MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Summary

The Laboratory Tests Management System

Group 6	
Group members	Bạch Minh Đức – SE61791 Châu Minh Ý– SE61419 Trần Tuấn Dũng – SE61982 Nguyễn Hữu Lâm – SE61234
Supervisor	Hồ Hoàn Kiếm
Ext. Supervisor	N/A
Capstone Project Code	eLTMS

- Ho Chi Minh City, 14 September 2018 -

Table of Contents

List of Tables

List of Figures

I. Capstone Project Register

TRƯỜNG ĐẠI HỌC FPT

CAPSTONE PROJECT REGISTER

Class: Duration time: from 09/2018.... To 11 /2018.....

(*) Profession: <Software Engineer> Specialty: <ES> ☐ <IS> ☒

(*) Kinds of person make registers: Lecturer ☒ Students ☐

1. Register information for supervisor (if have)

	Full name	Phone	E-Mail	Title
Supervisor 1	Hồ Hoàn Kiếm		kiemhh@fe.edu.vn	Mr.

2. Register information for students (if have)

	Full name	Student code	Phone	E-mail	Role in Group
Student 1	Huỳnh Tịnh	SE62713	0917351265	tinhhse62713@fpt.edu.vn	Leader
Student 2	Hà Trung Kiên	SE61993	01698162578	kienhtse61993@fpt.edu.vn	Member
Student 3	Nguyễn Huy Hoàng	SE61758	01283383166	nguyenphpse62075@fpt.edu.vn	Member
Student 4	Phạm Hoàng Phúc Nguyên	SE62075	0901246005	hoangnhse61758@fpt.edu.vn	Member

3. Register content of Capstone Project

(*) 3.1. Capstone Project name:

English: Building the bids online system

Vietnamese: Xây dựng hệ thống đấu giá trực tuyến

Abbreviation:

- eBOS

Building the bids online system combining mobile devices and web application provides following services:

+ Post product via mobile and web application

+ Product management

+ Bids transaction management

+ Bidders management

+ Notify automatically

Simulation

- Building the system with at least three categories of products such as technical products, appliance products, art products, ...

(*) 3.2. Main proposal content (including result and product)

a) Theory and practice (document):

- . Student should apply the software development process and the UML

- . Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages

- . 3 tiers should be applied

- . Server-side technique:

  - Database design, OOA, OOD, OOP, MVC, .Net , Java technology and so on

- . Client-side technique

  - HTML5, CSS, JavaScript, JQuery, Ajax, Angular, React Native, Ionic, Xamarin and so on

- . Communication technique

  - Exchange information and transfer data effective in multiple application by web services and notify based on cloud computing

- . Research

  - Algorithms, frameworks and cloud computing

b) Program:

- . Mobile application functions for Auctioneers and Bidders

  - Auctioneer and Bidders register on mobile or web application with business information (full name, position, main phone, other-info**)**
  - Registered information is written to database server from mobile application via RESTful web services
  - Bidders can bid, view product list, … via mobile / web application
  - Managed feedbacks from Auctioneers and Bidders

- Auctioneers and Bidders receive inform messages about rules and conditions when bids.
- Report

- . Web application functions
  - Make a bid
  - Auctioneer management
  - Product management
  - Notify messages for Auctioneers and Bidders
  - Bidders management
  - Report

- . Data service application
  - Provide data services for mobile application and web application
  - Perform data security.

c) Other products:

- . All of management functions of the system must be implemented to support the operating system in best

4. Other comment (propose all relative thing if have)

N/A

Supervisor (If have)
(Sign and full name)

HCM city, date /09/2018
On behalf of Registers
(Sign and full name)

II. Report 1

1. Project Information

- Project name: **Building the laboratory tests management system**
- Project Code: **eLTMS**
- Product Type: **Website & Mobile Application**
- Start Date: **September 10th, 2018**
- End Date: **December 14th, 2018**

2. Introduction

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the

hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus, keeping the working of the manual system as the basis of our project. We have developed "The Laboratory Tests Management System".

The main objective of this project is to analyses the concept and process of laboratory tests in the hospital, from which to build up the website to manage all patient information of laboratory tests and help doctor, users, and laboratory technician easy to control their job. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and provides robust & reliable storage and backup facilities.

3. Current Situation

Nowadays, In Vietnam, Except for the large hospitals on the upline have laboratory system, which help doctor conjecture and treat patients in the best way. Other hospital or clinics on the downline do not have this system, If the doctor want to conjecture and treat patients in the best way, they must require patients go to hospital on upline and check their health on the laboratory systems, after that, patient will come back and listen the diagnosis.

But all of hospitals on upline is over-load now. If the patients want to check health, they must take time consuming to change hospital, take time consuming to wait because large numbers overload of patients. Not only time but also money, they must spend money for this moving and laboratory system.

With our laboratory management system, we will help patients and doctor with life easier.

4. Problem Definition

Below are disadvantages of current situation:

- Patients spend too much time, effort to get the health result.
- Doctor do not have the best support tools for offering the best treatment options.
- Hospitals, clinics have difficulty in managing records, reviews.

5. Proposed Solution

Our solution is to build systems capable of placing and managing time (Calendar Manager) to resolve the current situations, we also design the system to be scalable, so we can deploy this system to a multiple insurance services company in future.

eLTMS includes web application and mobile applications with following functions:

1.1. Feature functions

- Web application:
  - Register Appointment: User can register a new account on website. A staff will reply by SMS message for authentication to confirm calendar management
- Mobile application:
  - Register Appointment: User can register a new account on mobile. A staff will reply by SMS message for authentication to confirm calendar management.

1.2. Advantages and disadvantages

- Advantages:
  - Patient is easier in manage time for appointment.
  - Hospitals/Clinics can control facilities, machines and supplies.
  - Hospitals can record the types of illnesses, disease trends, which can be used for deciding how to treat, how to disease prevention.
  - Hospitals/Clinics can maximize the free time of each laboratory.
  - High Automation.
- Disadvantages:
  - Human must still interfere with system in some cases.

6. Functional Requirements

Function requirements of the system are listed as below:

- Base Component:
  - Login.
  - Register.
- User (Patient) component:
  - New account request / Feedback about account request.
  - Book Appointment/ View Booked Appointment/Edit-Cancel Appointment.
  - View test result-View suggestion, note.
  - Edit, Update profile account.
- Receptionist component:
  - View detail appointment.
  - Change status of appointment.
  - View list patient who has testing following day.
- Cashier component:
  - New account creates for user (Patient).

    - View list patient who has testing following day
    - Manage Laboratory (Operate with patient by appropriate/real time)
    - Change status of appointment.
- Nurse component:
    - Change status of appointment.
    - Change status of lab-testing.
- Warehouse-keeper component:
    - Manage items on system.
    - Import and control quantity on the system.
    - Stock control, stock out, stock in
- Doctor component:
    - Check the disease.
    - Suggest Medication (types and quantities)
    - Some good advice for patient.
    - Write note and conclusion
    - Request re-testing
- Laboratory Technician component:
    - Import the test result in the system.
    - Mapping code of specimen.
    - Create New Group of Specimen.
    - Create New Type of Specimen.
- Manager component:
    - Manage Service of Lab Test.
    - Manage Staff account (Add, Edit and Delete).
    - View and Check Result Test.
    - View Feedback.

7. Roles and Responsibility

No	Full Name	Role	Position	Contact
1	Hồ Hoàn Kiếm	Project Manager	Supervisor	Kiemhh@fpt.edu.vn
2	Bạch Minh Đức	Developer	Leader	Ducbmse61791@fpt.edu.vn
3	Châu Minh Ý	Developer	Member	Ycmse61419@fpt.edu.vn
4	Trần Tuấn Dũng	Developer	Member	Dungttse61982@fpt.edu.vn
5	Nguyễn Hữu Lâm	Developer	Member	Lamnhse61234@fpt.edu.vn

Table 1 - Roles and Responsibility

III. Software Process Model

In this project, we choose Scrum which is an agile framework that allows step by step to set up a plan successfully. We choose this model because of the following reasons:

- Time-saving: daily meetings ensure that the process is at correct stage, as established at the beginning of the project
- Easy to use: suitable model for small and medium project
- Fast response to changes: product owner may change requirements or extend / reduce scope and we can adapt better
- Encourage teamwork: roles and tasks are divided and assigned efficiently

