {

    public userClass() { //DEFAULT CONSTRUCTURE}

    private string UserName;

    private string UserPassword;

    private int HospitalNumber;

    public void SelectTypeOfUser() { // TODO code here}

    public void login() { // TODO code here}

    public void exit() { // TODO code here}

    public string getUsername() {

        return UserName;

    }

    Public void setUsername(int userName) {

        UserName = userName;

    }

    public string getPassword() {

        return UserPassword;

    }

    Public void setPassword(int userPassword) {

        UserPassword = userPassword;

    }

}
```

```

    public int getHospitalNumber() {
        returnHospitalNumber;
    }

    Public void setHospitalNumber(int hospitalNumber) {
        HospitalNumber = hospitalNumber;
    }
}

```

## **PATIENT ENROLL CLASS**

```

import java.util.*;

public class PatientEnrollClass {
    public patientEnrollClass() { //DEFAULT CONSTRUCTURE}
    private string PatientName;
    private int PatientID;
    private int PatientNumber;
    public void selectTypeOfEnroll() { // TODO code here}
    public void getPassword() { // TODO code here}
    public void enrollTheHIS() { // TODO code here}
    public string getPatientName() {
        return PatientName;
    }
    Public void setPatientName(int patientName) {
        PatientName = patientName;
    }
    public int getPatientID() {
        return PatientID;
    }
}

```

```

    Public void setPatientID(int patientID) {
        PatientID = patientID;
    }
    public int getPatientNumber() {
        return PatientNumber;
    }
    Public void setPatientNumber(int patientNumber) {
        PatientNumber = patientNumber;
    }
}

```

## **MEDICAL PERSONNEL ENROLL CLASS**

```

import java.util.*;

public class MedicalPersonnelEnrollClass {
    public medicalPersonnelEnrollClass() { //DEFAULT CONSTRUCTURE}
    private string MedPerName;
    private int MedPerID;
    private int MedPerNumber;
    public void selectTypeOfEnroll() { // TODO code here}
    public void getPassword() { // TODO code here}
    public void enrollTheHIS() { // TODO code here}
    public string getMedPerName() {
        return MedPerName;
    }

    Public void setMedPerName(int medPerName) {
        MedPerName = medPerName;
    }
}

```

```

public int getMedPerID () {
    return MedPerID;
}

Public void setMedPerID(int medPerID) {
    MedPerID= medPerID;
}

public int getMedPerNumber () {
    return MedPerNumber;
}

Public void setMedPerNumber(int medPerNumber) {
    MedPerNumber = medPerNumber;
}
}

```

## APPOINTMENT CLASS

```

import java.util.*;

public class AppointmentClass {

    public appointmentClass() { //DEFAULT CONSTRUCTURE}

    private int Hour;

    private int Minute;

    private int Date;

    public void viewPolyclinic() { // TODO code here}

    public void selectPolyclinic() { // TODO code here}

    public void selectDoctor() { // TODO code here}

    public void selectDate() { // TODO code here}

    public void selectTime() { // TODO code here}

    public void saveInMyAppointmentPage() { // TODO code here}
}

```

```

    public int getHour () {
        return Hour;
    }
    Public void setHour(int hour) {
        Hour = hour;
    }
    public int getMinute () {
        return Minute;
    }
    Public void setMinute (int minute) {
        Minute = minute;
    }
    public int getDate () {
        return Date;
    }
    Public void setDate (intdate) {
        Date = date;
    }
}

```

## REQUEST CONTROLLER CLASS

```

import java.util.*;

public class RequestControllerClass {

    public requestControllerClass() { //DEFAULT CONSTRUCTURE}

    private bool IsTrue;

    private int Hour;

    private int Minute;

```

```

private int Date;
private bool IsAccomplished;
private int ConfirmationCode;
public void sendCode() { // TODO code here }
public boolean checkTheSendingCode() { // TODO code here }
public void startTimer() { // TODO code here }
public boolean checkTheEnroll() { // TODO code here }
public void acceptTheEnrollment() { // TODO code here }
    public boolean checkTheLogin() { // TODO code here }
public void acceptTheLogin() { // TODO code here }
public void sendMessageForTheWrongInformation() { // TODO code
here }
public void fixWrongInformationInLoginAndEnroll() { // TODO code
here }
public bool getIsTrue () {
    return IsTrue;
}
Public void setIsTrue(int isTrue) {
    IsTrue = isTrue;
}
public int getHour () {
    return Hour;
}
Public void setHour(int hour) {
    Hour = hour;
}

```

