



Ain Shams University
Faculty of Engineering

Third year - Computers and Systems

Report on

CSE 321 : Software Engineering Project
Hospital Management System

Submitted to

Dr. Gamal A. Ebrahim

1701596	نوران احمد فؤاد
17E0026	روان خالد احمد
17 00553	ريناد مصطفى محمد
1701595	نورا نبيل علي
1701111	ماهينور عمرو احمد
1701577	نهال ياسر عبد الوهاب
1701571	نرمين عاطف عبد الكريم
1701636	هبة الله عاطف احمد

Abstract

Hospital Management System

A software system that manages the information of a hospital by integrating all the information regarding patients, doctors, staff, hospital administrative details ... etc. into one software. To facilitate the processes and communication system in the hospital with an easy to use, reliable and secure website.

Table of Contents

1.Introduction:	10
1.1 Purpose:	10
1.2 List of definitions:	10
1.3 Scope:.....	10
1.4 Overview:	10
2.General Description:.....	11
2.1 Product perspective:.....	11
2.2 General capability:.....	11
2.3 General constraints:.....	11
2.4 User characteristics:.....	12
2.5 environment description:.....	12
2.6 Assumptions and Dependencies:.....	12
2.7 Other resources needed:	12
3.System requirements:	13
3.1 Functional requirements:.....	13
3.2 Non-functional requirements:.....	14
4.Requirements validation:.....	15
Stakeholders :.....	15
Source traceability matrix:.....	15
5. Use Case Diagram:.....	18
6.Narrative description and swim lane diagram for use cases:	19
1. Narrative description for use cases:	19
2. Swimlane diagram for use cases:.....	26
• Sign up	26
• Login	27
• Add an Appointment.....	28
• Gets Services/Surgery	29
• Check the Availability of Rooms/Beds.....	30
• Allocate Room/Bed for Patient.....	31
• View the medical history.....	32

• Editing medical history.....	33
• View monthly income.....	33
• Transform patient to another hospital/doctor.....	34
• Transform patient to another room.....	35
• Manage the staff.....	36
• Collect statistics about the patient	37
7.Noun Extraction and CRC cards:.....	38
a. All nouns extracted:.....	38
b. Classes:.....	38
c. CRC:.....	38
8. Class Model	41
9.State diagram:	44
10.Interaction diagram:.....	45
11.Detailed class diagram:.....	54
12. Client-object relation diagram:	55
13.Architectural model:.....	56
13.1. Model (MVC):	56
13.2. Justifications:.....	56
13.2.1. Advantages of MVC architecture:	56
13.2.2. Disadvantages of MVC architecture:.....	57
14.Component diagram:.....	57
15.User Interface design:.....	58
1. Home page:	58
2.Patient page:	63
3. Doctor page:.....	71
4. Manger page:	76
5. Front-desk page:.....	80
6. Finance page:.....	86
16.Detailed design:.....	89
I. Actions in Registration controller:.....	89
II. Actions in Patient controller:.....	92

III.	Actions in Front desk Controller:	99
IV.	Actions in Manager Controller:.....	113
V.	Actions in Finance Controller	124
VI.	Actions in Doctor Controller:.....	129
VII.	Profile controller:.....	140
VIII.	Actions in Lab controller:.....	141
17.	Testing:	148
18.	Estimated cost:.....	179
19.	User guide:.....	184

Table of Figures

Figure 1 Use Case Diagram	18
Figure 2 Sign Up swim lane diagram	26
Figure 3 Login Swim Lane Diagram.....	27
Figure 4 Add an appointment Swim Lane Diagram.....	28
Figure 5 Get Service/Surgery Swim lane diagram.....	29
Figure 6 Check availability of rooms/beds swim lane diagram.....	30
Figure 7 Allocate Room/bed for patient swim lane diagram.....	31
Figure 8 View the medical record swim lane diagram	32
Figure 9 Editing medical history swim lane diagram.....	33
Figure 10 view monthly income swim lane diagram.....	33
Figure 11 Transfer patient to another hospital/doctor swim lane diagram.....	34
Figure 12 Transfer patient to another room swim lane diagram.....	35
Figure 13 Manage the staff swim lane diagram.....	36
Figure 14 Collect statistics about the patient swim lane diagram	37
Figure 15 Class Diagram.....	43
Figure 16 State Diagram	44
Figure 17 Use Case 1: Sign up	45
Figure 18 Use Case 2: Log in	45
Figure 19 Use Case 3: Add an appointment.....	46
Figure 20 Use Case 4: Get services / surgery	46
Figure 21 Use Case 5: Check availability of rooms/beds	47
Figure 22 Use Case 6: Allocate room/bed for patient	47
Figure 23 Use Case 7: View the medical history	48
Figure 24 Use Case 8: Edit the medical history	49
Figure 25 Use Case 9: View monthly income	49
Figure 26 Use Case 10: Transform patient to another hospital/doctor	50
Figure 27 Use Case 11: Transform patient to another room.....	51

Figure 28 Use Case 12: Manage the staff	51
Figure 29 Use Case 13: Collect statistics about the patient.....	52
Figure 30 Use Case 14: Donation request.....	53
Figure 31 Use Case 15: Add lab appointment.....	53
Figure 32 Detailed class diagram.....	54
Figure 33 Client-Object relation diagram	55
Figure 34 Architectural model (MVC).....	56
Figure 35 Component diagram	57
Figure 36 User Interface: Home page.....	58
Figure 37 User Interface: Activation code page	59
Figure 38 User Interface: Registration for Staff page.....	59
Figure 39 User Interface: Registration for patient page.....	60
Figure 40 User Interface: login page	60
Figure 41 User Interface: Donation page.....	61
Figure 42 User Interface: Contact us page.....	62
Figure 43 User Interface: About page	62
Figure 44 User Interface: patient home page	63
Figure 45 User Interface: patient home page services section.....	63
Figure 46 User Interface: patient home page sectors section.....	64
Figure 47 User Interface: add appointment with doctor page	65
Figure 48 User Interface: filling appointment data with doctor page.....	65
Figure 49 User Interface: add appointment in the lab page.....	66
Figure 50 User Interface: add surgery with a doctor page	67
Figure 51 User Interface: pay online page.....	67
Figure 52 User Interface: followed doctors page.....	68
Figure 53 User Interface: patient profile page	69
Figure 54 User Interface: Edit profile of patient page	70
Figure 55 User Interface: doctor home page	71
Figure 56 User Interface: doctor schedule page	71
Figure 57 User Interface: doctor profile page.....	72
Figure 58 User Interface: updating patient's medical data page part1.....	72
Figure 59 User Interface: updating patient's medical data page part2.....	73
Figure 60 User Interface: transferring patient page.....	74
Figure 61 User Interface: transferring patient to another department page.....	74
Figure 62 User Interface: transferring patient to another hospital page	75
Figure 63 User Interface: manager home page part1	76
Figure 64 User Interface: manager home page part2	76
Figure 65 User Interface: search for staff page.....	77
Figure 66 User Interface: managing staff page	77
Figure 67 User Interface: statistics page	78
Figure 68 User Interface: managing specializations page	79
Figure 69 User Interface: manage prices page.....	79

Figure 70 User Interface: adding hospital page	80
Figure 71 User Interface: front desk home page	80
Figure 72 User Interface: room info page.....	81
Figure 73 User Interface: show room reservation page	81
Figure 74 User Interface: check available room page	82
Figure 75 User Interface: Search page.....	82
Figure 76 User Interface: Room info page.....	83
Figure 77 User Interface: surgery rooms page.....	83
Figure 78 User Interface: rooms page	84
Figure 79 User Interface: Room details page.....	84
Figure 80 User Interface: cancel reservation page.....	85
Figure 81 User Interface: finance home page.....	86
Figure 82 User Interface: financial statistics page.....	87
Figure 83 User Interface: services financial details page.....	88
Figure 84 User Interface: patient bill page	88
Figure 85 Testing: Sign up as patient (Activation code)	148
Figure 86 Testing: Sign up as patient (Filling information)	148
Figure 87 Testing: login as patient	149
Figure 88 Testing: after patient login	149
Figure 89 Testing: Enter wrong email format and password not matching during sign up	150
Figure 90 Testing: Login with the wrong email or password.....	151
Figure 91 Testing: Sectors in our hospital.....	151
Figure 92 Testing: choose doctor	152
Figure 93 Testing: choose to pay online or offline	152
Figure 94 Testing: pay offline	153
Figure 95 Testing: Doctor's list.....	154
Figure 96 Testing: Doctor's Schedule	155
Figure 97 Testing: select doctor and pay online	155
Figure 98 Testing: online payment.....	156
Figure 99 Testing: online payment successfully recorded.....	156
Figure 100 Testing: choose lab.....	157
Figure 101 Testing: Filling lab scan request	157
Figure 102 Testing: Patient Profile	158
Figure 103 Testing: edit patient profile	158
Figure 104 Testing: patient profile	159
Figure 105 Testing: doctor login.....	160
Figure 106 Testing: doctor home page	160
Figure 107 Testing: Updating patient's medical data.....	161
Figure 108 Testing: After updating health progress & status	162
Figure 109 Testing: Transferring patient	162
Figure 110 Testing: Transferring patient to another doctor.....	163

Figure 111 Testing: The doctor whom the patient is transferred to is shown in the followed doctor list of the patient	163
Figure 112 Testing: Show doctor schedule	164
Figure 113 Testing: show Doctor Profile.....	164
Figure 114 Testing: View Emergency beds and blood units	165
Figure 115 Testing: login as a lab specialist	165
Figure 116 Testing: View lab requests.....	166
Figure 117 Testing: Updating lab request status part1	166
Figure 118 Testing: Updating lab request status part2	167
Figure 119 Testing: Sort requests by date ascending.....	167
Figure 120 Testing: Search for a certain request part1	168
Figure 121 Testing: Search for a certain request part2	168
Figure 122 Testing: add new blood units part1	169
Figure 123 Testing: add new blood units part2	169
Figure 124 Testing: Show the bill of the patient in finance	170
Figure 125:Total income	170
Figure 126 Testing: Finance collects the outcome for the hospital during a certain period	171
Figure 127 Testing: The manager view transfer hospital	171
Figure 128 Testing: The manager adds transfer hospital part1	171
Figure 129 Testing: The manager adds transfer hospital part2	172
Figure 130Testing : Add Specialization	172
Figure 131 Testing :Add service	172
Figure 132: Delete specialization	173
Figure 133:Delete Service	173
Figure 134: Edit Price.....	173
Figure 135 Testing: Total income	174
Figure 136 Testing :Income for Examination Service	174
Figure 137 Testing: Staff Information.....	175
Figure 138 Testing: Login as a front desk.....	175
Figure 139 Testing: front desk home page	176
Figure 140 Testing: Choose reserve room part1.....	176
Figure 141 Testing: Choose reserve room part2.....	176
Figure 142 Testing: reserve room part1	177
Figure 143 Testing: reserve room part2	177
Figure 144 Testing :Transfer Patient	177
Figure 145 Testing: search for patient.....	178
Figure 146 Testing : Cancel	178
Figure 147 User Guide: Home Page.....	184
Figure 148 User Guide: About Page	185
Figure 149 User Guide: Activation Code Page	185
Figure 150 User Guide: Patient Registration Page	186
Figure 151 User Guide: Login Page.....	186