Figure 1 - Scrum Model

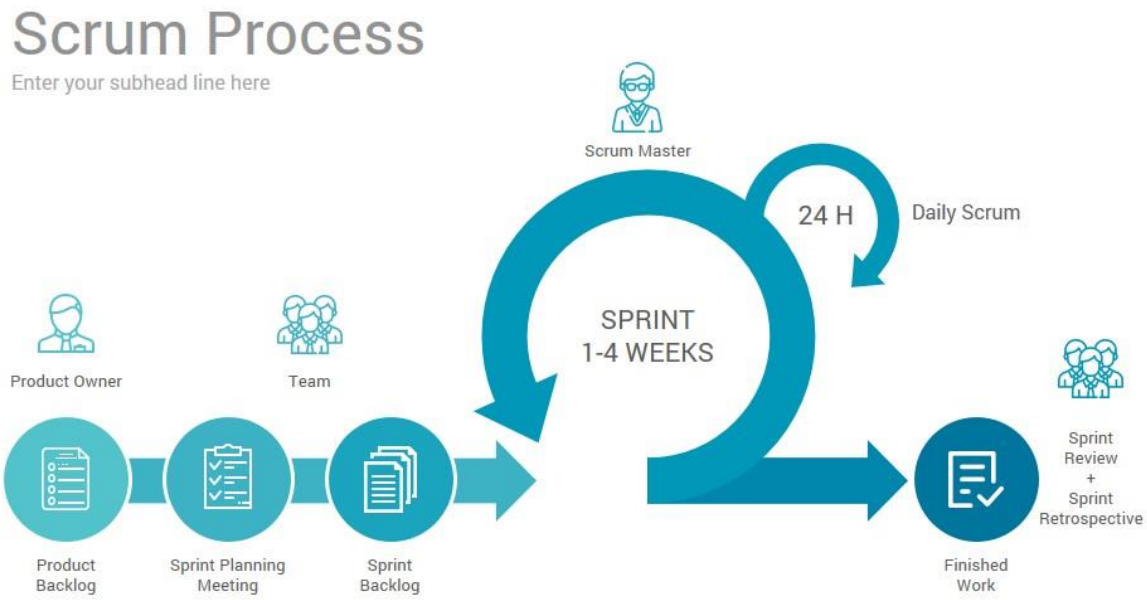


Figure 2 - Scrum Model

IV. Conceptual Diagram

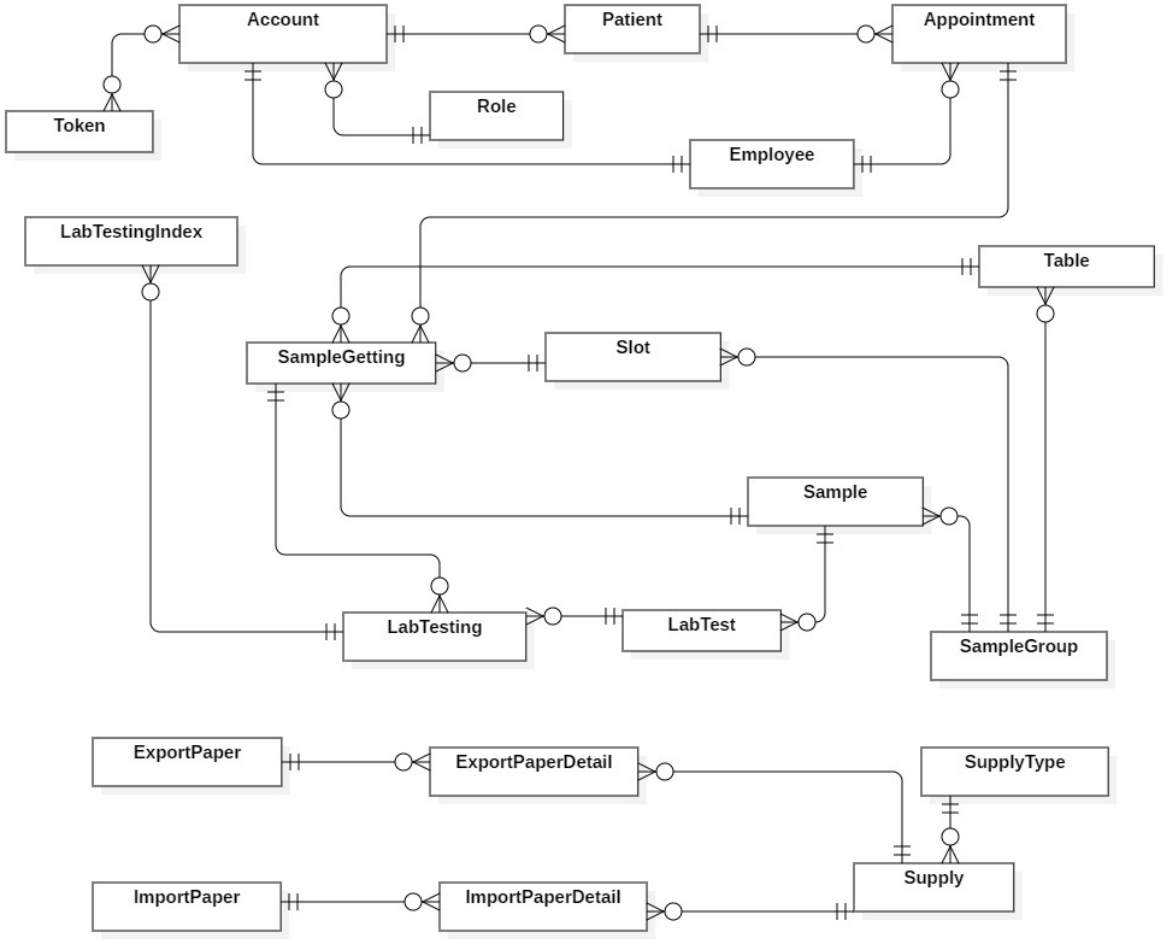


Figure 3 - Conceptual Diagram

Data Dictionary

Entity Data dictionary: describe all content of all entities	
Entity Name	**Description**
Account	Contain account information
Appointment	Contain Appointment information
Employee	Contain staff's information
ExportPaper	Contain Export Paper information
ExportPaperDetail	Contain Export Paper Detail information
ImportPaper	Contain information about Import Paper
ImportPaperDetail	Contain Import Paper Detail information
LabTest	Contain Lab Test information

LabTesting	Contain information about Lab Testing
LabTestingIndex	Contain transaction information of Lab Testing Index
Patient	Contain Patient information
Role	Contain Role information
Sample	Contain Sample information
SampleGetting	Contain Sample Getting information
SampleGroup	Contain Sample Group information
Slot	Contain Slot information
Supply	Contain Supply information
SupplyType	Contain Supply Type information
Table	Contain Table information
Token	Contain Token information

Table 2 - Conceptual diagram data dictionary

V. System Overview Use Case

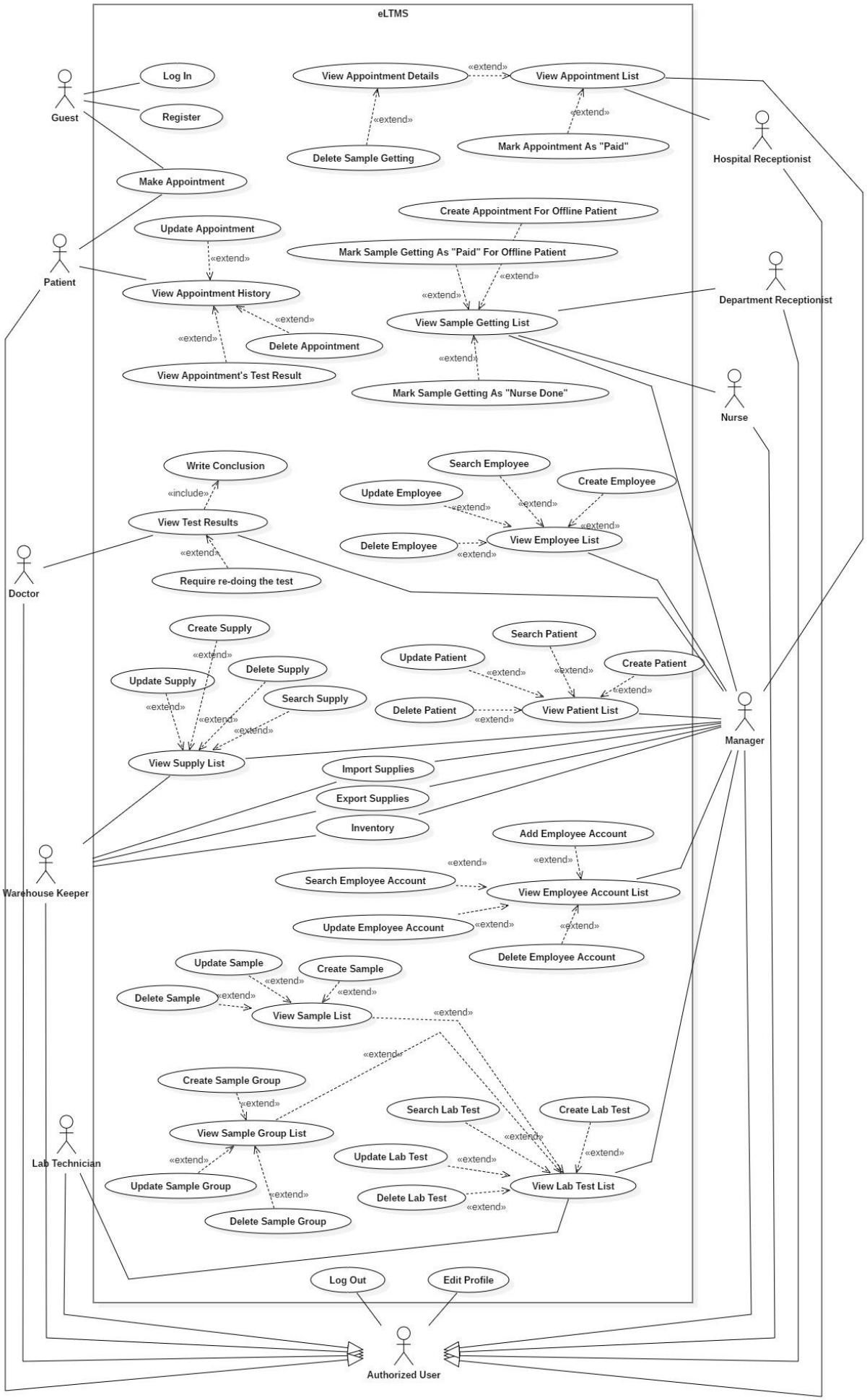

eLTMS
Guest
Log In
Register
Make Appointment
Patient
Update Appointment
View Appointment History
Delete Appointment
View Appointment's Test Result
View Appointment Details
View Appointment List
Delete Sample Getting
Mark Appointment As "Paid"
Hospital Receptionist
Create Appointment For Offline Patient
Mark Sample Getting As "Paid" For Offline Patient
View Sample Getting List
Mark Sample Getting As "Nurse Done"
Department Receptionist
Nurse
Write Conclusion
View Test Results
Require re-doing the test
Doctor
Search Employee
Create Employee
Update Employee
Delete Employee
View Employee List
Create Supply
Update Supply
Delete Supply
Search Supply
View Supply List
Search Patient
Update Patient
Create Patient
Delete Patient
View Patient List
Manager
Import Supplies
Export Supplies
Inventory
Warehouse Keeper
Add Employee Account
Search Employee Account
View Employee Account List
Update Employee Account
Delete Employee Account
Update Sample
Create Sample
Delete Sample
View Sample List
Create Sample Group
View Sample Group List
Update Sample Group
Delete Sample Group
Lab Technician
Search Lab Test
Create Lab Test
Update Lab Test
Delete Lab Test
View Lab Test List
Log Out
Edit Profile
Authorized User
«extend»
«include»

Figure 4 - System Overview Use Case

VI. List of Core Flow Use Case

1. Guest Overview Use Case

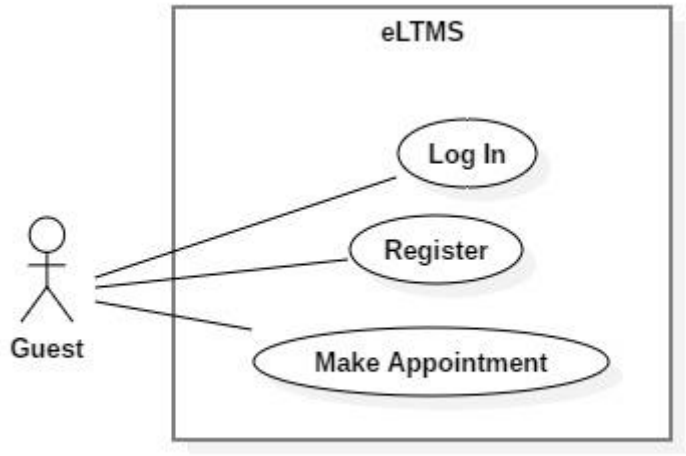


Figure 5 - <Guest>Overview Use Case

1.1. <Guest> Login

Use Case Diagram

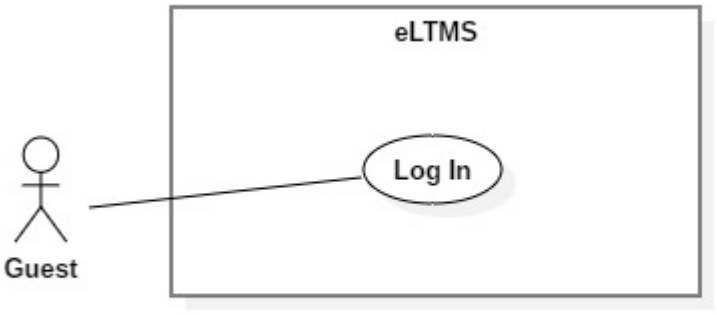


Figure 6 - <Guest> Login

Use Case Specification

USE CASE - UC_01			
Use Case No.	01	**Use Case Version**	1.0
Use Case Name	Login		
Author	DucBM		
Date	1/10/2018	**Priority**	Normal
Actor: - Guest.			

Summary:

- This use case allows Guest to login to the system.

Goal:

- Guests are authorized as their role.

Triggers:

- User clicks “Đăng nhập” button.

Preconditions:

- User has an account.

Post Conditions:

- Success: User logs into the system successfully and gets redirected to a specified page depends on user’s role.
- Fail: System shows error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User goes to Login page.	System requires information: • Phone: text input. • Password: password input.
2	User inputs information. User clicks “Đăng Nhập” button.	User will login to the system with roles of this account.

Exceptions:

No	Cause	System Response
1	Phone is invalid length.	Show error message.
2	Password is invalid length.	Show error message.
3	Phone and password does not match with any accounts.	Show error message.

Relationships: N/A

Business Rules:

- Patient can only log in to user-web.
- Employee (Hospital Receptionist, Department Receptionist, Nurse, Doctor, Warehouse Keeper and Manager) can only log in to admin-web.

1.2. <Guest> Register

Use Case Diagram

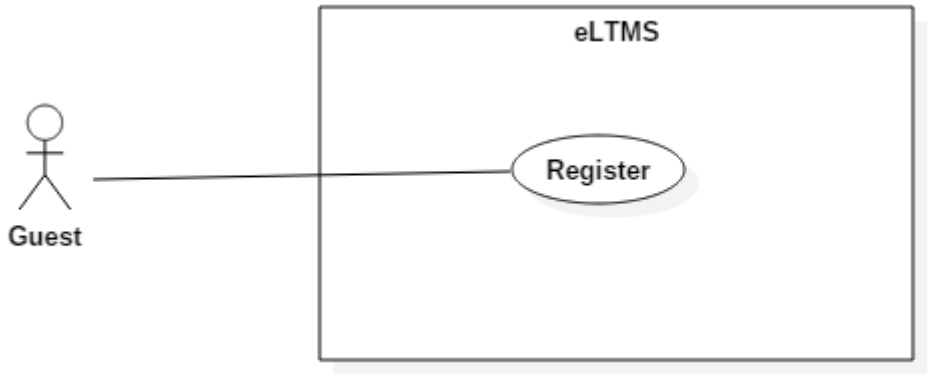


Figure 7 - <Guest> Register

Use Case Specification

USE CASE - UC_02			
Use Case No.	02	**Use Case Version**	1.0
Use Case Name	Register		
Author	DucBM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Guest.