```

    public int getMinute () {
        return Minute;
    }
    Public void setMinute(int minute) {
        Minute = minute;
    }
    public int getDate () {
        return Date;
    }
    Public void setDate (intdate) {
        Date = date;
    }
    public bool getIsAccomplished () {
        return IsAccomplished;
    }
    Public void setIsAccomplished (intisAccomplished) {
        IsAccomplished = isAccomplished;
    }
    public int getConfirmationCode () {
        return ConfirmationCode;
    }
    Public void setConfirmationCode (intconfirmationCode) {
        ConfirmationCode = confirmationCode;
    }
}

```

## APPOINTMENT CANCELLATION CLASS

```
import java.util.*;

public class AppointmentCancellationClass {

    public appointmentCancellationClass() {//DEFAULT
    CONSTRUCTURE}

    public void selectTheMyAppointmentPage() {// TODO code here}

    public void selectMadeAppointmetInMyAppointment() {// TODO code
    here}

    public void doingAppointmentCancellation() {// TODO code here}

}
```

## DOCTOR CLASS

```
import java.util.*;

public class DoctorClass {

    public doctorClass() {//DEFAULT CONSTRUCTURE}

    private int PatientID;

    private int Hour;

    private int Minute;

    private int Date;

    Public void selectExamination() {// TODO code here}

    Public void reviewExaminationResults() {// TODO code here}

    Public void addTimeAndDateinAllPage() {// TODO code here}

    Public void diagnosisWritinginMyDiseasesPage() {// TODO code here}

    Public void prescriptionWritingInMyPrescriptionPage() {// TODO code
    here}

    Public void save()

    public int getPatientID() {

        return PatientID;

    }

}
```



```

    Public void setPatientID(int patientID) {
        PatientID = patientID;
    }
    public int getHour () {
        return Hour;
    }
    Public void setHour(int hour) {
        Hour = hour;
    }
    public int getMinute () {
        return Minute;
    }
    Public void setMinute(int minute) {
        Minute = minute;
    }
    public int getDate () {
        return Date;
    }
    Public void setDate(intdate) {
        Date = date;
    }
}

```

## **LABMANAGER CLASS**

```

import java.util.*;

public class LabManagerClass {
    public laboratoryManageClass() { //DEFAULT CONSTRUCTURE}
    private int PatientID;

```

```
private int Hour;
private int Minute;
private int Date;
public void viewDesiredExamination() { // TODO code here }
public void addTimeAndDateinAllPage() { // TODO code here }
public void examinationResultWritingInMyResultsPage() { // TODO code here }
public void save() { // TODO code here }
```

```
public int getPatientID() {
    return PatientID;
}
public void setPatientID(int patientID) {
    PatientID = patientID;
}
public int getHour () {
    return Hour;
}
public void setHour(int hour) {
    Hour = hour;
}
```

```
public int getMinute () {
    return Minute;
}
public void setMinute(int minute) {
    Minute = minute;
}
```

```

    public int getDate() {
        return Date;
    }

    public void setDate(intdate) {
        Date = date;
    }
}

```

## SYSTEM CONTROLLER CLASS

```

import java.util.*;

public class SystemControllerClass {

    public systemControllerClass() { //DEFAULT CONSTRUCTURE}

    private bool IsAccomplished;

    private bool IsTrue;

    public boolean checkTheAllList() { // TODO code here}

    public boolean checkTheAllSet() { // TODO code here}

    public boolean checkTheAppointmentAcception() { // TODO code here}

    public boolean checkTheAllDeleting() { // TODO code here}

    public boolean checkTheAppointmentsInMyAppointmentPage() { //
    TODO code here}

    public boolean checkTheExaminationResultWritingInMyResultsPage()
    { // TODO code here}

    public boolean checkTheDiagnosisWritingInMyDiseasesPage() { // TODO
    code here}

    public boolean checkThePrescriptionWritingInMyPrescriptionPage() { //
    TODO code here}

    public boolean checkTheAllSendingMessageWasRecorded() { // TODO
    code here}

    public boolean checkTheAllSendingMessageForTheWrongInformation()
    { // TODO code here}
}

```

```

public boolean checkTheAllSaved() { // TODO code here }
public boolean checkTheTimeAndDate() { // TODO code here }
public bool getIsAccomplished() {
    return IsAccomplished;
}
Public void setIsAccomplished(int isAccomplished) {
    IsAccomplished = isAccomplished;
}
public bool getIsTrue() {
    return IsTrue;
}
Public void setIsTrue(int isTrue) {
    IsTrue = isTrue;
}
}

```

### ADMIN (HIS) CLASS