Figure 152 User Guide: Donation Page	187
Figure 153 User Guide: Patient Home Page.....	187
Figure 154 User Guide: Patient Home Page Services Section	188
Figure 155 User Guide: Patient Home Page Hospital Sectors Section	188
Figure 156 User Guide: Add appointment with doctor Page Part1	189
Figure 157 User Guide: Add appointment with doctor Page Part2	190
Figure 158 User Guide: Online Payment Page	190
Figure 159 User Guide: Offline Payment Page.....	191
Figure 160User Guide: Add appointment with lab Page Part1	192
Figure 161User Guide: Add appointment with lab Page Part2	193
Figure 162User Guide: Add Surgery with doctor Page	194
Figure 163 User Guide: Followed Doctors in Home Page	194
Figure 164 User Guide: Patient Profile Page.....	195
Figure 165 User Guide: Patient Edit Profile Page.....	195
Figure 166 User Guide: Manager Home Page part1	196
Figure 167 User Guide: Manager Home Page part2	196
Figure 168 User Guide: Manager Search for Staff Page	197
Figure 169 User Guide: Manager Manage Staff Page	198
Figure 170 User Guide: Manager Statistics Page	198
Figure 171 User Guide: Manager Manage Specializations Page.....	199
Figure 172 User Guide: Manager Add Specialization Page.....	199
Figure 173 User Guide: Manager Delete Specialization Page.....	200
Figure 174 User Guide: Manager Manage Prices Page	200
Figure 175 User Guide: Manager Add Service Page.....	201
Figure 176 User Guide: Manager Delete Service Page.....	201
Figure 177 User Guide: Manager Edit Price Page	202
Figure 178 User Guide: Manager View Hospital Page.....	202
Figure 179 User Guide: Manager Add Hospital Page	203
Figure 180 User Guide: Front Desk Home Page	203
Figure 181User Guide: Front Desk Check Room Availability Page	204
Figure 182 User Guide: Front Desk Check Surgery Room Availability Page part1	204
Figure 183 User Guide: Front Desk Check Surgery Room Availability Page part2	205
Figure 184 User Guide: Front Desk Reserve Room Page	206
Figure 185 User Guide: Front Desk Transfer Patient Page	206
Figure 186 User Guide: Front Desk Search Page.....	207
Figure 187 User Guide: Front Desk Room Info Page	208
Figure 188 User Guide: Front Desk Surgery Room Info Page	208
Figure 189 User Guide: Front Desk Rooms Info Page.....	209
Figure 190 User Guide: Front Desk Room Details Page.....	209
Figure 191 User Guide: Front Desk Cancel Reservation Page	210
Figure 192 User Guide: Finance Home Page.....	211
Figure 193 User Guide: Finance Statistics Page	212

Figure 194 User Guide: Finance Service Income Details Page	213
Figure 195 User Guide: Finance Service Patient Bill Page	213
Figure 196 User Guide: Doctor Home Page	214
Figure 197 User Guide: Doctor Profile Page	214
Figure 198 User Guide: Doctor Schedule Page	215
Figure 199 User Guide: Doctor Emergency Page	215
Figure 200 User Guide: Doctor Patient Follow Up Page Part1.....	216
Figure 201 User Guide: Doctor Patient Follow Up Page Part2.....	216
Figure 202 User Guide: Doctor Transfer Patient Page	217
Figure 203User Guide: Doctor Transfer Patient to Another Doctor Page	217
Figure 204 User Guide: Doctor Transfer Patient to Another Hospital Page	218
Figure 205 User Guide: Lab Home Page	218
Figure 206 User Guide: Lab Patient Follow Up Page	219
Figure 207 User Guide: Blood Bank View and Edit Blood Units Page.....	220

Table of Tables

Table 1 Source traceability matrix.....	15
Table 2 Function Point.....	179
Table 3 Function Point Estimation	180
Table 4 Scale Factor Estimation	181
Table 5 Effort Adjustment Factors	182

1. Introduction:

1.1 Purpose:

The goal of this document is to clarify the functionality of our hospital management software system and its features to the reader by showing different diagrams that describe our systems in details. It shows UML diagrams and UI design to describe our system and shows all the supported features by the system and how to use it.

1.2 List of definitions:

We have some abbreviations used in the document as:

UI: User interface

CRC: (class, responsibility, collaboration)

It is one of UML diagrams that shows the responsibility of a class and any collaboration between it and other classes.

1.3 Scope:

“Ain Shams hospital” is our product name. It a software system to handle main hospital services like searching doctors, reserving appointments. It also allows doctors to transfer patients to other doctors over the internet without annoying the patient by transferring their files in the traditional way. Data is secured for the purpose of confidence as patient’s information is stored and shared only with the appropriate users who has the access to view this information.

1.4 Overview:

The document includes all the functional and non-functional requirements of the system, UML notations, user interface and showing how the user can deal with the system. It also includes screenshots for our system and results of testing of our project.

2.General Description:

2.1 Product perspective:

Our software system could be considered an example for hospital systems as (Saudi German hospital website) that help in creating easy and user-friendly interface for both hospital staff and patients.

2.2 General capability:

Patients, doctors, managers and all hospital staff register on the system.

Patients can follow doctors and book appointments with doctors.

Doctors can view data of their patients and view their medical records.

Doctors can edit the medical records of their patients and their health progress.

Doctors can transfer their patients to other hospitals or other departments in our hospital.

Doctors can view the data of the blood bank.

Manager can delete or add a new doctor.

Manager can add new specialization or delete an existing one.

Manager can add a new hospital to the system so that we doctors can transfer patients there.

Front-desk employee can check available rooms and reserve rooms.

Front-desk employee can transfer a patient from one room to another one.

2.3 General constraints:

- I. Users should have an account to be able to use the services of our system.
i.e. if a patient doesn't have an account he would not be able to book an appointment with any doctor.
- II. On signing up, hospital staff must write their activation code according to their specialization in order to be recognized by the system so they can be provided by the appropriate services and a patient writes 0000 as the activation code.

- a) The activation code of doctors starts by digit 1. However, the rest digits change according to the specialization.
 - b) The activation code of the manager starts by digit 2.
 - c) The activation code of a lab specialist starts by digit 3.
 - d) The activation code of a finance employee starts by digit 4.
 - e) The activation code of a front desk starts by digit 5.
- III. If a doctor wants to reserve an appointment with another doctor as a patient, he should create another account as a patient to be able to get the services of patients.
- IV. All lab specialist's login to our system with the same account.
- V. All radiology specialist's login to our system with the same account.
- VI. Manager can only add a new doctor to the system database but can not add any other employee.

2.4 User characteristics:

The user is a patient or one of the hospital staff.

2.5 environment description:

The user uses the system in a computer platform.

2.6 Assumptions and Dependencies:

We assume that each member in hospital staff knows his activation code.

2.7 Other resources needed:

We used Asp.Net Core framework and used NuGet packages as :

- 1) BCrypt.Net-Next
- 2) Microsoft.EntityFrameworkCore
- 3) Microsoft.EntityFrameworkCore.Design
- 4) Microsoft.EntityFrameworkCore.SqlServer
- 5) Microsoft.VisualStudio.Web.CodeGeneration

3.System requirements:

3.1 Functional requirements:

- Doctors can register on the system.
- Doctors can add their professional information.
- Doctors can add/edit their schedules.
- Doctors can find their patients and view/edit/add their medical records.
- Doctors can transfer a patient to another doctor or another hospital.
- Doctors can collect statistics regarding his patients.
- In the Emergency section the doctor can have access to view the blood bank and book a surgery room
- Patients can register on the system.
- Patients can add the doctor(s) they are following up with.
- Patients can view different services offered by the hospital.
- Patients can book appointments with doctors or specific services.
- Patients can find and pay the fees for the service he has.
- Front-Desk can register on the system.
- Front-Desk can check the availability of rooms/beds.
- Front-Desk can allocate room/bed to a patient.
- Front-Desk can transform a patient from one ward to another.
- Finance employees can register on the system.
- Finance employees can organize payment processes.
- Radiology specialists can register on the system.
- Radiology specialists can view/ add/edit their schedules.

- Blood bank specialists can register on the system.
- Blood bank specialists can view/edit/add on blood bank.
- Lab specialists can register on the system.
- Lab specialists can view/ add/edit their schedules.
- The hospital manager can register on the system.
- The hospital manager can delete/view staff from data.
- The hospital manager monitors the regularity of work in the hospital.

3.2 Non-functional requirements:

- Security: Any user who uses the system shall have a Login ID and Password.
- Reliability: (Availability) The system shall be available all the time.
- Maintainability: (Back Up) The system shall provide the capability to back-up the Data (Errors) The system shall keep a log of all the errors.
- Portability: The System must be compatible with both Mobile and Computer devices, so it can be used and accessed at all times.
- The software should have a response time of at most 100 ms.
- If the software fails for any reason, it should recover within 5 minutes.
- It shall be implemented in framework asp.net core.
- The project will have a limited budget of 50000.
- A development span of 1 year.
- Follows all the applicable copyright laws.

4. Requirements validation:

Stakeholders :

- a. Doctor
- b. Patient
- c. Manager
- d. Front-Desk employee
- e. Finance employee
- f. Lab specialist
- g. Radiology specialist
- h. Blood bank specialist
- i. Project manager
- j. Developer

Source traceability matrix:

Table 1 Source traceability matrix

Task	Doctor	Patient	Manager	Front- Desk employee	Finance employee	Lab specialist	Radiology specialist	Blood bank specialist	Project manager	Developer
1	@		@							
2	@									
3	@	@	@							
4	@	@								
5	@	@								
6	@	@								

7	@			@				@		
8		@								
9	@	@								
10		@	@							
11	@	@								
12		@			@					
13			@	@						
14				@						
15		@		@						
16		@		@						
17			@		@					
18		@			@					
19							@			
20		@	@				@			
21			@					@		
22								@		
23			@			@				
24		@	@			@				
25			@							

26	@		@	@	@	@	@	@		
27			@							
28	@	@	@	@	@	@	@	@		@
29	@	@	@	@	@	@	@	@	@	@
30									@	@
31									@	@
32									@	@
33									@	@
34									@	@
35									@	
36									@	
37									@	

5. Use Case Diagram:

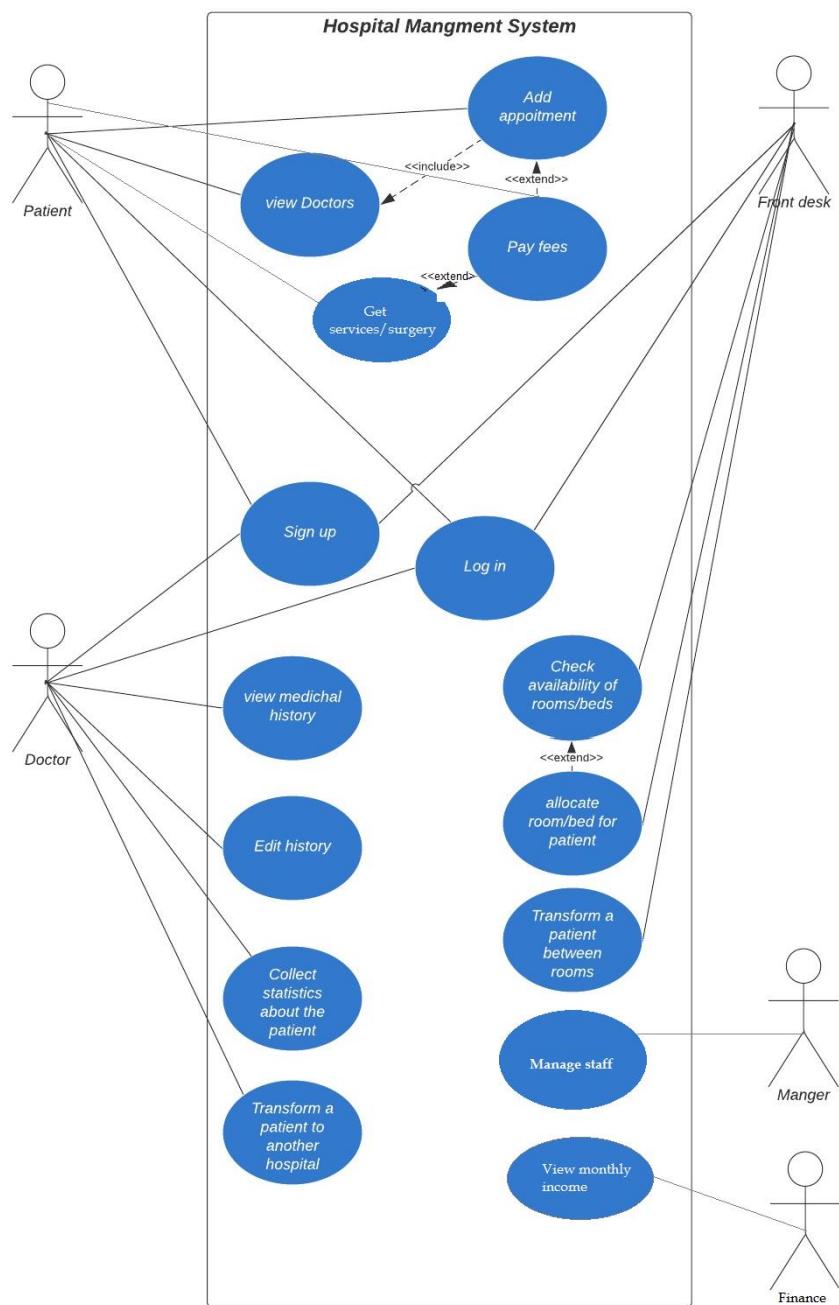


Figure 1 Use Case Diagram

6.Narrative description and swim lane diagram for use cases:

1. Narrative description for use cases:

Use Case 1: Sign up

Use case name: Sign Up.