Summary:

- This use case allows Guest to register account.

Goal:

- Create a new account for guest.

Triggers:

- User clicks "Đăng ký" button.

Preconditions:

- N/A

Post Conditions:

- **Success**: System saves account on database and shows success message.
- **Fail**: System shows error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User goes to "Đăng ký" page.	System requires information: • Phone: text input. • Password: password input.

		• Confirm password: password input • Full name: text input. • Date of birth: date input. • Email: text input. • Home address: text input. • Company address: text input.
2	User inputs all required field. Then clicks "Đăng ký" button.	User will register the system with roles of this account. [Exception from 1 to 8].

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Phone is invalid.	Show error message.
2	Password is invalid.	Show error message.
3	Confirm password does not match	Show error message.
4	Full name is empty.	Show error message.
5	Date of birth is invalid	Show error message.
6	Email is invalid.	Show error message.
7	Home address is invalid.	Show error message.
8	Company address is invalid.	Show error message.

Relationships: N/A

Business Rules:

- Guest can only register as patient.
- Phone is not null and must be verified by code.
- Email is not null, correct email format.
- Password is not null, in range of 8-255 characters, **includes** at least one **digit** and includes both **lowercase** and **uppercase** letters.
- Date of birth is not null and not over the current date.
- Full name is not null, in range of 5-50 characters.

1.3. <Guest, Patient> Make Appointment

Use Case Diagram

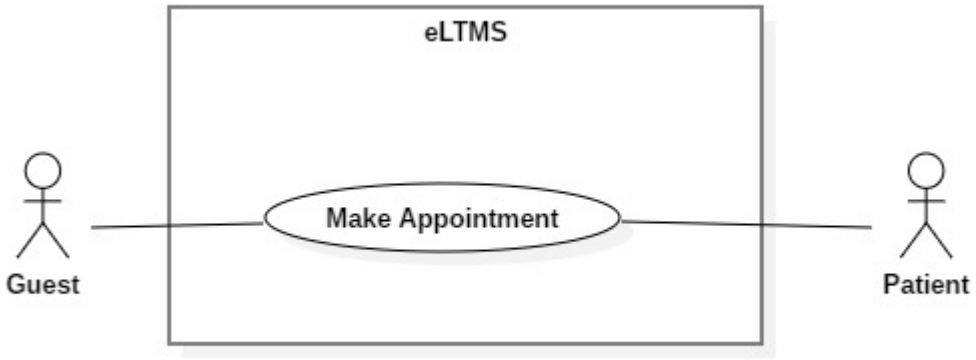


Figure 8 - <Guest, Patient> Make Appointment

Use Case Specification

USE CASE - UC_07			
Use Case No.	07	**Use Case Version**	1.0
Use Case Name	Make Appointment		
Author	DucBM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Patient

Summary:

- Patient makes appointment with personal information, types of LabTest, date and time.

Goal:

- A new appointment of the patient is made.

Triggers:

- User clicks on "Đặt lịch hẹn" menu item on the website or "Đặt lịch" button on mobile app.

Preconditions:

- N/A.

Post Conditions:

- **Success:** Show success message.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response

1	User clicks on “Đặt lịch hẹn”.	System requires information input: • Full name • Phone number
2	User clicks on “Bước 2”.	System displays a checkbox list of LabTest types for user to select.
3	User clicks on “Bước 3”.	System displays a table that shows the selected LabTests and requires input date and time for each of them.
4	User clicks on “Đặt lịch”.	System show message on a popup (success/error).

Exceptions:

No	Cause	System Response
1	Full name or phone number is empty.	Show error message.
2	No LabTests are selected.	Show error message.
3	A date or time is empty.	Show error message.
4	Two date-times intersects.	Show error message.

Relationships: N/A

Business Rules:

- User cannot book a Sample twice or more on the same day.
- Number of bookings of a Sample-Group, on the same day, at the same slot cannot exceed the number of tables of that Sample-Group.
- User cannot make an appointment with date-time in the past.
- User cannot book 2 date-times which intersects each other.

2. Patient Overview Use Case

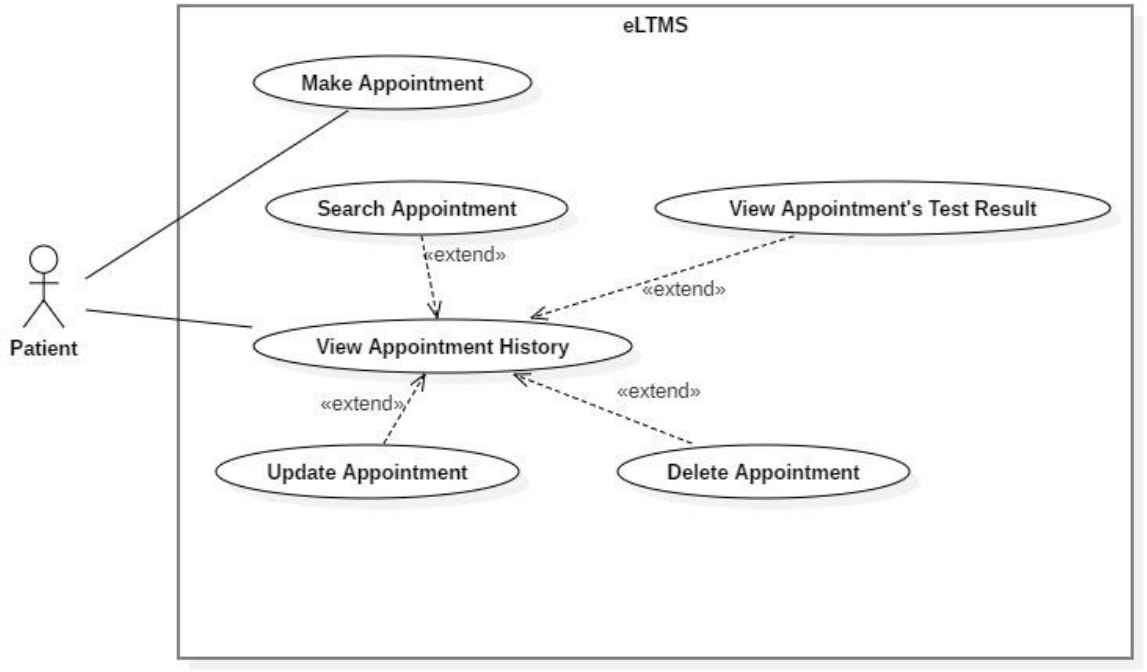


Figure 9 - < Patient > Overview Use Case

2.1. <Patient> View Appointment History

Use Case Diagram

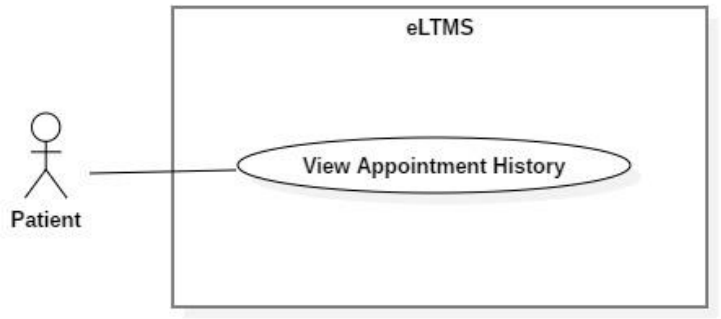


Figure 10 - <Patient> View Appointment History

Use Case Specification

USE CASE - UC_07			
Use Case No.	07	**Use Case Version**	1.0
Use Case Name	View Appointment History		
Author	DucBM		

Date	1/10/2018	**Priority**	Normal

Actor:

- Patient

Summary:

- Patient views his/her appointment history including new appointment, being processed appointment and done appointment.

Goal:

- A list of appointments of the patient, with search, filter and pagination controls, is shown.

Triggers:

- User clicks on “Danh sách cuộc hẹn” on the Website/Mobile App.

Preconditions:

- User is logged in as Patient.

Post Conditions:

- **Has data**: Show a list of appointments of the patient with search, filter and pagination controls.
- **Has no data**: Show “Bạn chưa có cuộc hẹn nào.” text.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on “Danh sách cuộc hẹn” on the Website or Mobile App.	System shows a list of appointments of the patient with search, filter and pagination controls.

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on “Danh sách cuộc hẹn” on the Website or Mobile App.	System shows a text: “Bạn chưa có cuộc hẹn nào.”

Exceptions: N/A

Relationships: N/A.

Business Rules:

- In this list, each appointment has 3 status:
  - "Mới tạo": User can edit and delete on Web. User can view detail (selected Samples, LabTests, date, time and price), edit and delete on Mobile App.
  - "Đang xử lý": User can do nothing
  - "Hoàn tất": User can view result on Web. User can view detail and view result on Mobile App.

2.2. <Patient> Update Appointment

Use Case Diagram

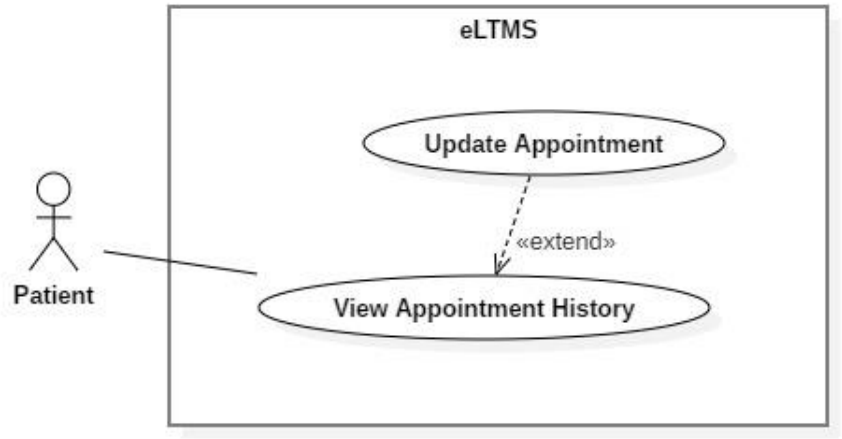


Figure 11 - <Patient> Update Appointment

Use Case Specification

USE CASE - UC_08			
Use Case No.	08	**Use Case Version**	1.0
Use Case Name	Update Appointment		
Author	DucBM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Patient

Summary:

- Update an appointment's LabTest types, date and time.

Goal:

- LabTest types, date and time of an appointment are updated.

Triggers:

- User clicks Edit Button in the appointment row.

Preconditions:

- User is logged in as Patient.

Post Conditions:

- **Success**: A success message is shown.
- **Fail**: An error message is shown.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on Edit Button in the appointment row.	System redirect to Edit page. Old values of LabTests are set on a checkbox list.
2	User re-selects new LabTests. Then, user clicks "Bước 2"	System show a table of Samples with date and time.
3	User re-selects date and time. Then, user clicks "Cập Nhật"	System processes and return message.

Alternative Scenario: N/A

Exceptions: N/A

Relationships: Extended from View Appointment History use case.

Business Rules:

- User cannot book a Sample twice or more on the same day.
- Number of bookings of a Sample-Group, on the same day, at the same slot cannot exceed the number of tables of that Sample-Group.
- User cannot make an appointment with date-time in the past.
- User cannot book 2 date-times which intersects each other.

2.3. <Patient> View Appointment's Test Result

Use Case Diagram

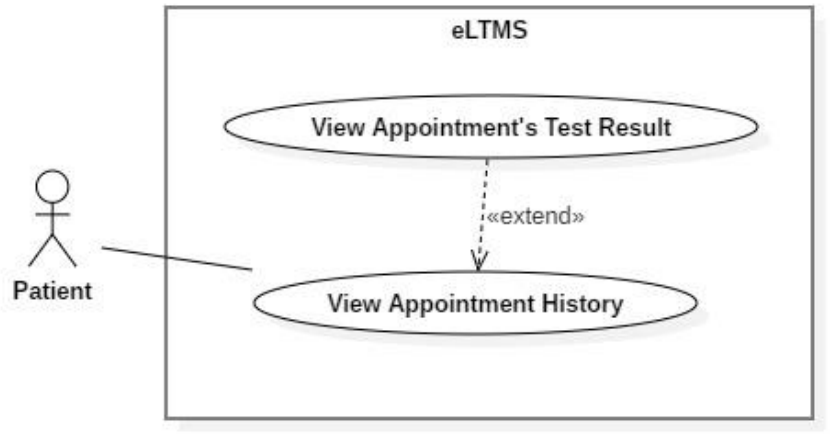


Figure 12 - <Patient> View Appointment's Test Result

Use Case Specification

USE CASE - UC_10			
Use Case No.	10	**Use Case Version**	1.0
Use Case Name	View Appointment's Test Result		
Author	DucBM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Patient

Summary:

- Patient views appointment's test result.