```

import java.util.*;
public class AdminClass {
    public adminClass() { //DEFAULT CONSTRUCTURE}
    private int Hour;
    private int Minute;
    private int Date;
    private int MedicalPersonnelHospitalNumber;
    private int MedicalPersonnelPassword;
    private int PatientPassword;
    private string Polyclinic;
    private string Examination;
}

```

```
private string PatientButton;
private string MedicalPersonnelButton;
private string ExitButton;
private string Enroll;
private string Message;
private string MyDiseases;
private string MyResults;
private string MyPrescriptionstring;
private string MyAppointment;
private string Save;
public void deleteTheUser() { // TODO code here }
public void setTheHospitalNumber() { // TODO code here }
public void setPasswordAllUsers() { // TODO code here }
public void setTheDateAndTimeDisplay() { // TODO code here }
public void listPolyclinic() { // TODO code here }
public void listExamination() { // TODO code here }
public void listDoctorsNames() { // TODO code here }
public void setTheButtons() { // TODO code here }
public void setTheAllPage() { // TODO code here }
public void sendAmessageWasRecorded() { // TODO code here }
public void acceptTheAllSaved() { // TODO code here }
public void deleteAppointment() { // TODO code here }
public void acceptTheAllExit() { // TODO code here }
public int getHour () {
    return Hour;
}
```

```
Public void setHour(int hour) {  
    Hour = hour;  
}  
public int getMinute () {  
    return Minute;  
}  
Public void setMinute(int minute) {  
    Minute = minute;  
}  
public int getDate () {  
    return Date;  
}  
Public void setDate (int date) {  
    Date = date;  
}  
public int getMedicalPersonnelHospitalNumber () {  
    return MedicalPersonnelHospitalNumber;  
}  
Public void setMedicalPersonnelHospitalNumber (int  
medicalPersonnelHospitalNumber) {  
    MedicalPersonnelHospitalNumber =  
    medicalPersonnelHospitalNumber;  
}  
public int getMedicalPersonnelPassword () {  
    return MedicalPersonnelPassword;  
}
```

```
Public void setMedicalPersonnelPassword (int
medicalPersonnelPassword) {
    MedicalPersonnelPassword = medicalPersonnelPassword;
}
public int getPatientPassword () {
    return PatientPassword;
}
Public void setPatientPassword (int patientPassword) {
    PatientPassword = patientPassword;
}
public string getPolyclinic () {
    return Polyclinic;
}
Public void setPolyclinic(int polyclinic) {
    Polyclinic = polyclinic;
}
public string getExamination () {
    return Examination;
}

Public void setExamination(int examination) {
    Examination = examination;
}
public string getPatientButton() {
    return PatientButton;
}
```

```
Public void setPatientButton(int patientButton) {  
    PatientButton = patientButton;  
}  
public string getMedicalPersonnelButton() {  
    return MedicalPersonnelButton;  
}  
Public void setMedicalPersonnelButton (int medicalPersonnelButton) {  
    MedicalPersonnelButton = medicalPersonnelButton;  
}  
public string getExitButton() {  
    return ExitButton;  
}  
Public void setExitButton (int exitButton) {  
    ExitButton = exitButton;  
}  
public string getEnroll () {  
    return Enroll;  
}  
Public void setEnroll(int enroll) {  
    Enroll = enroll;  
}  
public int getMessage () {  
    return Message;  
}  
Public void setMessage (int message) {  
    Message = message;  
}
```