Goal in context: register in the website.

Preconditions: User enters his information.

Successful end condition: User signs up in the website.

Failed end condition: missing information so sign up failed.

Primary actor: Doctor / Patient / Front desk / Finance.

Secondary actor: None.

Main flow:

User visits the hospital website.

Registration page appear for the user.

User enters his information.

Sign up successfully.

Use Case 2: Log in

Use case name: log in.

Goal in context: log in to the website.

Preconditions: User enters the right Email and password.

Successful end condition: User logs in.

Failed end condition: User can't log in.

Primary actor: Doctor / Patient / Front desk / Finance.

Secondary actor: None.

Main flow:

User opens Log in page.

User enters his right Email and password.

User log in successfully.

Use Case 3: Add an appointment

Use case name: Add an appointment.

Goal in context: Patient adds an appointment.

Preconditions: patient views the available appointments.

Successful end condition: patient finds suitable appointment.

Failed end condition: patient can't find an appointment.

Primary actor: Patient.

Secondary actor: Doctor.

Main flow:

Patient views different appointments.

Patient chooses an appointment.

Extensions:

Pay fees for the appointment.

Use Case 4: Get services / surgery

Use case name: Get services / surgery.

Goal in context: Patient gets the service.

Preconditions: User chooses a service.

Successful end condition: patient finds suitable appointment.

Failed end condition: patient can't find suitable service.

Primary actor: Patient.

Secondary actor: Doctor.

Main flow:

Patient views different services.

Patient chooses a service.

Patient gets the service.

Extensions:

Pay fees of the service/surgery.

Use Case 5: Check availability of rooms/beds

Use case name: Check availability of rooms/beds.

Goal in context: see the available rooms and beds.

Preconditions: Patient asks for room/bed.

Successful end condition: Front desk knows the available rooms/beds.

Failed end condition: Front desk can't know what the available rooms/beds are.

Primary actor: Front-desk.

Secondary actor: Patient.

Main flow:

Patient asks for room/bed.

Front-desk searches for available rooms/beds.

Front-desk gets the list of the available rooms/beds.

Extensions:

Allocate room/bed for patient.

Use Case 6: Allocate room/bed for patient

Use case name: Allocate room/bed for patient.

Goal in context: Allocate room/bed for patient.

Preconditions: Patient asks for room/bed.

Successful end condition: Patient gets the room/bed.

Failed end condition: Can't find available room/bed.

Primary actor: Front-desk.

Secondary actor: Patient.

Main flow:

Patient asks for room/bed.

Front-desk searches for available rooms/beds.

Front-desk gets the list of the available rooms/beds.

Front-desk allocates a room/bed for the patient

Use Case 7: View the medical history

Use case name: View the medical history.

Goal in context: Doctor gets the patient's medical history.

Preconditions: Doctor asks for the medical history.

Successful end condition: medical history issued to the doctor.

Failed end condition: medical history doesn't issue to the doctor.

Primary actor: Doctor.

Main flow:

Doctor asks for patient's medical history.

Medical history issued to the doctor.

Use Case 8: Edit the medical history

Use case name: Edit the medical history.

Goal in context: Doctor edits the patient's medical history.

Preconditions: Doctor asks for the medical history.

Successful end condition: medical history successfully edited.

Failed end condition: doctor can't edit the medical history.

Primary actor: Doctor.

Main flow:

Doctor asks for patient's medical history.

Doctor adds new information for patient's medical history.

Patient's medical history successfully edited

Use Case 9: View monthly income

Use case name: View monthly income.

Goal in context: View the monthly income and fees of the hospital.

Preconditions: thought about the monthly income of the hospital.

Successful end condition: manger sees the curve of the incomes.

Failed end condition: can't get the curve of the hospital incomes.

Primary actor: Finance.

Main flow:

Finance views the monthly incomes to the hospital

Use Case 10: Transform patient to another hospital/doctor

Use case name: Transform patient to another hospital/doctor.

Goal in context: Transfer patient to another hospital/doctor.

Preconditions: patient wants to change the doctor or leave the hospital.

Successful end condition: patient's file transfers to other hospital/doctor.

Failed end condition: can't transfer the patient.

Primary actor: Doctor.

Secondary actor: Patient.

Main flow:

Patient wants to change the doctor or leave the hospital

Doctor chooses the second hospital.

Patient's file transfers to other hospital/doctor

Use Case 11: Transform patient to another room

Use case name: Transform patient to another room.

Goal in context: Transfer patient to another room.

Preconditions: patient wants to change his room.

Successful end condition: patient transfers to another room.

Failed end condition: No available rooms to transfer to.

Primary actor: Front-Desk.

Secondary actor: Patient.

Main flow:

Patient wants to change his room.

Front-desk searches for another available room.

Patient changes his room.

2. Swimlane diagram for use cases:

- Sign up

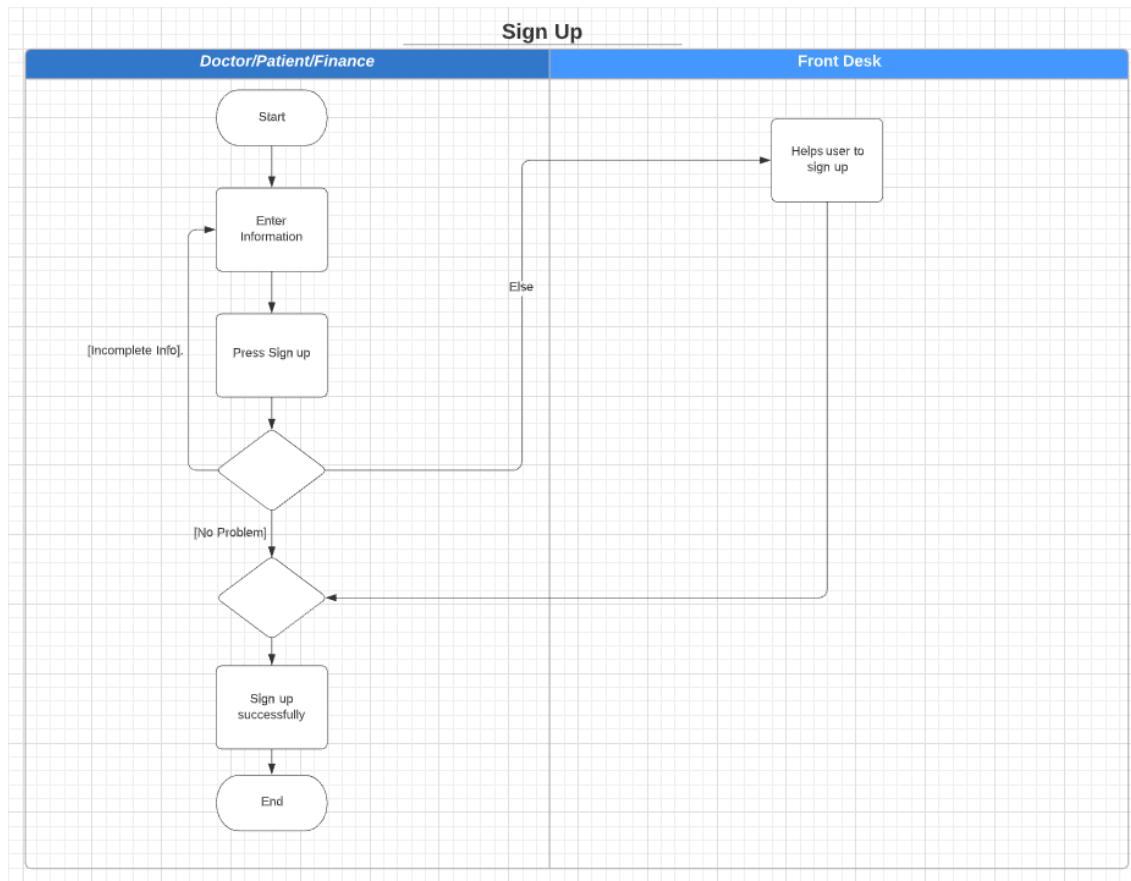


Figure 2 Sign Up swim lane diagram

- Login

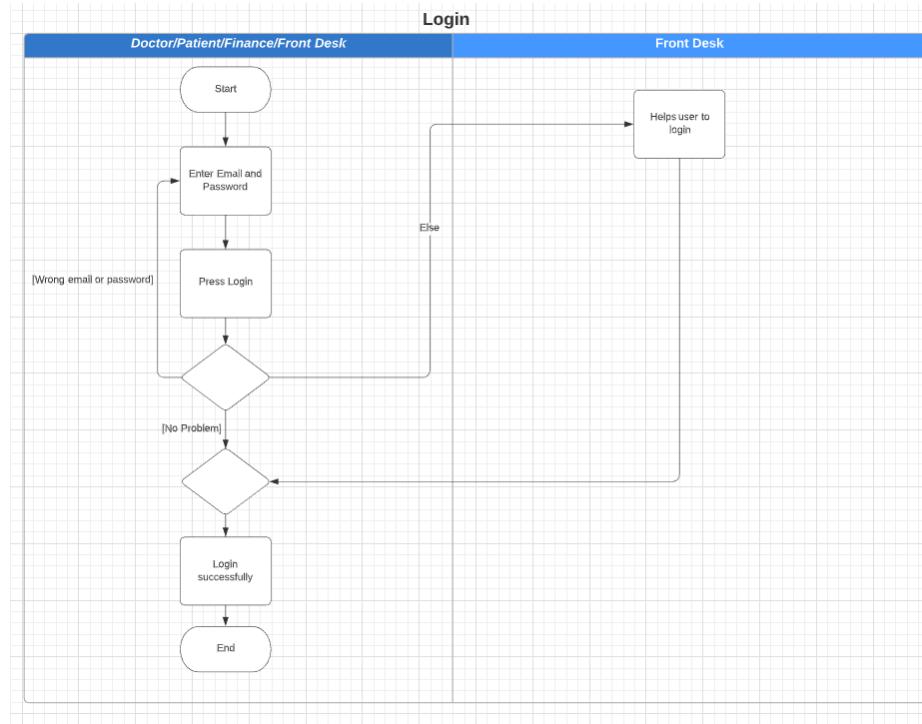


Figure 3 Login Swim Lane Diagram

- Add an Appointment.