Goal:

- Patient views appointment's test result, a soft version of result paper.

Triggers:

- Patient clicks View Result Button in the appointment row.

Preconditions:

- User is logged in as Patient.

Post Conditions:

- The test result is shown.

Main Success Scenario:

Step	Actor Action	System Response

1	User clicks on View Result Button on the appointment row.	System redirects to View Result page which shows: - Test results indexes: Name, value, status (normal/low/high), result, unit - "Tải tập tin PDF" button.
2	User clicks on "Tải tập tin PDF".	A PDF version is downloaded and opened on browser.

Exceptions: N/A

Relationships: Extended from View Appointment History use-case.

Business Rules:

- In the test result grid: the rows has high or low status must be highlighted (bold, italic and underlined). If high, the result number marked "H" and righ-aligned. If low, the result number marked "L" and left-aligned.

2.4. <Patient> Delete Appointment

Use Case Diagram

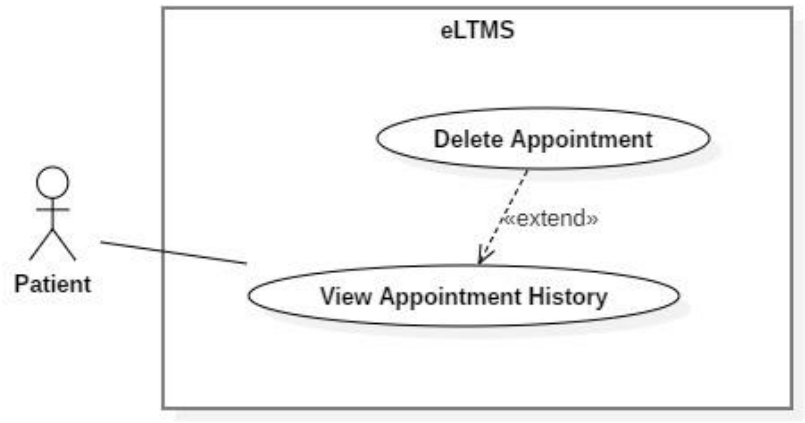


Figure 13 - <Patient> Delete Appointment

Use Case Specification

USE CASE - UC_12			
Use Case No.	12	**Use Case Version**	1.0
Use Case Name	Delete Appointment		
Author	DucBM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Patient

Summary:

- Delete (cancel) an appointment that patient made.

Goal:

- Help patients can flexibly delete the appointments that they cannot come.

Triggers:

- Patient clicks on Delete Button on the appointment row.

Preconditions:

- User is logged in as Patient.

Post Conditions:

- Show success message.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on Delete Button on the appointment row.	System show a confirm popup with message: "Bạn có chắc chắn muốn hủy cuộc hẹn này không?"
2	User clicks "Có".	System deletes the appointment and show popup: "Cuộc hẹn đã được hủy thành công."

Alternative Scenario:

Step	Actor Action	System Response
1	User clicks on Delete Button on the appointment row.	System show a confirm popup with message: "Bạn có chắc chắn muốn hủy cuộc hẹn này không?"
2	User clicks "Không".	System closes the popup and come back to the appointment history list.

Exceptions:

No	Cause	System Response
1	The appointment have been deleted before user confirmed "Có". It is deleted from another device (maybe web or mobile).	Show error message.

Relationships: Extended from View Appointment History List.

Business Rules: N/A

3. Hospital Receptionist Overview Use Case

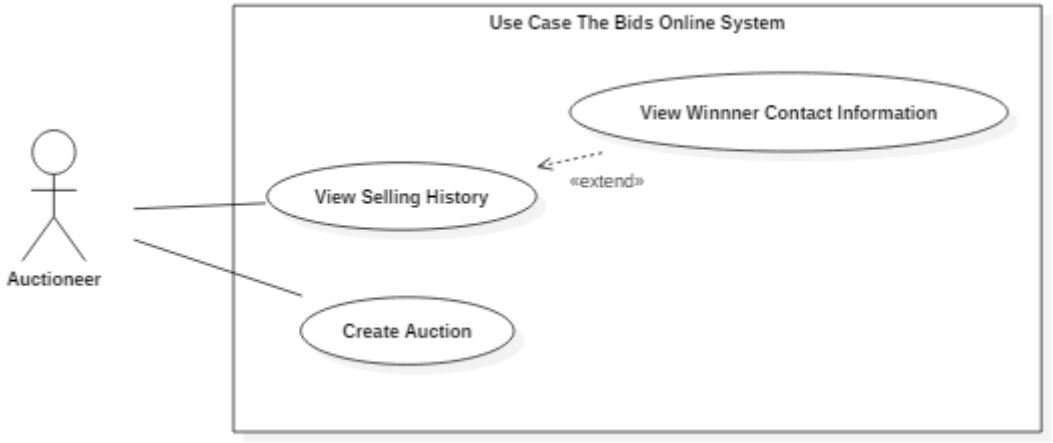


Figure 14 - < Hospital Receptionist> Overview Use Case

3.1. <Hospital Receptionist, Manager> View Appointment List

Use Case Diagram

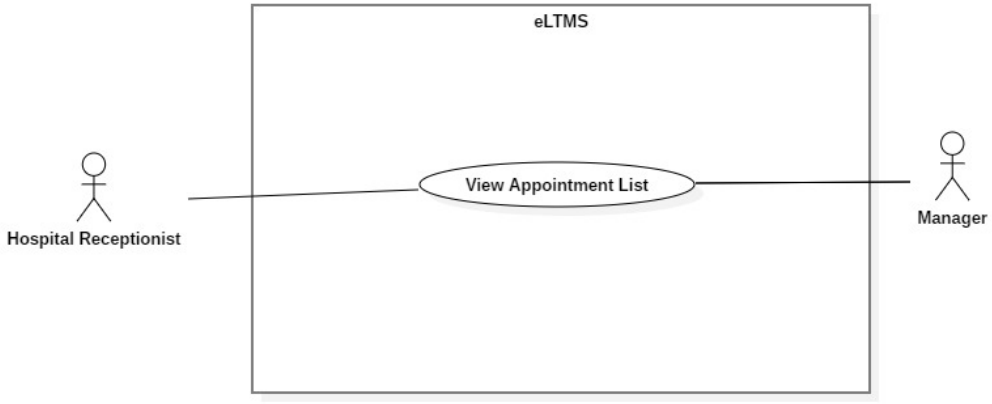


Figure 15 - <Hospital Receptionist> View Appointment List

Use Case Specification

USE CASE - UC_10			
Use Case No.	10	**Use Case Version**	1.0

Use Case Name	View Appointment List		
Author	YCM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Auctioneer.

Summary:

- Hospital Receptionist views a list of appointments to guide patients coming to the hospital.

Goal:

- Hospital Receptionist can view all appointments, search by appointment code and filter by date.

Triggers:

- After Hospital Receptionist logged in successfully or clicks on "Quản lý cuộc hẹn" menu item.

Preconditions:

- User logged in as Hospital Receptionist.

Post Conditions:

- A list of appointments is shown, with search, filter and pagination controls.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on "Quản lý cuộc hẹn" menu item.	A list of appointments is shown, with search, filter and pagination controls.

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Appointment table has no data.	Show message: "Chưa có cuộc hẹn nào"

Relationships: N/A

Business Rules:

- Initially, all IsPaid Switches are set to OFF.

3.2. <Hospital Receptionist, Manager> View Appointment Detail

Use Case Diagram

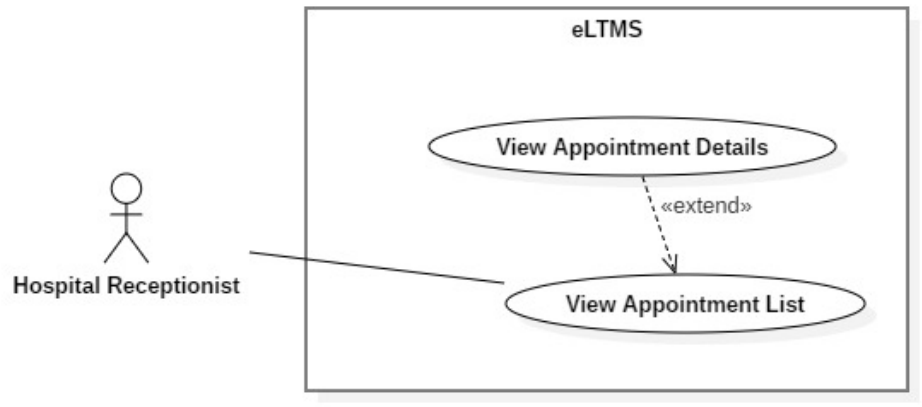


Figure 16 - <Hospital Receptionist, Manager> View Appointment Details

Use Case Specification

USE CASE - UC_11			
Use Case No.	11	**Use Case Version**	1.0
Use Case Name	View Appointment Details		
Author	YCM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Hospital Receptionist.
- Manager

Summary:

- This use-case allows Hospital Receptionist to view appointment details.

Goal:

- Appointment details, includes patient's information and a list of booked Sample-Getting of the appointment, are shown.

Triggers:

- User clicks on "Xem chi tiết" button on the appointment row.

Preconditions:

- User logged in as Hospital Receptionist.
- User is at View Appointment List page.

Post Conditions:

- **Success**: A pop up with appointment details is shown.

Main Success Scenario:

Step	Actor Action	System Response

1	User clicks on on "Xem chi tiết" button on the appointment row.	A pop up appears and shows these information: - Patient profile: Full name, phone number, gender, date of birth, email, home address, company address. - A list of booked Sample-Getting.

Alternative Scenario: N/A

Exceptions: N/A

Relationships: Extend from View Appointment List use case

Business Rules: N/A.

3.3. <Hospital Receptionist, Manager> Delete Sample Getting

Use Case Diagram

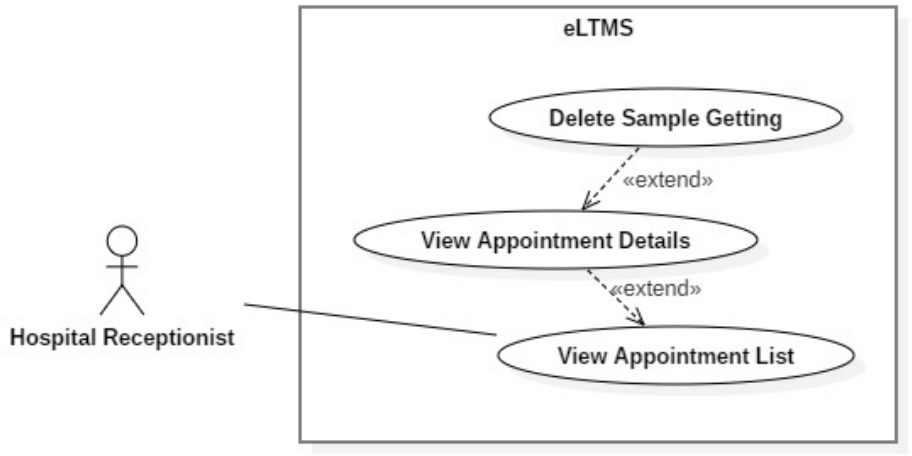


Figure 17 - <Hospital Receptionist, Manager> Delete Sample Getting

Use Case Specification

USE CASE - UC_10			
Use Case No.	10	**Use Case Version**	1.0
Use Case Name	Delete Sample Getting		
Author	YCM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Hospital Receptionist

Summary:

- This use-case allows Hospital Receptionist to delete Sample-Getting for online patient.

Goal:

- Sample-Getting of online patient is deleted.

Triggers:

- Hospital Receptionist clicks on Delete Button on the Sample-Getting row.

Preconditions:

- User logged in as Hospital Receptionist.
- User is at Appointment Detail Popup.

Post Conditions:

- **Success**: Show success message.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User clicks on Delete Button on the Sample-Getting row.	System shows a popup: "Bạn có chắc chắn muốn xóa ca lấy mẫu này không?"
2	User clicks "Có".	System deleted the Sample-Getting and shows pop up: "Ca lấy mẫu đã được xóa."