```
public int getMyDiseases () {  
    return MyDiseases;  
}  
Public void setMyDiseases (int myDiseases) {  
    MyDiseases = myDiseases;  
}  
public int getMyResults() {  
    return MyResults;  
}  
Public void setMyResults (int myResults) {  
    MyResults = myResults;  
}  
public int get MyPrescriptionstring () {  
    return MyPrescriptionstring;  
}  
Public void set MyPrescriptionstring (int myPrescriptionstring) {  
    MyPrescriptionstring = myPrescriptionstring;  
}  
public int getMyAppointment () {  
    return MyAppointment;  
}  
Public void setMyAppointment (int myAppointment) {  
    MyAppointment = myAppointment;  
}
```

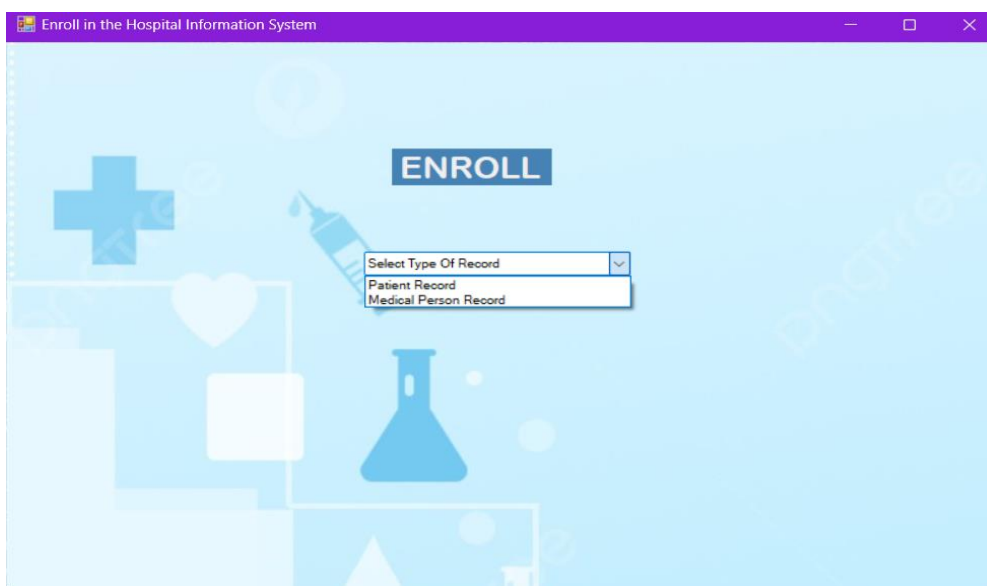
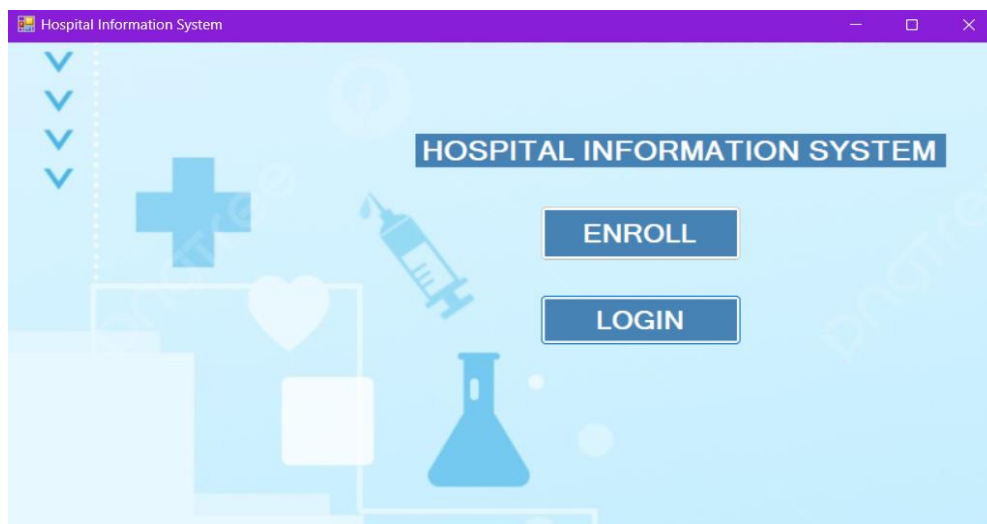
```