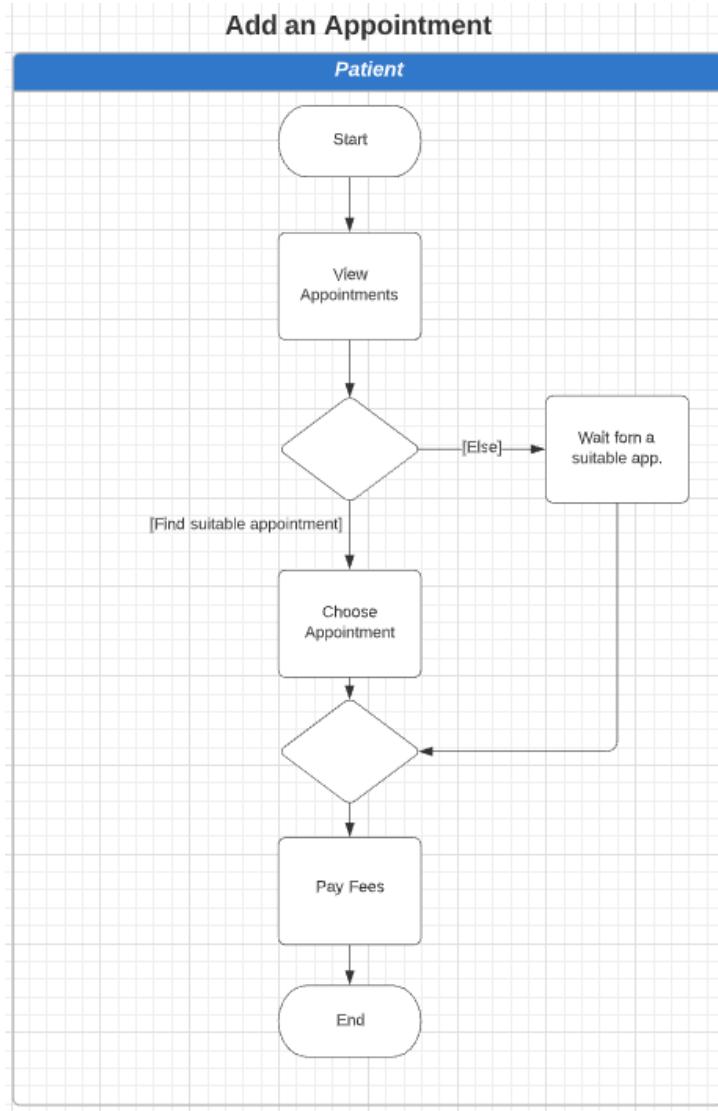


Figure 4 Add an appointment Swim Lane Diagram

- Gets Services/Surgery

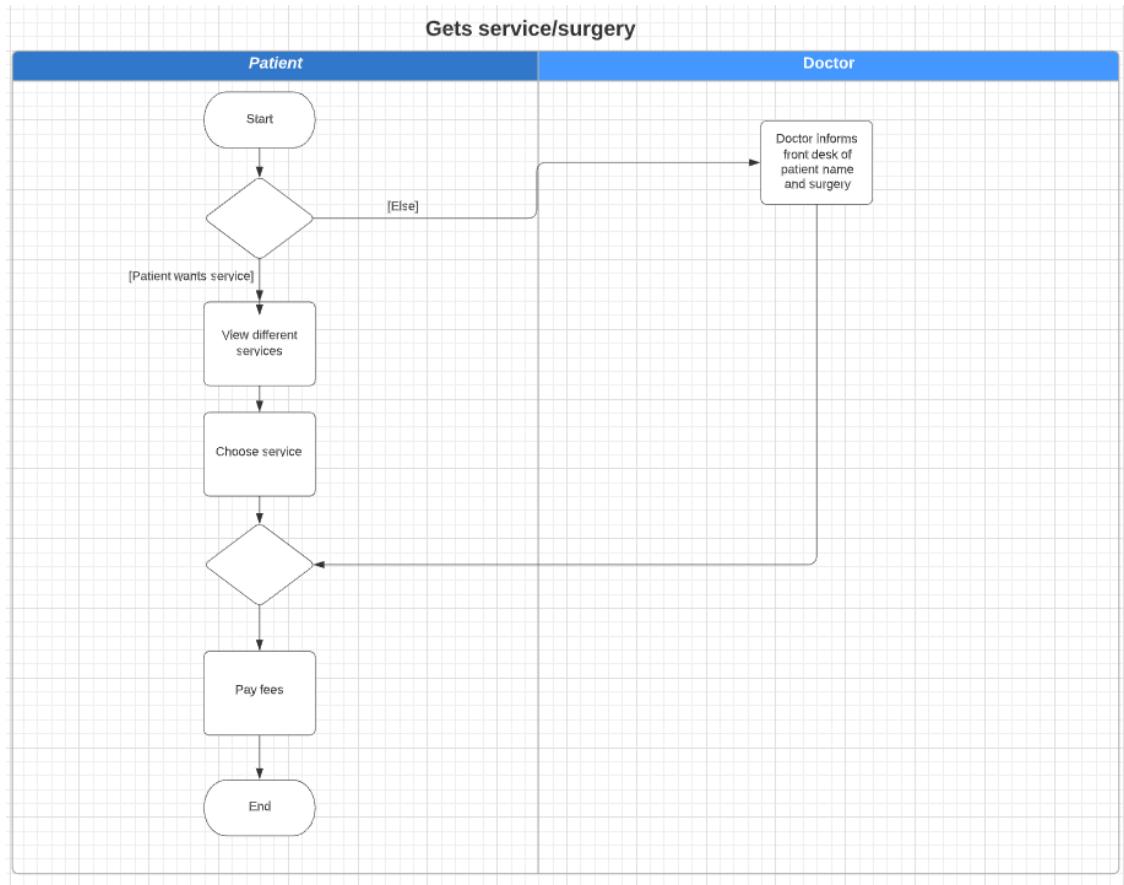


Figure 5 Get Service/Surgery Swim lane diagram

- Check the Availability of Rooms/Beds

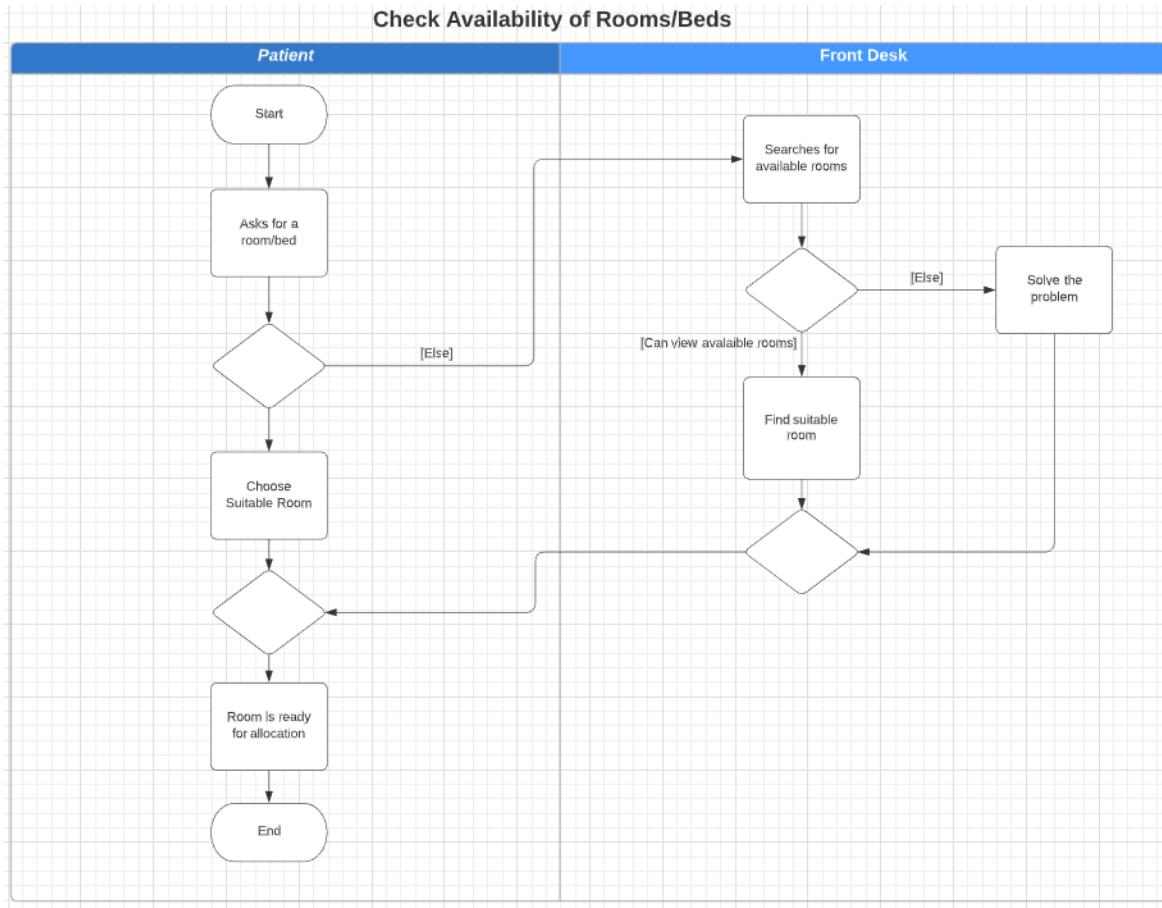


Figure 6 Check availability of rooms/beds swim lane diagram

- Allocate Room/Bed for Patient

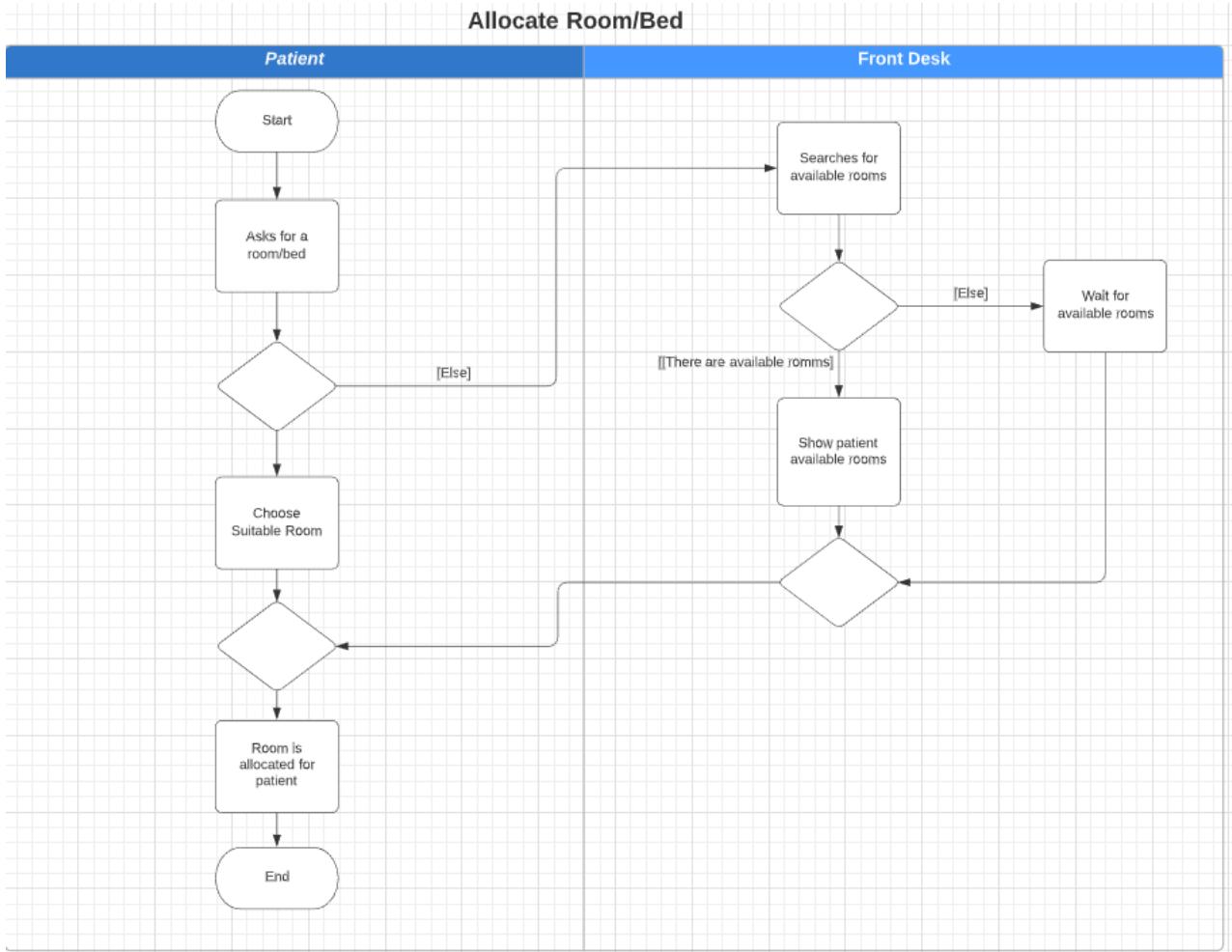


Figure 7 Allocate Room/bed for patient swim lane diagram

- View the medical history.