Alternative Scenario 1:

Step	Actor Action	System Response
1	User clicks on Delete Button on the Sample-Getting row.	System shows a popup: "Bạn có chắc chắn muốn xóa ca lấy mẫu này không?"
2	User clicks "Không".	System closes the popup and come back to the Sample-Getting list.

Exceptions: N/A

Relationships: N/A

Business Rules:

- Price should be displayed as VND currency format.
- Date should be displayed as Vietnamese date format.
- For active auctions: user can sort the result by active time (newly listed/ending soon), number of bids (asc/desc), current price (asc/desc). User also can click on the row to visit auction's detail page.
- For ended auctions: user can sort the result by ended time (asc/desc), number of bids (asc/desc), current price (asc/desc). Each row provides a Contact icon for user can click and view seller's contact information if the auction has a winner.

4. Department Receptionist Overview Use Case

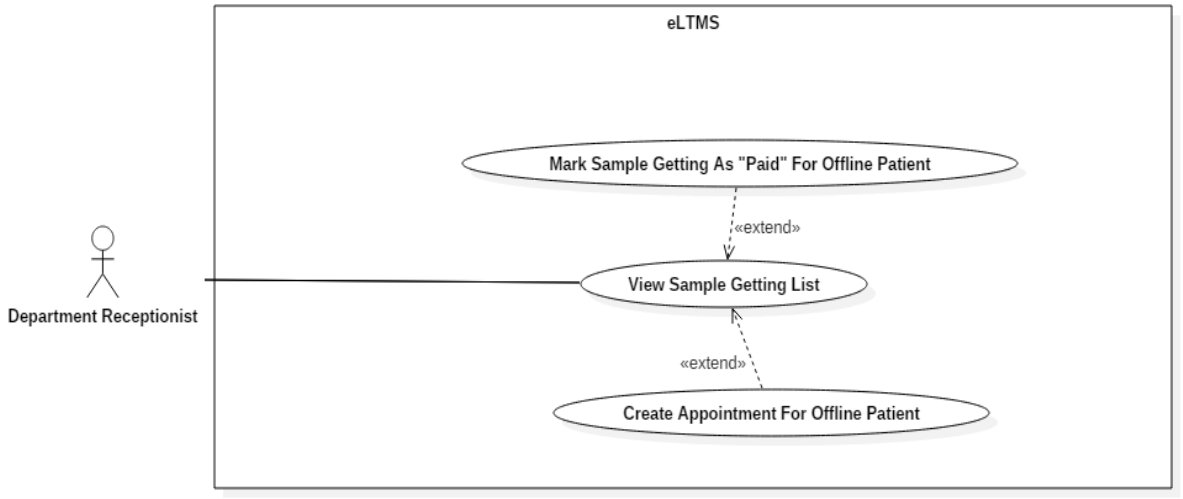
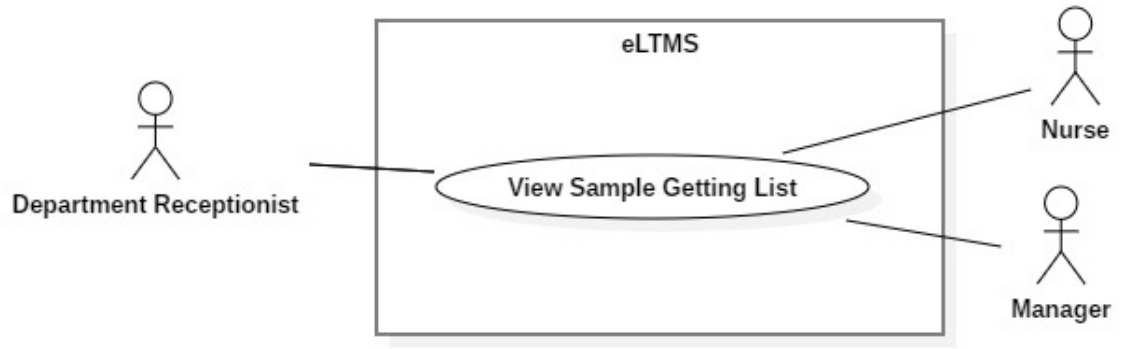


Figure 18 - < Department Receptionist > Overview Use Case

4.1. <Department Receptionist, Manager, Nurse> View Sample Getting List

Use Case Diagram



Use Case Specification

USE CASE - UC_13			
Use Case No.	13	**Use Case Version**	1.0
Use Case Name	View Sample Getting List		
Author	YCM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Department Receptionist
- Nurse
- Manager

Summary:

- View a list of Sample Gettings.

Goal:

- A list of Sample Gettings with suitable controls is shown.

Triggers:

- User Logged in successfully as Department Receptionist or Manager.

Preconditions:

- User logged in as Department Receptionist or Manager.

Post Conditions:

- Success: A list of Sample Getting is shown, with search, filter and pagination controls.

Main Success Scenario:

Step	Actor Action	System Response

1	Users logged in successfully as Department Receptionist, Nurse or Manager.	System shows list of Sample Getting which includes following information: • Patient's full name • Patient's phone number • Patient's home address • Sample Getting's occurring date • Sample Getting's occurring slot • Action button: Paid

Exceptions:

No	Cause	System Response
1	There are no data.	Show message: "Không có ca lấy mẫu nào."

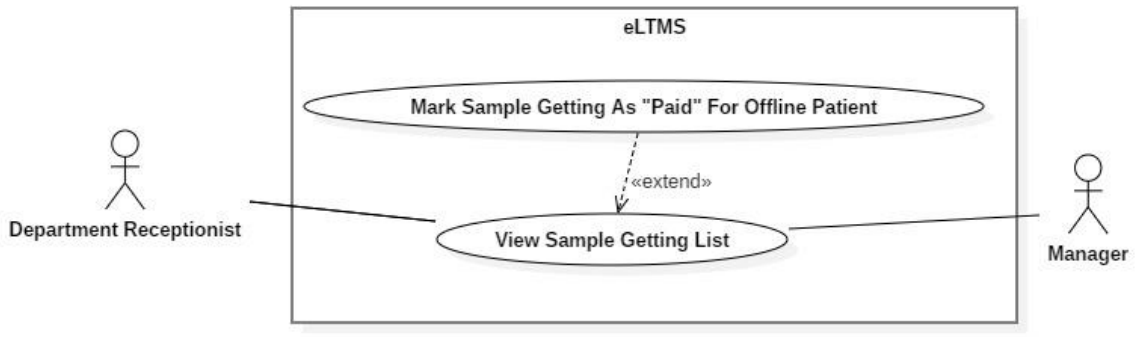
Relationships: N/A

Business Rules:

- Initially, system displays a list of Sample-getting of the current date.

4.2. <Department Receptionist, Manager> Mark Sample Getting As "Paid" For Offline Patient

Use Case Diagram



Use Case Specification

USE CASE - UC_13			
Use Case No.	13	**Use Case Version**	1.0
Use Case Name	Mark Sample Getting As "Paid" For Offline Patient		
Author	YCM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Department Receptionist
- Manager

Summary:

- Mark Sample Getting as "Paid" for offline patient.

Goal:

- Sample Getting is marked as "Paid" to confirm that the patient has paid for his/her Sample Getting.

Triggers:

- Department Receptionist/Manager clicks on Paid Switch.

Preconditions:

- User logged in as Department Receptionist or Manager.
- User is at View Sample Getting List page.

Post Conditions:

- **Success**: Sample Getting's status is changed into "*Paid*".

Main Success Scenario:

Step	Actor Action	System Response
1	Click on Paid Switch.	Show a confirm pop up with message "Bạn có chắc chắn muốn thanh toán không?"
2	Click "Có" button.	Change status of the Sample Getting and display message "Đã thanh toán".

Alternative Scenario:

Step	Actor Action	System Response
1	Click on Paid Switch.	Show a confirm pop up with message "Bạn có chắc chắn muốn thanh toán không?"
2	Click "Có" button.	Change status of the Sample Getting and display message "Đã thanh toán" Set Paid switch to ON

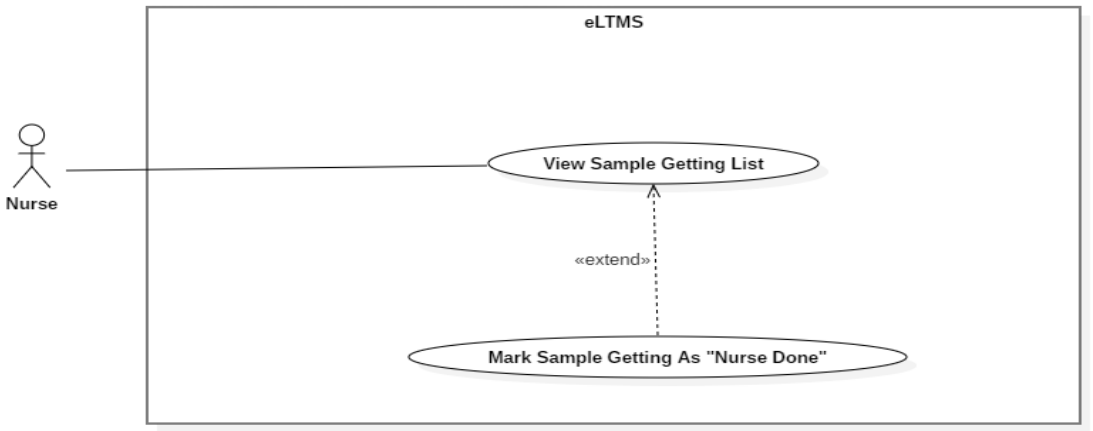
Exceptions: N/A

Relationships: Extend from View Sample Getting List

Business Rules:

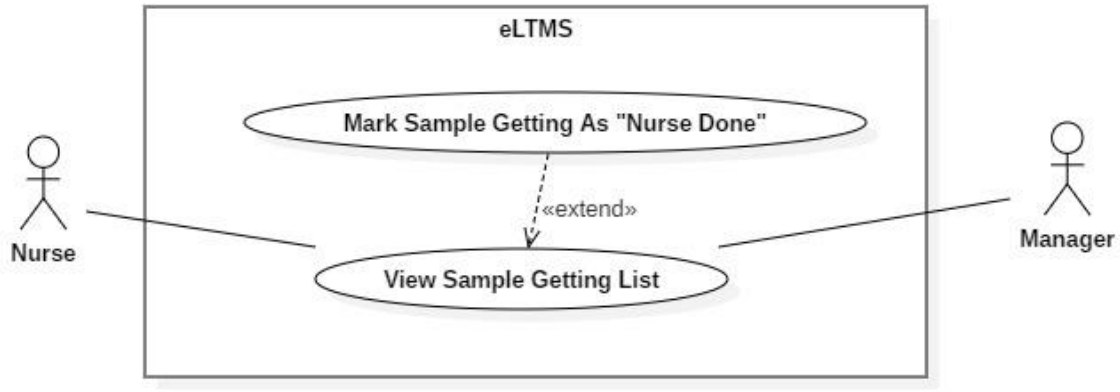
- Initially, all Paid switches are set as OFF.

5. Nurse Overview Use Case



5.1. <Nurse, Manager> Mark Sample Getting As “Nurse Done”

Use Case Diagram



Use Case Specification

USE CASE - UC_14			
Use Case No.	14	**Use Case Version**	1.0
Use Case Name	Mark Sample Getting As “Nurse Done”		
Author	YCM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Nurse.
- Manager

Summary:

- Mark Sample Getting as “Nurse Done”

Goal:

- Sample Getting is marked as “Nurse Done”.

Triggers:

- User clicks on the switch on the “Đã lấy mẫu” column of the Sample Getting’s row.

Preconditions:

- User logged in as Nurse or Manager.
- User is at View Sample Getting List page.

Post Conditions:

- **Success**: Show success message. Sample Getting status is changed into “*Nurse Done*”

Main Success Scenario:

Step	Actor Action	System Response
1	Click on the switch on the “Đã lấy mẫu” column of the Sample Getting’s row.	Show a confirm pop up: “Bạn có chắc chắn đã lấy mẫu?”
2	Click “Có” button.	Show success message.

Alternative Scenario:

Step	Actor Action	System Response
1	Click on the switch on the “Đã lấy mẫu” column of the Sample Getting’s row.	Show a confirm pop up: “Bạn có chắc chắn đã lấy mẫu?”
2	Click “Không” button.	Come back to the Sample Getting List.