public int getSave () {
    return Save;
}

Public void setSave (int save) {
    Save = save;
}
}

```

## FORM INTERFACES



Enroll in the Hospital Information System

## ENROLL

Patient Enroll

Name

IDNumber

Phone

Password

Enroll The HIS

Enroll in the Hospital Information System

## ENROLL

Medical Person Enroll

Name

HospitalNumber

Phone

Password

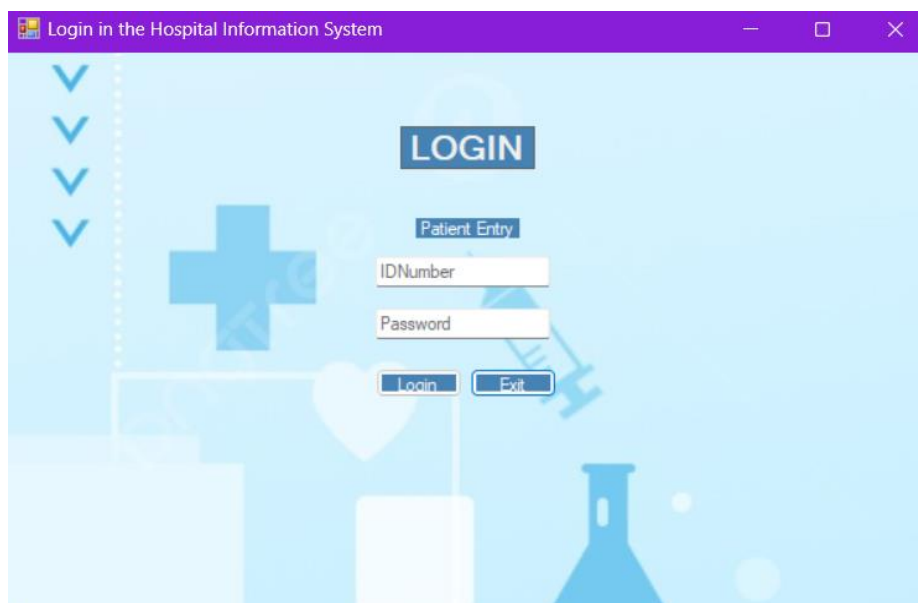
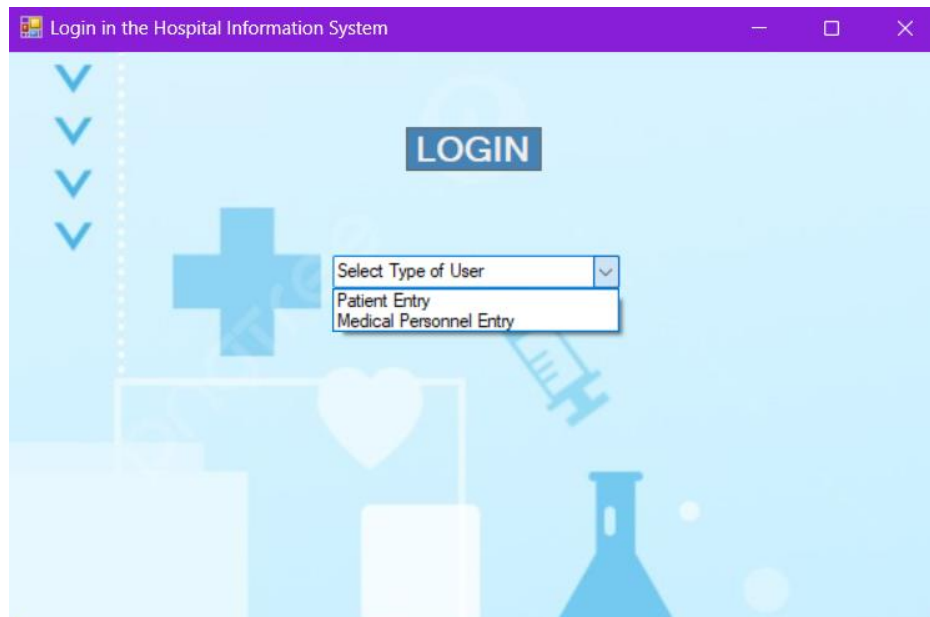
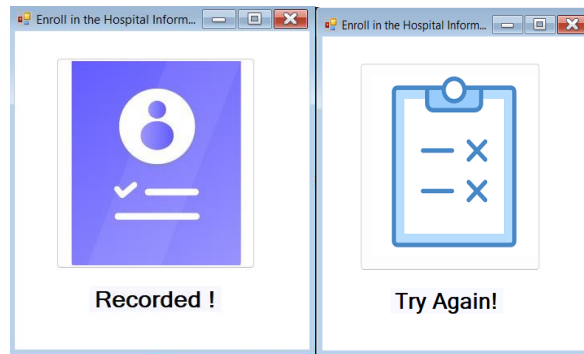
Enroll The HIS

Enroll in the Hospital Information System

## LOGIN CONFIRMATION CODE

exampleAB1234

☒ Confirmation Code



Login in the Hospital Information System

LOGIN

Medical Personnel Entry

HospitalNumber

Password

Login Exit

Patient Portal



NAME  
PHONE

Make Appointment


Interrogate/Delete Appointment

Lab Results

Diagnosis

Prescription

Medical Personnel Portal



NAME  
PHONE NUMBER

Patient ID number Enter

Examination Select

Desired Examination

Examination Result Enter

Diagnosis Enter

Prescription Enter