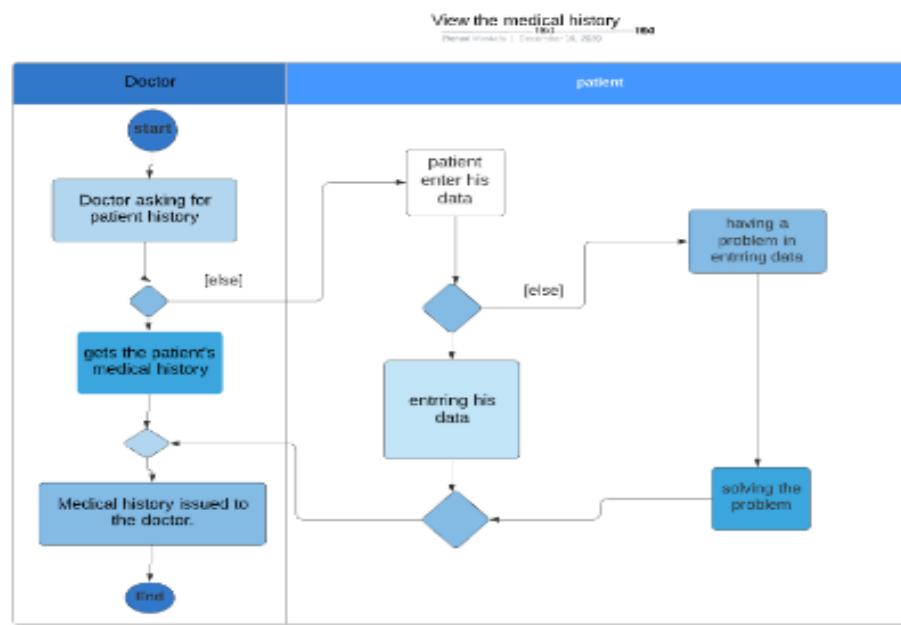


Figure 8 View the medical record swim lane diagram

- Editing medical history.

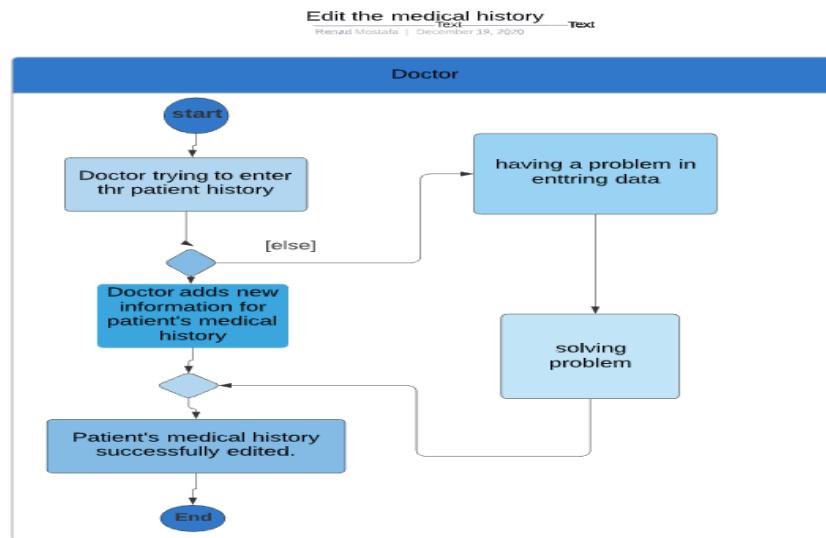


Figure 9 Editing medical history swim lane diagram

- View monthly income.

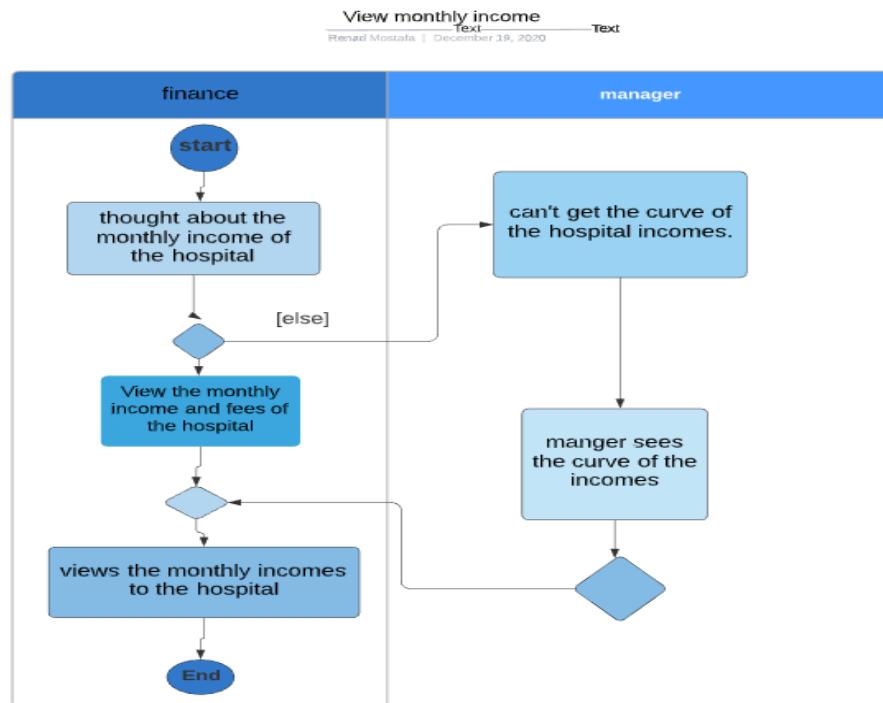


Figure 10 view monthly income swim lane diagram

- Transform patient to another hospital/doctor.

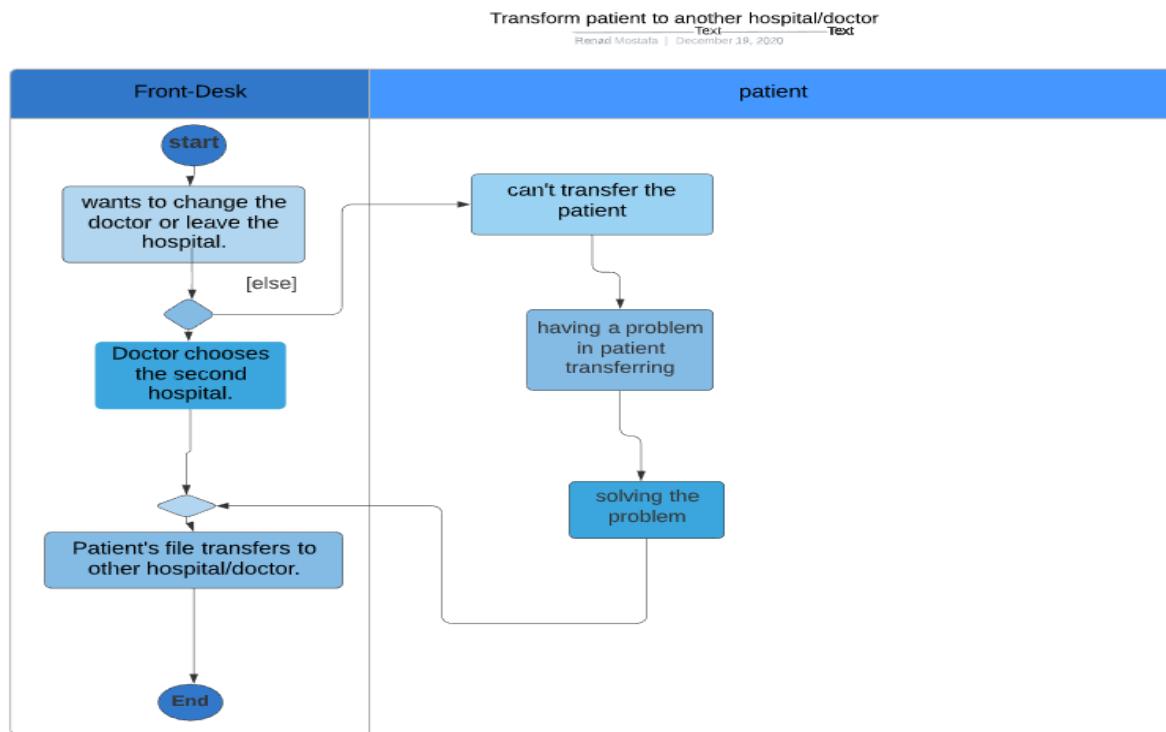


Figure 11 Transfer patient to another hospital/doctor swim lane diagram

- Transform patient to another room.

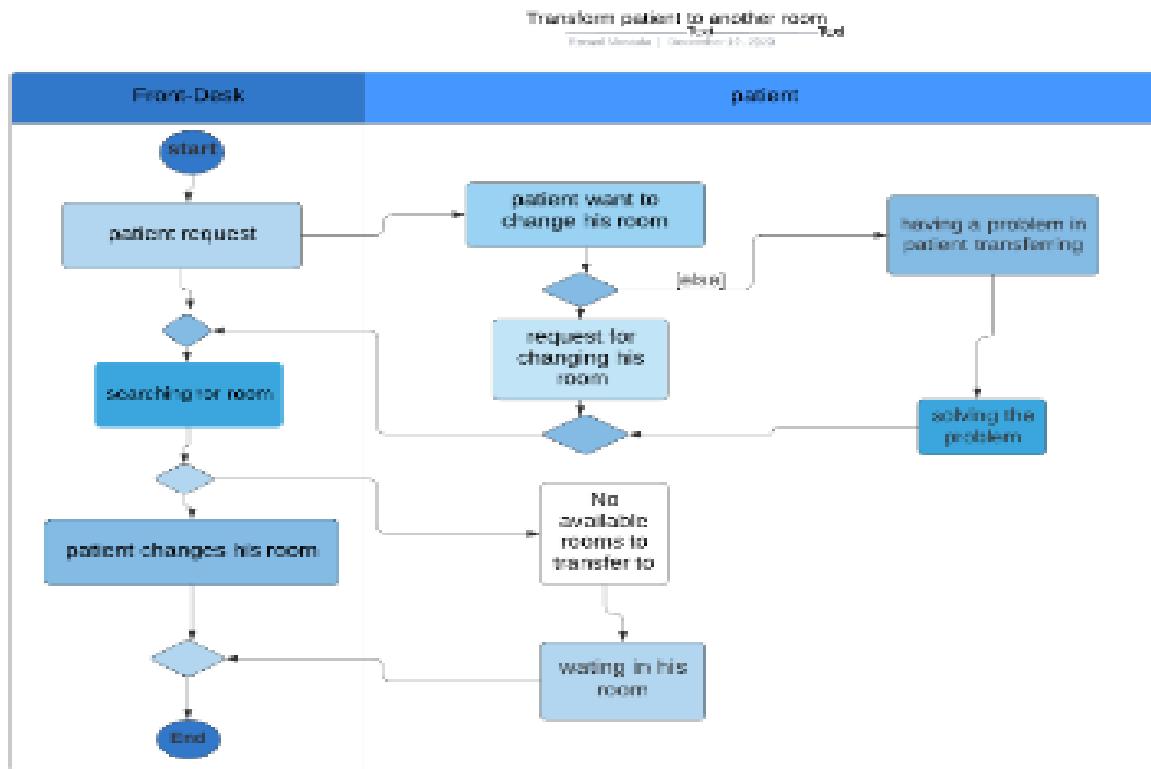


Figure 12 Transfer patient to another room swim lane diagram

- Manage the staff.