Exceptions: N/A

Relationships: Extended from View Sample Getting List

Business Rules:

- Initially, all switches are set to OFF.

Use Case Specification

USE CASE - UC_13			
Use Case No.	13	**Use Case Version**	1.0
Use Case Name	Mark Sample Getting As "Paid" For Offline Patient		
Author	YCM		
Date	1/10/2018	**Priority**	Normal

Actor:

- Department Receptionist
- Manager

Summary:

- Mark Sample Getting as "Paid" for offline patient.

Goal:

- Sample Getting is marked as "Paid" to confirm that the patient has paid for his/her Sample Getting.

Triggers:

- Department Receptionist/Manager clicks on Paid Switch.

Preconditions:

- User logged in as Department Receptionist or Manager.
- User is at View Sample Getting List page.

Post Conditions:

- **Success**: Sample Getting's status is changed into "*Paid*".

Main Success Scenario:

Step	Actor Action	System Response
1	Click on Paid Switch.	Show a confirm pop up with message "Bạn có chắc chắn muốn thanh toán không?"
2	Click "Có" button.	Change status of the Sample Getting and display message "Đã thanh toán".

Alternative Scenario:

Step	Actor Action	System Response
1	Click on Paid Switch.	Show a confirm pop up with message "Bạn có chắc chắn muốn thanh toán không?"
2	Click "Có" button.	Change status of the Sample Getting and display message "Đã thanh toán" Set Paid switch to ON

Exceptions: N/A

Relationships: Extend from View Sample Getting List **Business Rules:** • Initially, all Paid switches are set as OFF.

VII. System Architectural Design

Our system has been developed under the 3-layer architecture. The 3-layer architecture is different from the 3-tier architecture, because 3 components are separated logically (can be deployed on the same server), not physically (on different servers).

We choose this architecture for our system because of following advantages:

- **Speed of development:** a specific layer can be upgraded with minimal impact on the other layers.
- **Scalability**: By separating out the different layers we can scale each independently depending on the need at any given time.
- **Performance:** This allows you to load balance each layer independently, improving overall performance with minimal resources.

The 3-layer architecture consists of:

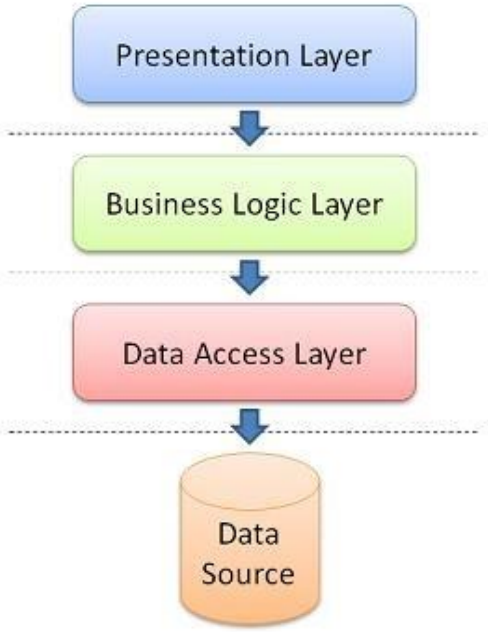


Figure 199 - System Architecture Design

- **Presentation Layer:** consists of the user interface. This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end user.
- **Business Logic Layer:** contains the functional business logic which drives an application's core capabilities.
- **Data Access Layer:** provides simplified access to data source

VIII. Component Diagram

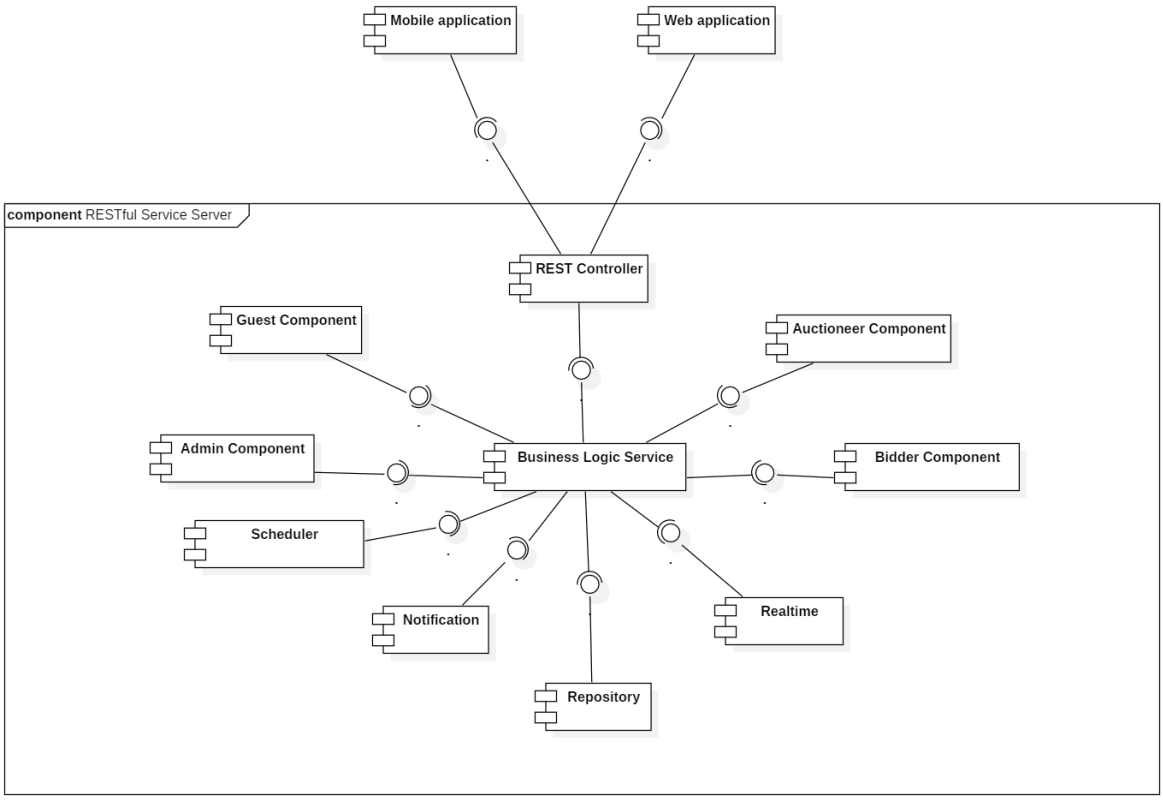


Figure 20 - Component Diagram

COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
Mobile Application	Component to contain system's functions for end user on mobile device
Web Application	Component to contain system's functions for end user on web

REST Controller	Component to receive requests, call services to process information, get data back from services and respond result to clients
Business Logic Service	Component to handle system's business operations
Guest Component	Component to handle guest's activities in the system
Admin Component	Component to handle admin's activities in the system
Bidder Component	Component to handle bidder's activities in the system
Auctioneer Component	Component to handle auctioneer's activities in the system
Scheduler	Component to handle scheduling future tasks
Realtime	Component to handle realtime operations
Notification	Component to handle sending notifications
Repository Component	Component to handle data reading/writing operations from/to database
COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
Mobile Application	Component to contain the system's functions for the end user on the mobile device.
Web Application	Component to contain system's functions for end user on web.
REST Controller	Component to receive requests, call services to process information, get data back from services and respond result to clients.
Business Logic Service	Component to handle the system's business operations.
Guest Component	Component to handle guest's activities in the system.
Admin Component	Component to handle admin's activities in the system.
Bidder Component	Component to handle bidder's activities in the system.
Auctioneer Component	Component to handle auctioneer's activities in the system.
Scheduler	Component to handle scheduling future tasks.
Realtime	Component to handle real-time operations.
Notification	Component to handle sending notifications.

Repository Component	Component to handle data reading/writing operations from/to database.

Table 3 - Component Dictionary

IX. Class Diagram

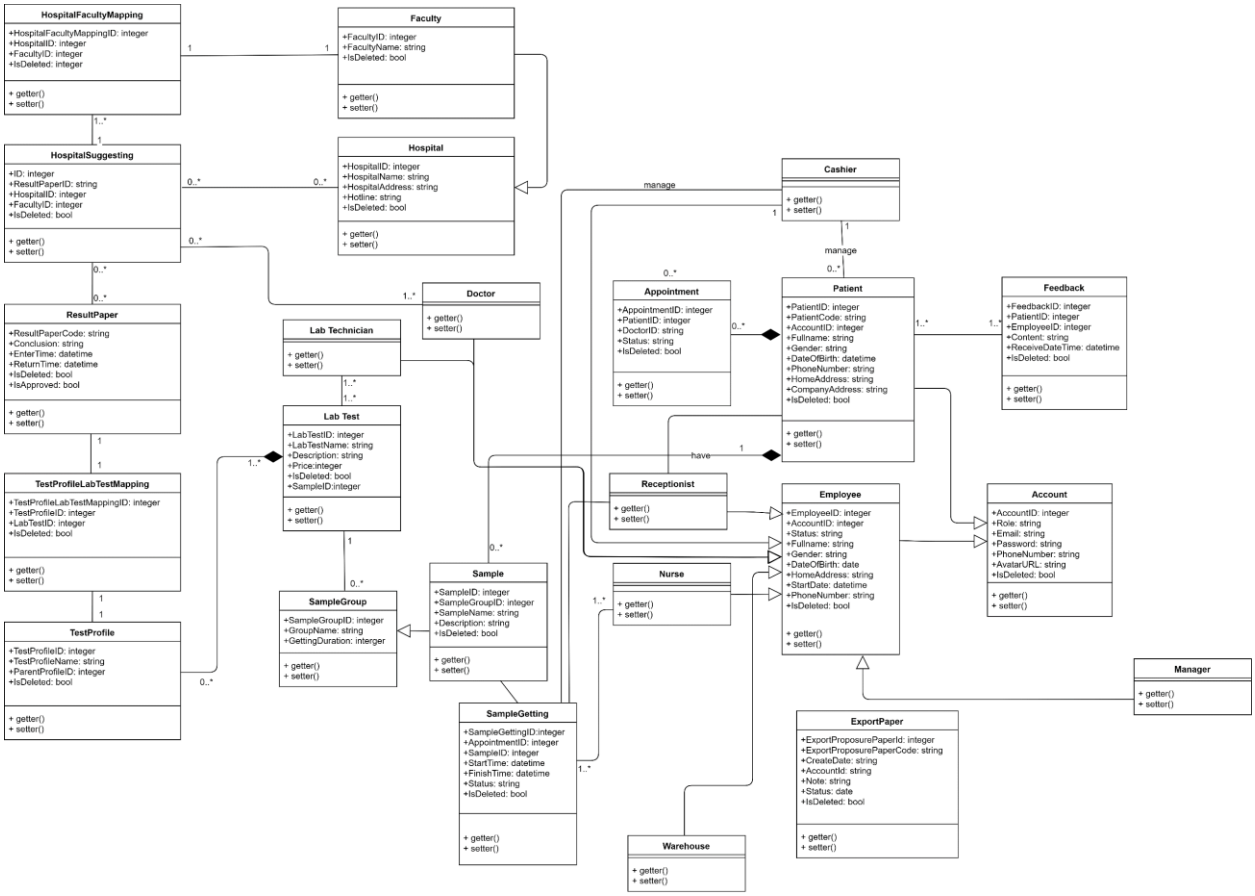


Figure 21 - Class Diagram

CLASS DICTIONARY: DESCRIBE CLASS	
Class Name	**Description**
Patient	Patient
Employee	Employee
Feedback	Feedback
Account	Account
Manager	Manager
Receptionist	Receptionist
Appointment	Appointment
ExportPaperDetail	ExportPaperDetail
ExportPaper	ExportPaper
Supply	Supply
SupplyType	SupplyType
ImportPaperDetail	ImportPaperDetail