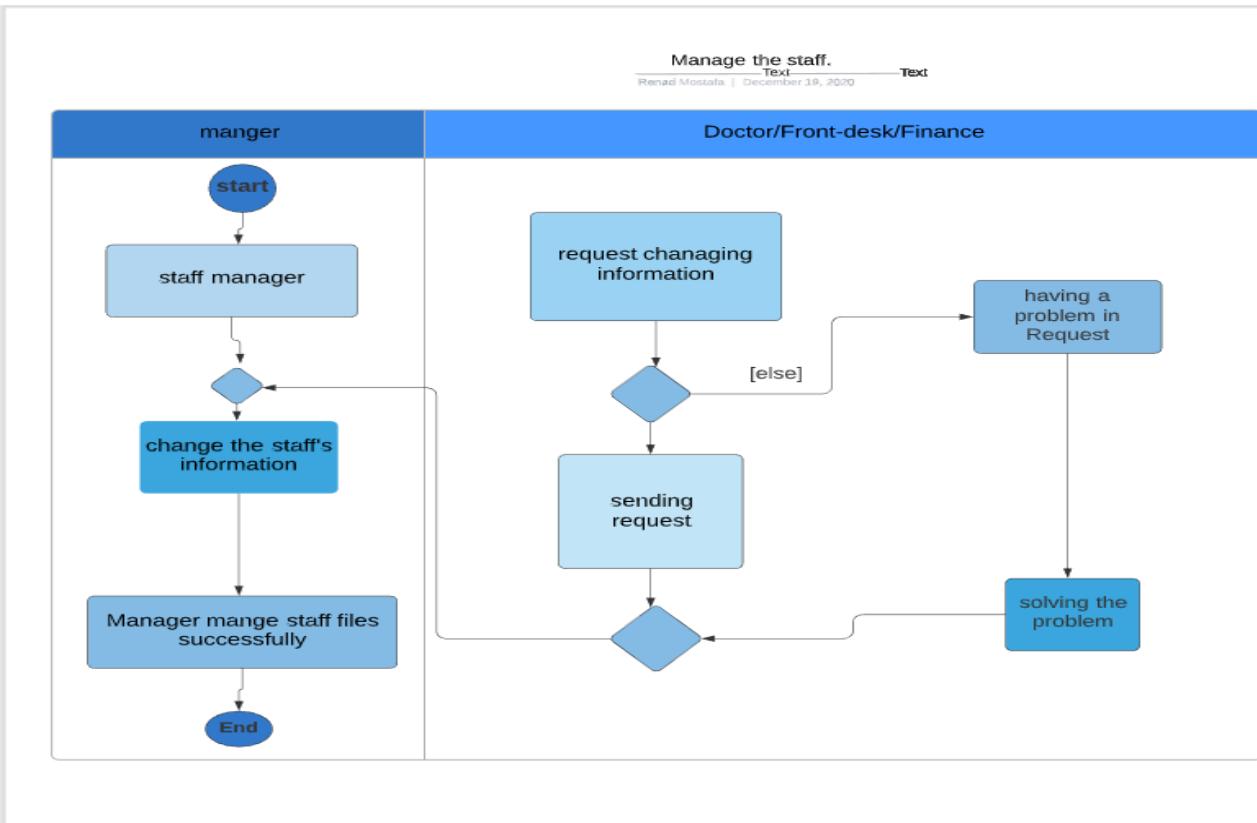


Figure 13 Manage the staff swim lane diagram

- Collect statistics about the patient

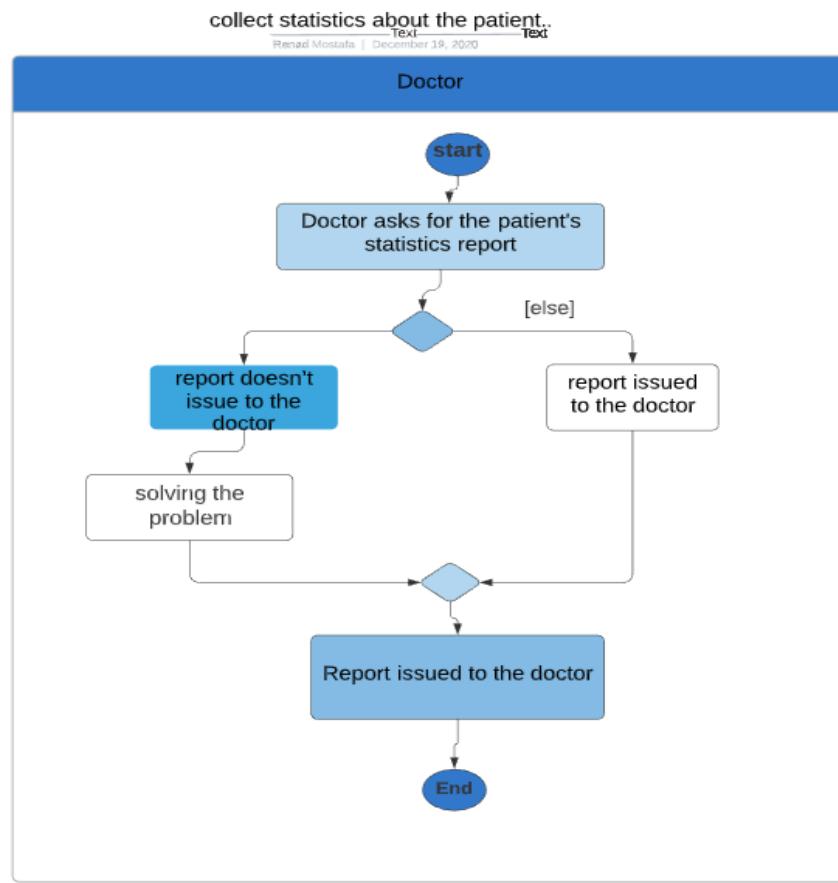


Figure 14 Collect statistics about the patient swim lane diagram

7.Noun Extraction and CRC cards:

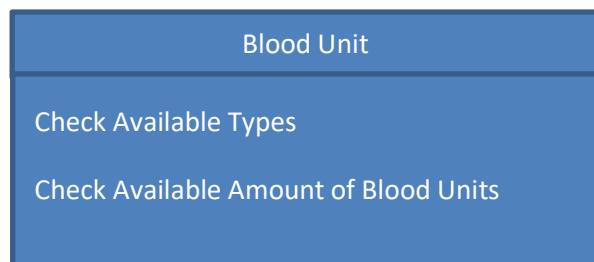
a. All nouns extracted:

Hospital, patients, doctors, staff, administrators, schedules, medical records, statistics, services, departments, appointments, payments, front-desk, rooms, beds, hospital manager, specialists (radiology specialist, lab specialist), emergency department, finance employee

b. Classes:

Blood Unit
Donation
Facility Reservation
Follow Up
Follow Ups History
Follow Ups Type
Hospital Facility
Patient
Payment
Registration
Specialization
Staff
Staff Schedule
Transfer Hospital

c. CRC:



Donation

Get Volunteer's Phone Number

Get Volunteer's Name

Get Volunteer's mail To Contact Him

Add The Blood To The Blood Bank

Blood Unit

Facility Reservation

Get The Transfer Start And End Time

Connect The Staff With Patient

Staff , Patient

Follow Up

Manage The Follow Up Status

Connect The Doctor With The Patient

Staff , Patient

Follow Up History

Get The Follow Up Start And End Time

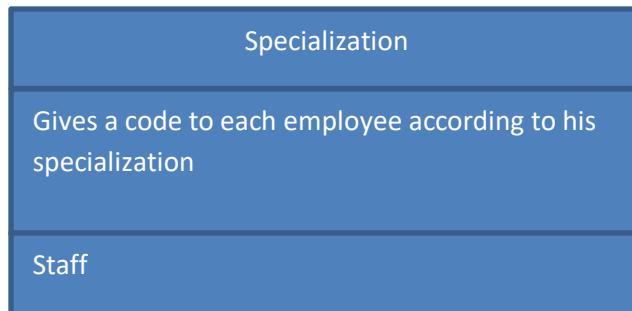
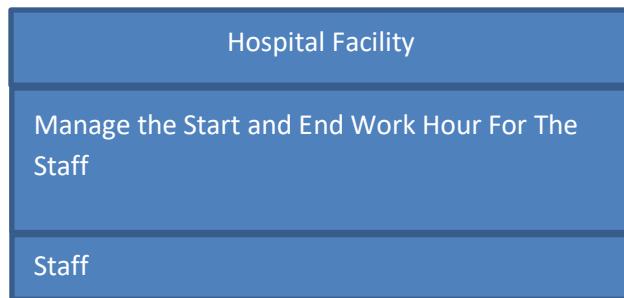
Connect The Staff With Patient

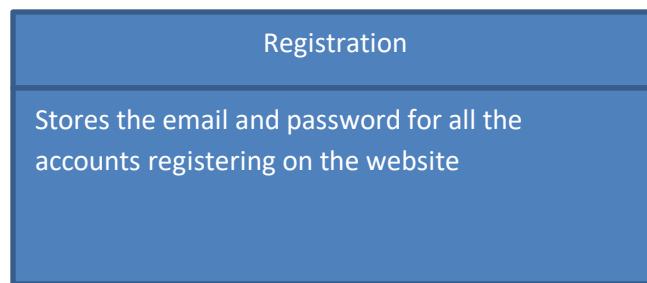
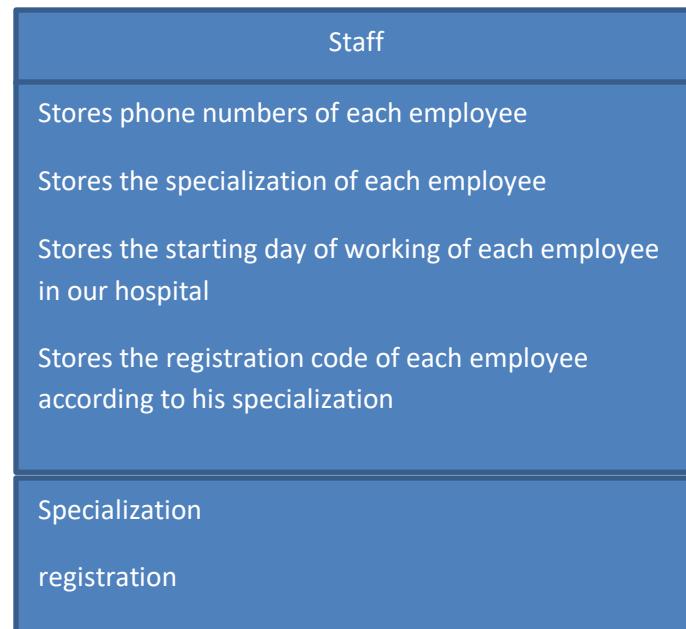
Follow Up , Follow Up Type

Payment

The patient pay the fees according to the services he got

Patient





8. Class Model

- Facility Reservation includes Patient data (Aggregation)
- Facility Reservation includes Staff data (Aggregation)
- Facility Reservation includes Hospital Facility (Aggregation)
- Follow Up includes Patient data (Aggregation)
- Follow Up includes Staff data (Aggregation)
- Patient makes Payment (Association)
- Registration contains Patient data (Composition)
- Registration contains Staff data (Composition)
- Patient can be transferred to Transfer Hospital (Association)
- Specialization is assigned to Staff (Composition)
- For each Specialization a Staff Schedule (Association)
- Follow Up has Follow Up History (Composition)
- Follow Up History has Follow Up Type (Aggregation)