Table 4 - Class Dictionary

X. Entity relationship diagram (ERD)

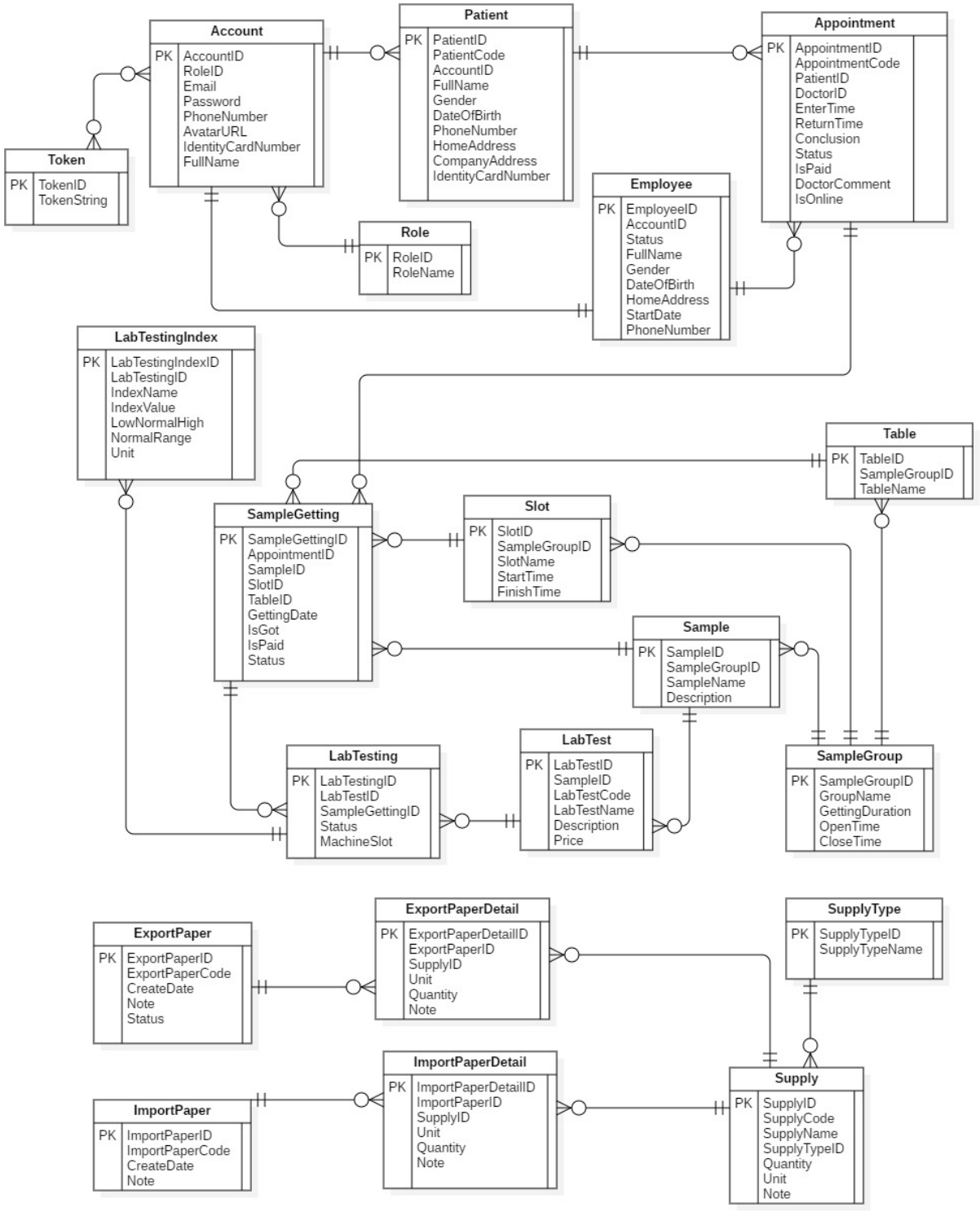


Figure 22 - Entity relationship diagram (ERD)

XI. Interactive Diagram

1. Guest

1.1. <Guest> Make Appointment

Summary this diagram shows process for making appointment.

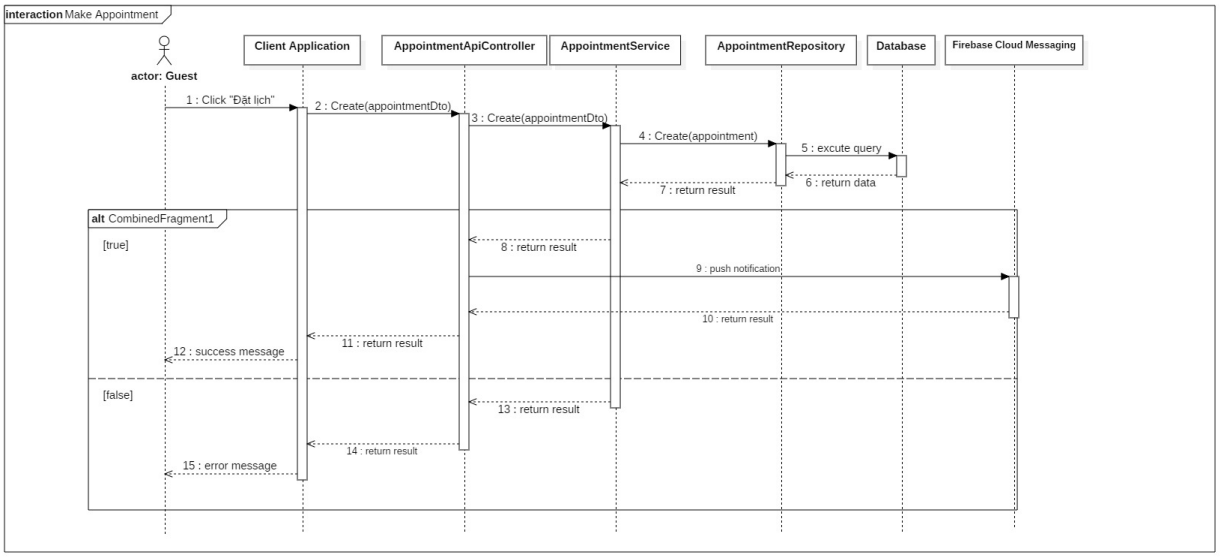


Figure 23 - Sequence Diagram for Make Appointemnt < Guest >

2. Patient

2.1. <Patient> View Appointment's Test Result

Summary: this diagram shows process for viewing appointment's test result.

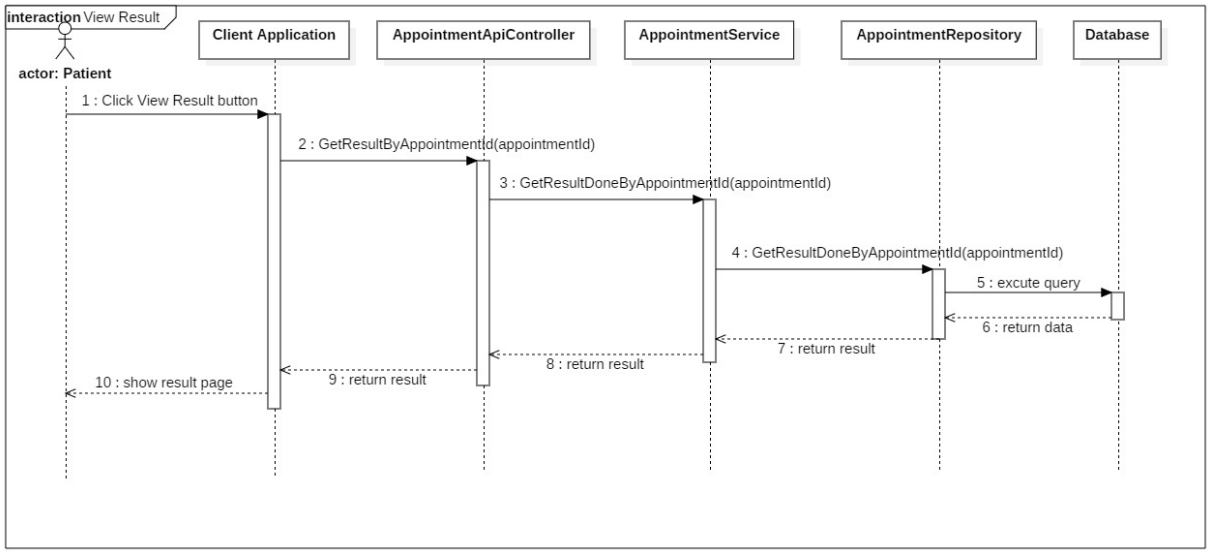


Figure 24 - Sequence Diagram for view appointment< Patient >

XII. State Machine Diagram

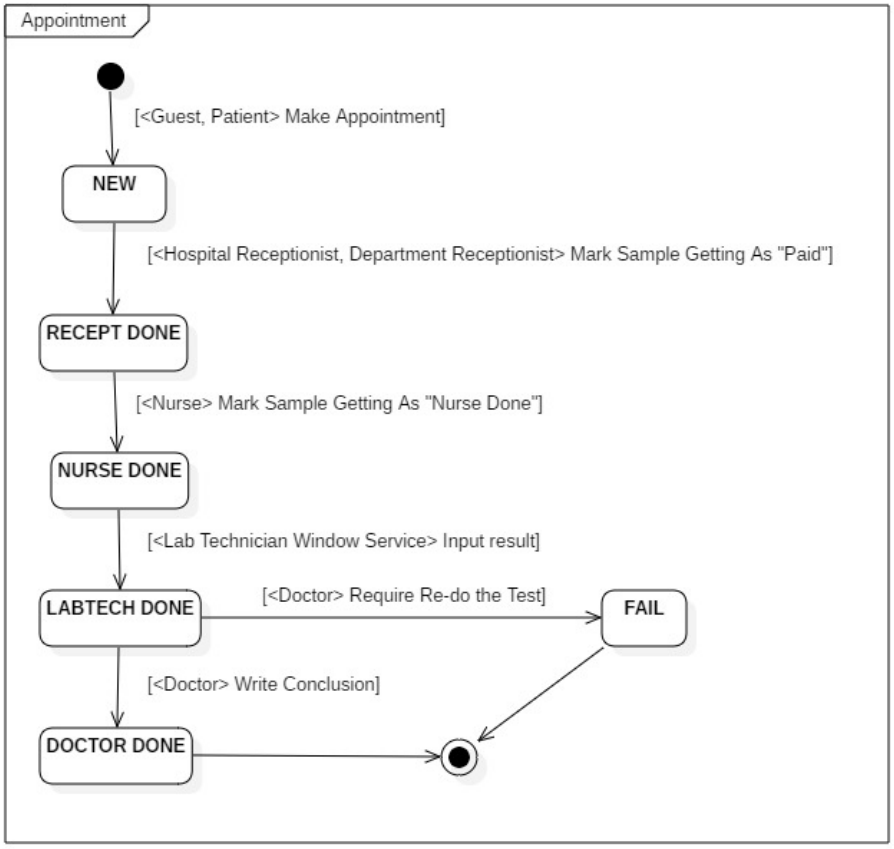


Figure 25 - State Machine Diagram of Appointment

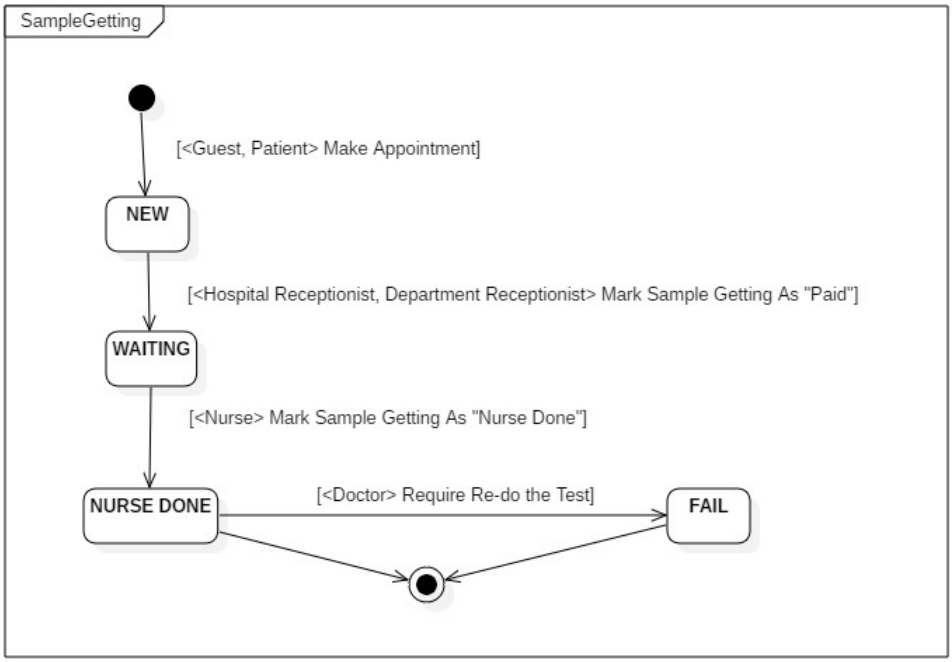


Figure 26 - State Machine Diagram of Sample Getting

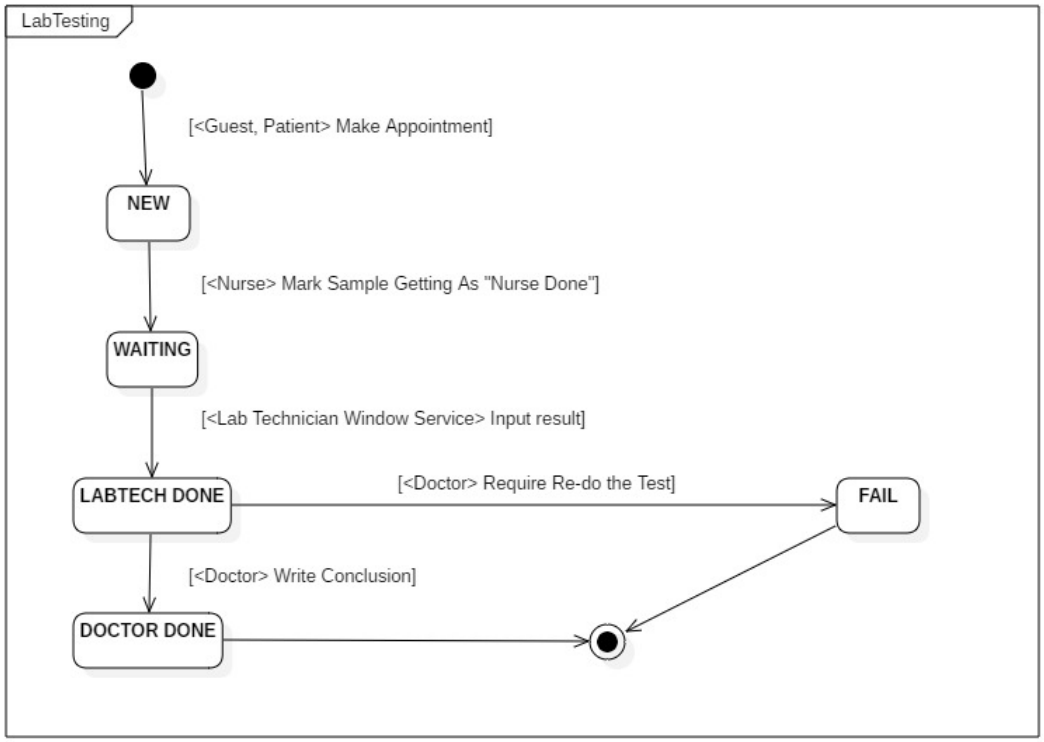


Figure 27 - State Machine Diagram of Lab Testing

XIII. Physical diagram

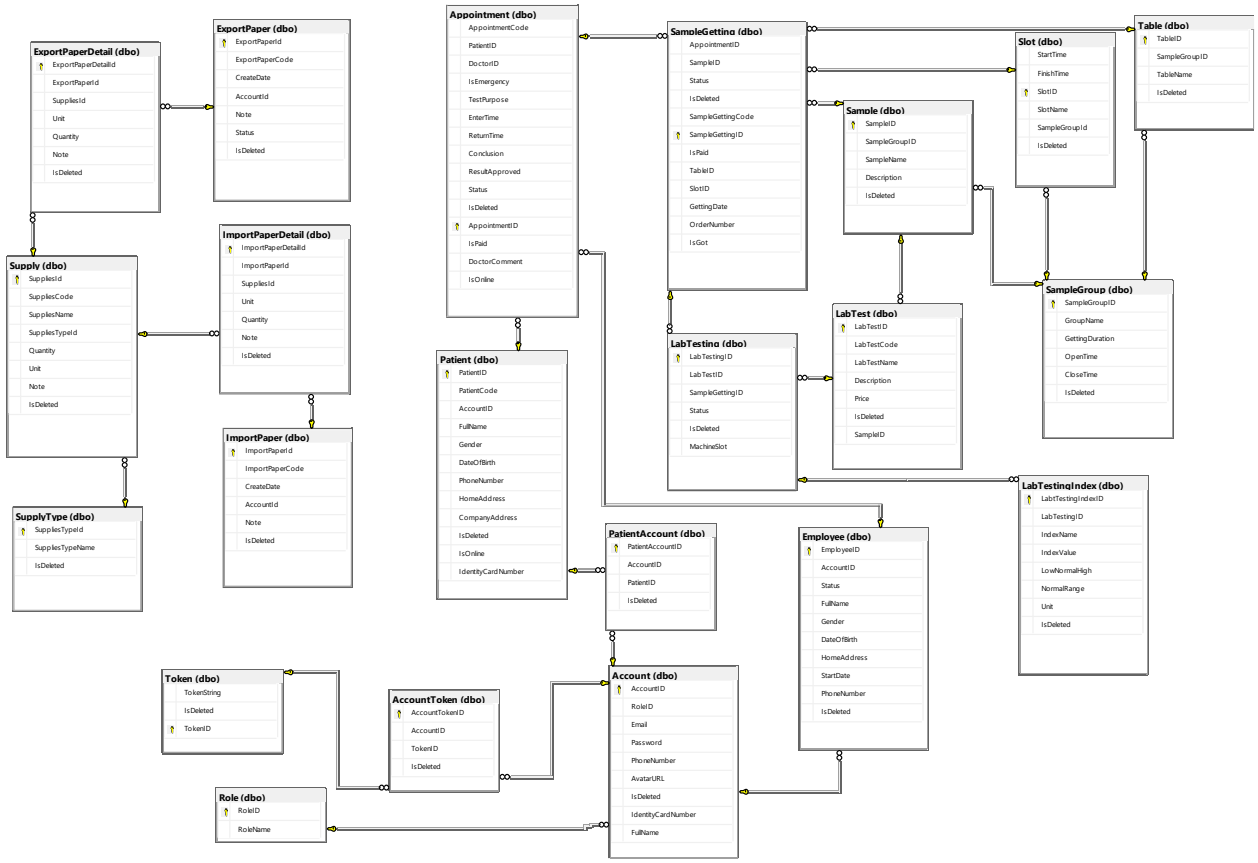


Figure 28 – Physical Database Diagram

XIV. Algorithms

1. Slot Suggestion

1.1. Definition

Generate the set of samples getting slots which has minimum total time.

1.2. Define problem

- When create a new appointment, user often have to spend a lot of effort to choose the suitable slots. Then, there is a need for an automatic solution using algorithms.
- There are N sample-gettings that the patient needs to perform. (blood getting, urine getting, are sample-gettings)
- Each room has a set of slots which have the same durations. Different sample getting rooms can have different slot durations.

1.3. Assume that patients need to do all sample-gettings that they chosed in ONE day, so that the total time is minimum Solution

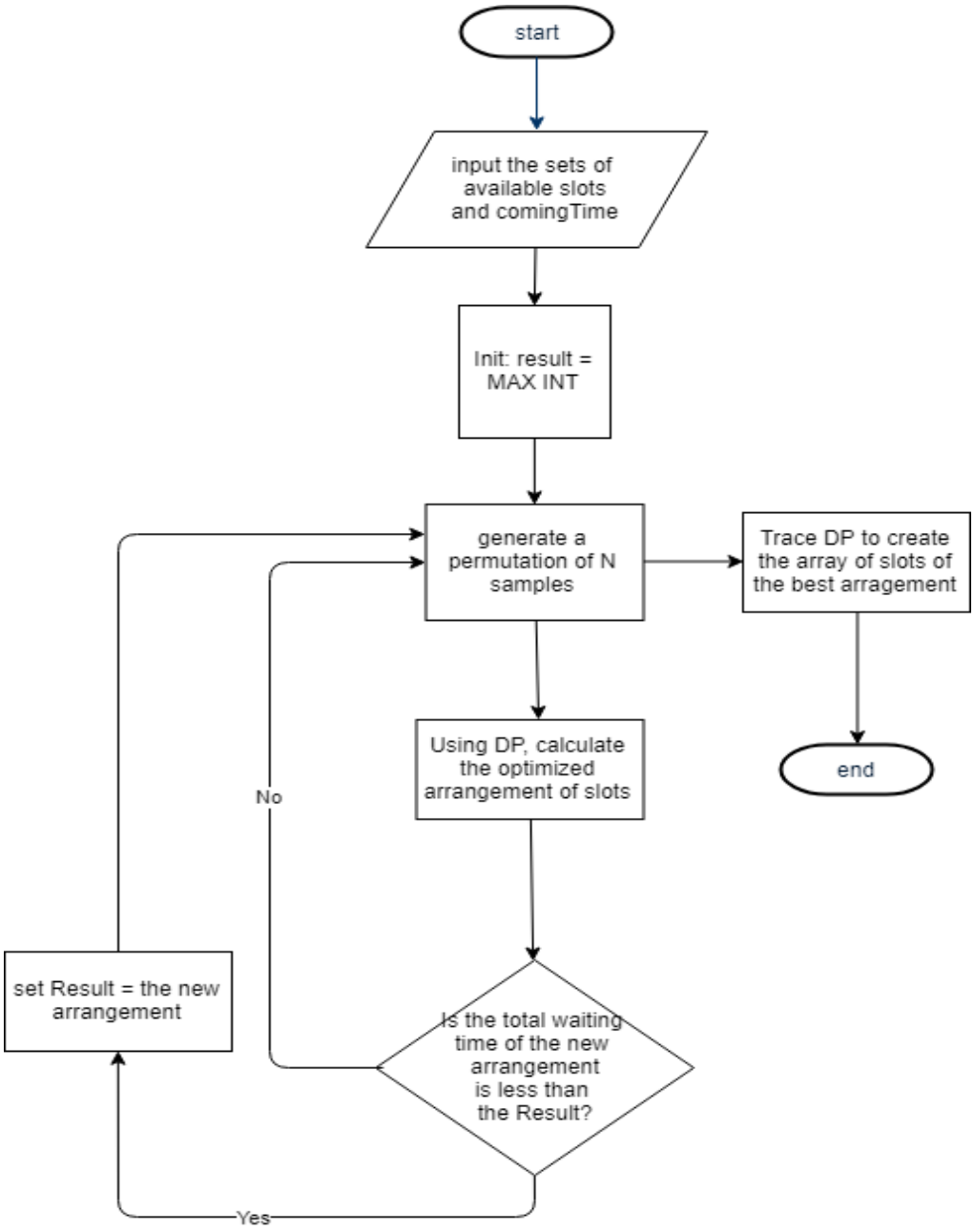
We use permutation recursive permutation and dynamic programming (DP) to solve this problem. We follow 2 steps:

- Step 1: Consider each permutation of N sample getting rooms in order, we use DP to calculate the optimized set of slots in the order of sample-gettings.
- Step 2: DP Formula:
  - Assume that T[i][j].StartTime: the start time of the j-slot in the i-th slot set. (the i-th slot set belongs to the i-th sample getting).
  - Assume that T[i][j].FinishTime: the finish time.
  - Assume that dp[i][j] is the minimum waiting time (between slots of sample gettings) if the last sample-getting is the i-th one and the it's chosed slot is the j-th slot. i is from 0 to N-1, j is from 0 to Length of i-th slot set – 1. Because the sample-gettings' duration is fixed, the order, which has the minimum waiting time, has the minimum total time.
  - dp[i][j] = min {dp[i-1][k] + (T[i-1][k].StartTime – T[i][j].FinishTime) | T[i-1][k].StartTime – T[i][j].FinishTime > 0}.
  - Base: dp[0][j] = T[0][j] – comingTime. (comingTime: the time that the patient arrives at the hospital).
  - The result of problem is min {dp[n-1][j]}.
- Step 3: Trace to get the optimized order of slots.

1.4. Complexity

The complexity of this algorithm is **O(N! * N^3).** In which, N is the number of selected Samples of a patient on a day**.**

1.5. Flowchart



XVI. Future plan:

Current system is concentrated on core business flow. Therefore, some supporting features are restricted for the development team. These features may be expanded in the future:

- Support more categories
- Develop stronger security to reduce the risk for the system and users
- Develop getting detailed statistics for testing by month, year
- Develop system on mobile for admin