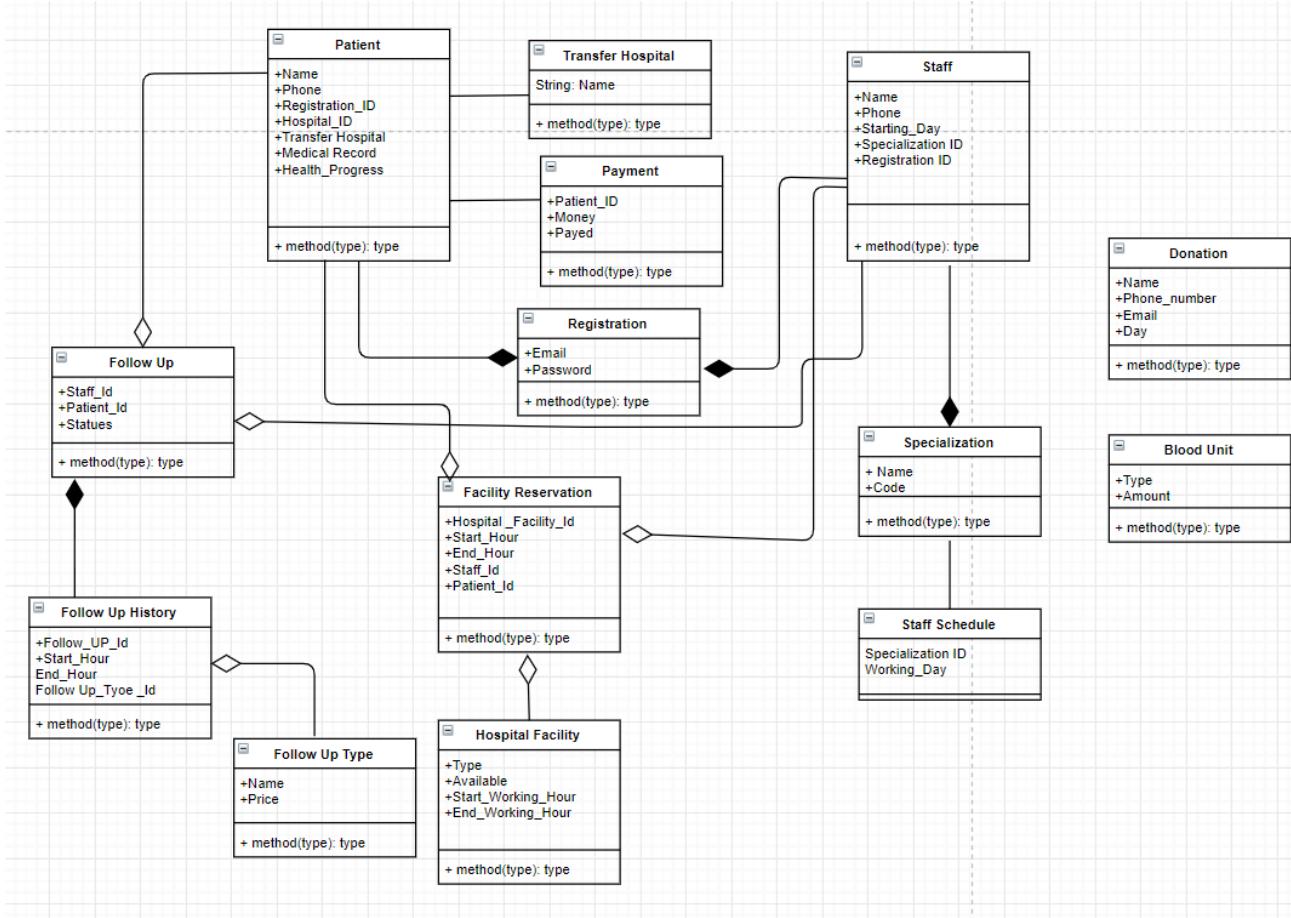


Figure 15 Class Diagram

9.State diagram:

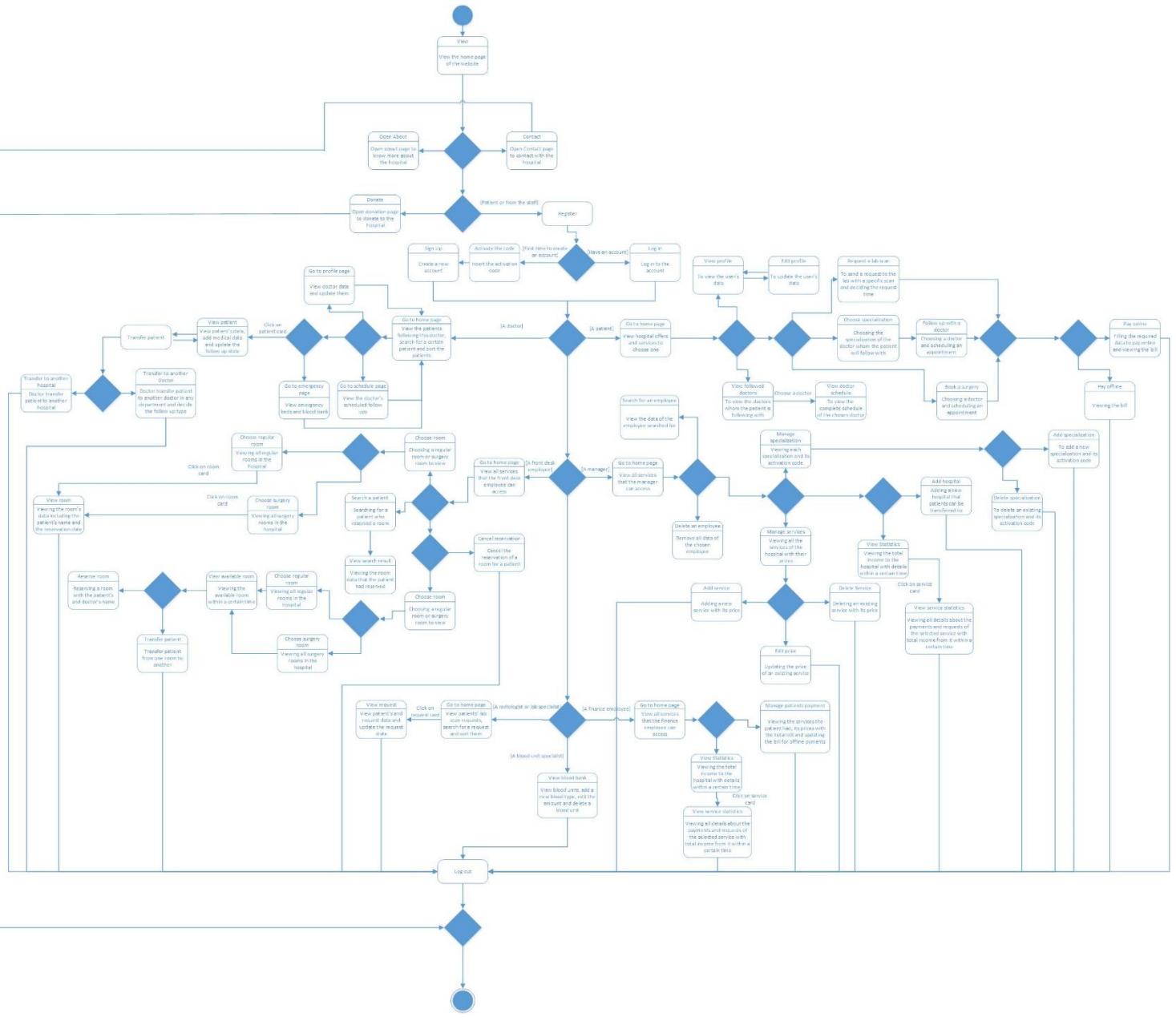


Figure 16 State Diagram

10. Interaction diagram:

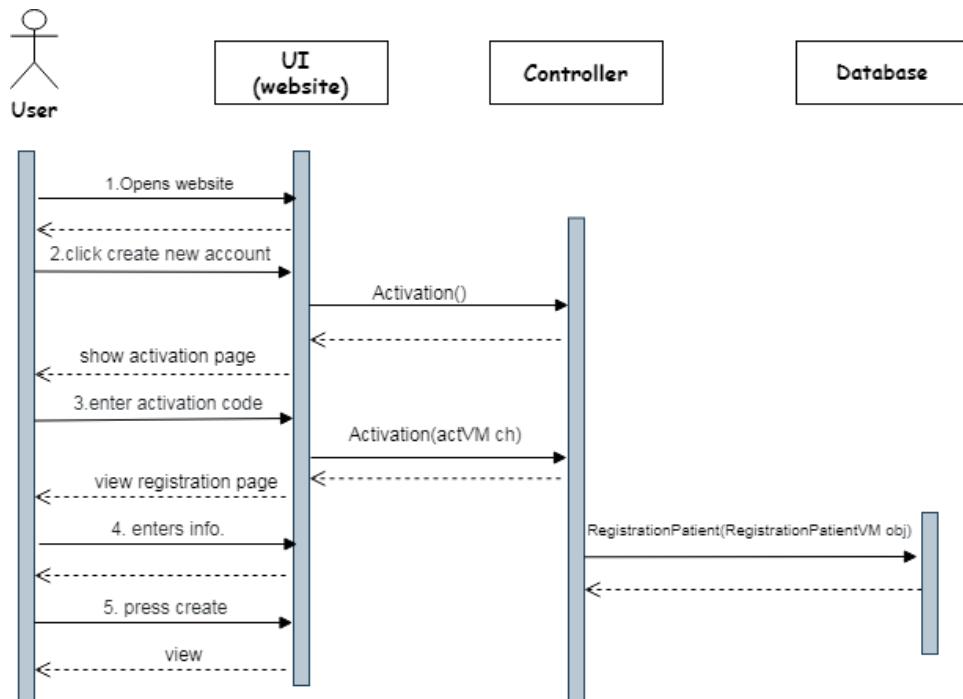


Figure 17 Use Case 1: Sign up

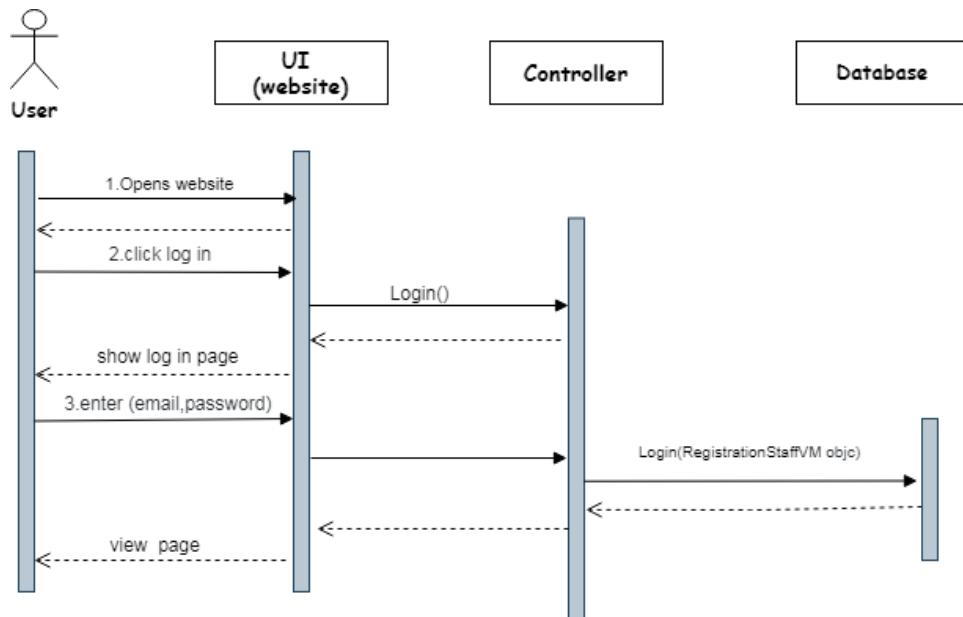


Figure 18 Use Case 2: Log in

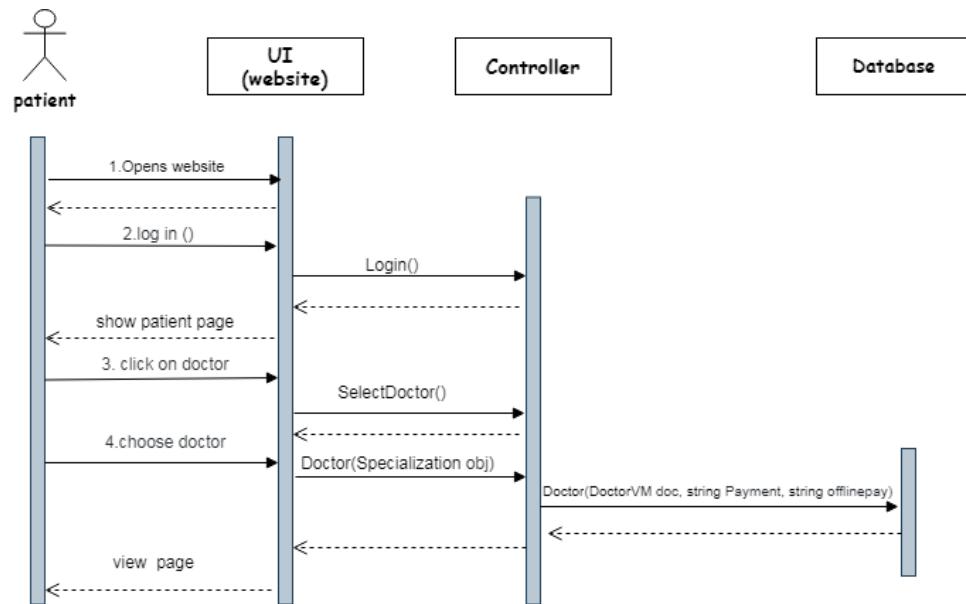


Figure 19 Use Case 3: Add an appointment

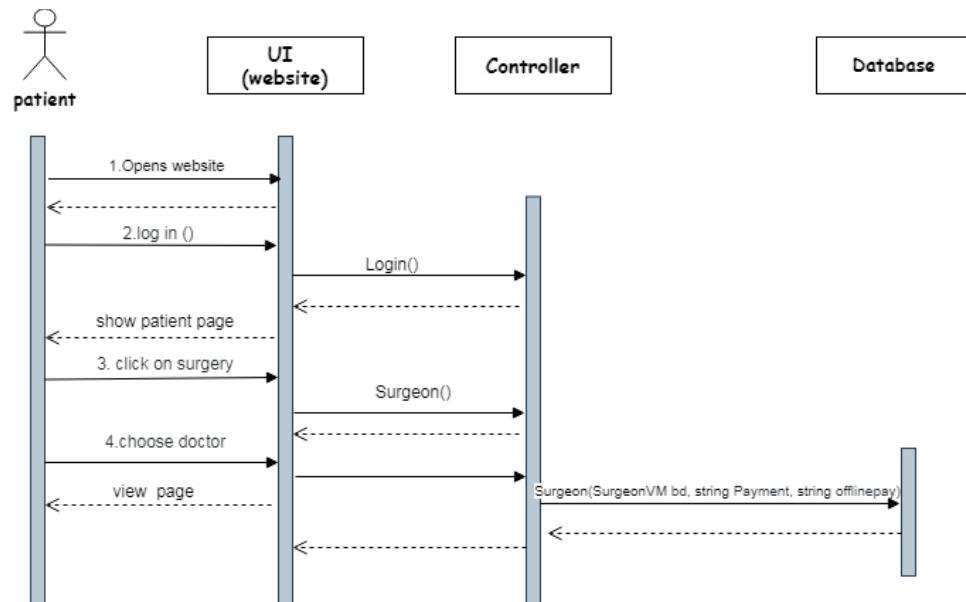


Figure 20 Use Case 4: Get services / surgery

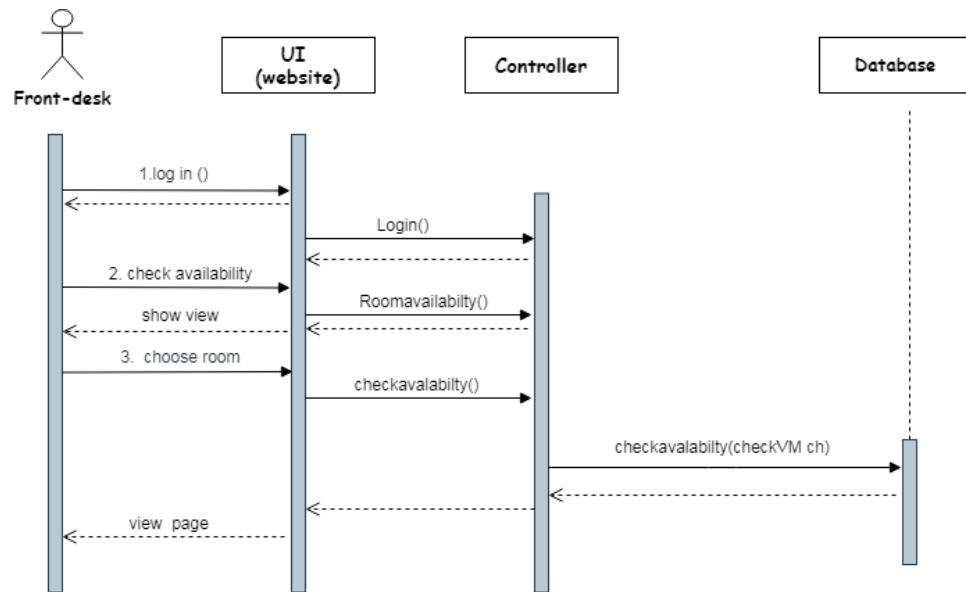


Figure 21 Use Case 5: Check availability of rooms/beds

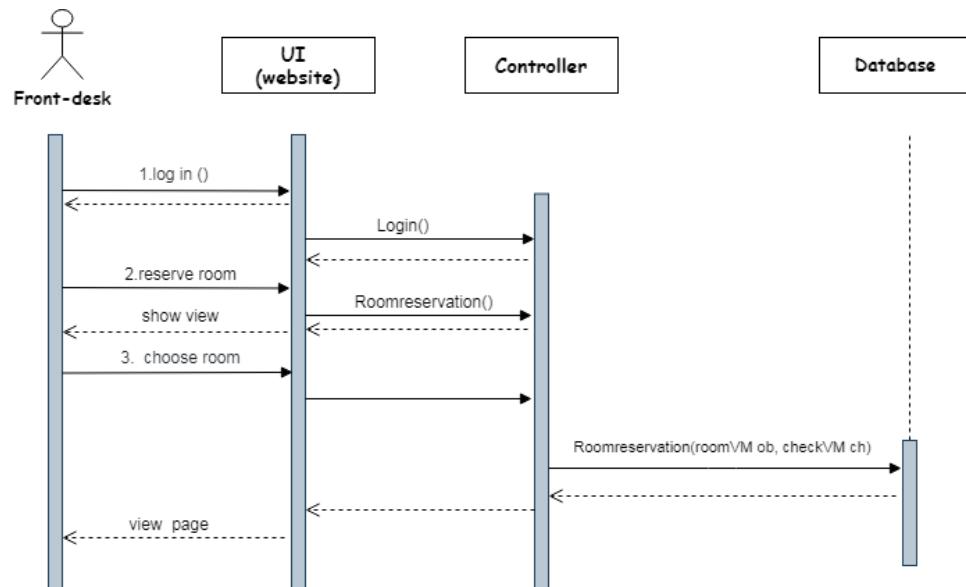


Figure 22 Use Case 6: Allocate room/bed for patient

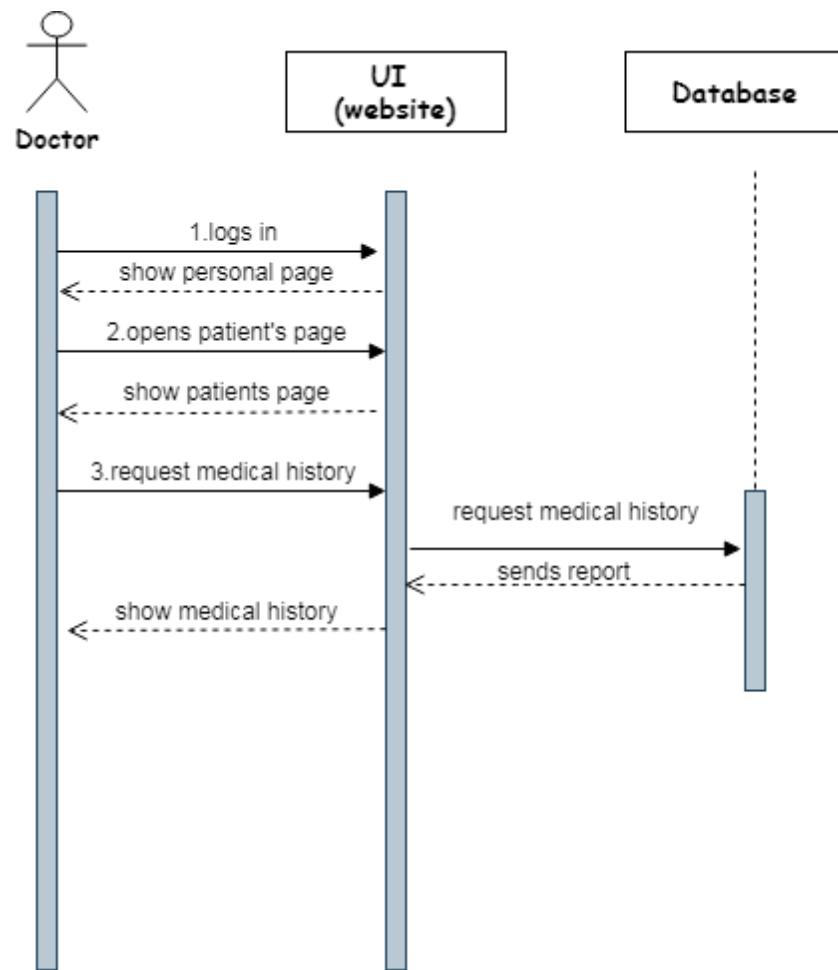


Figure 23 Use Case 7: View the medical history

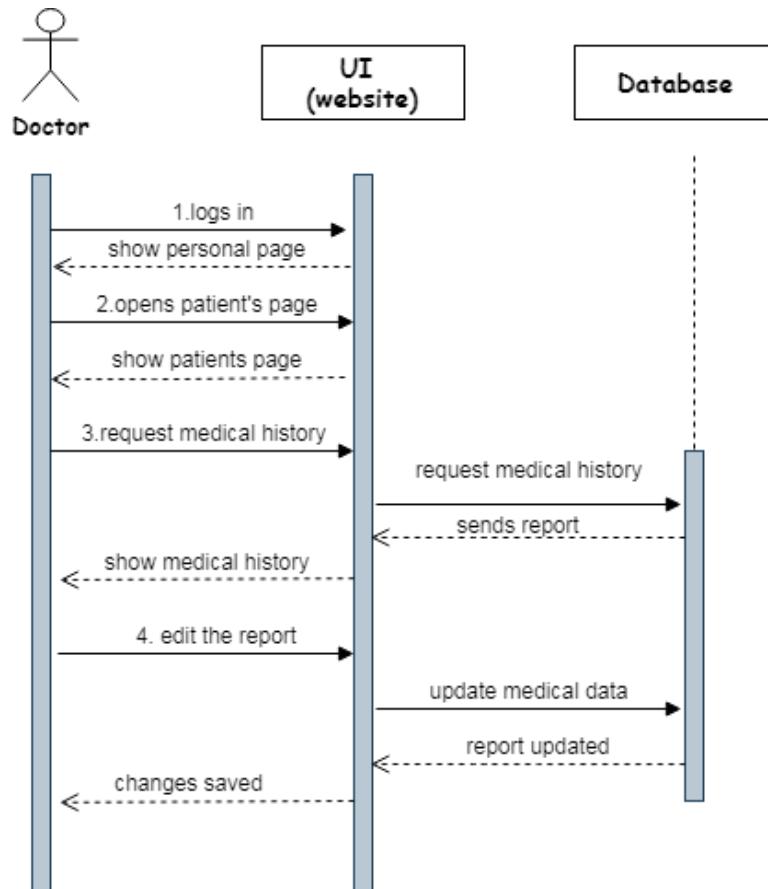


Figure 24 Use Case 8: Edit the medical history

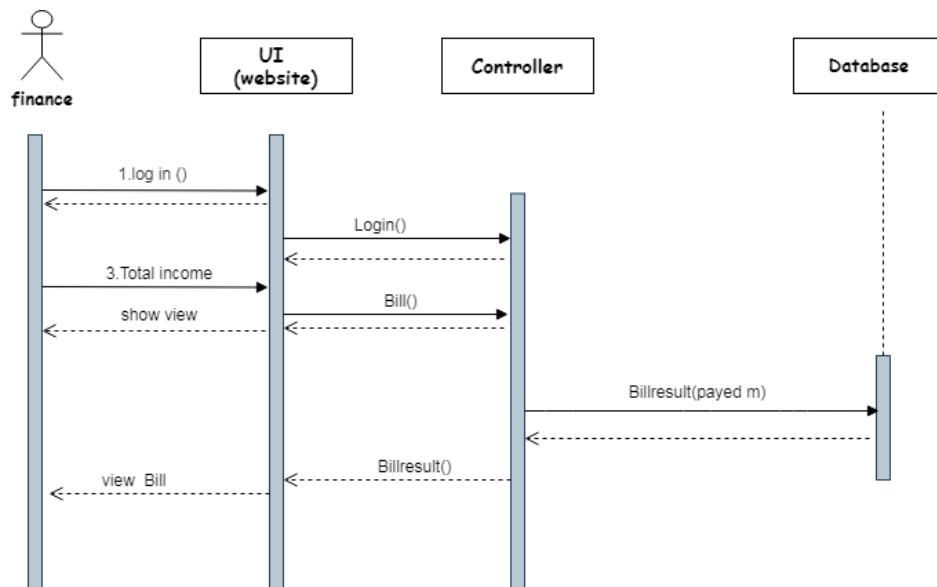


Figure 25 Use Case 9: View monthly income

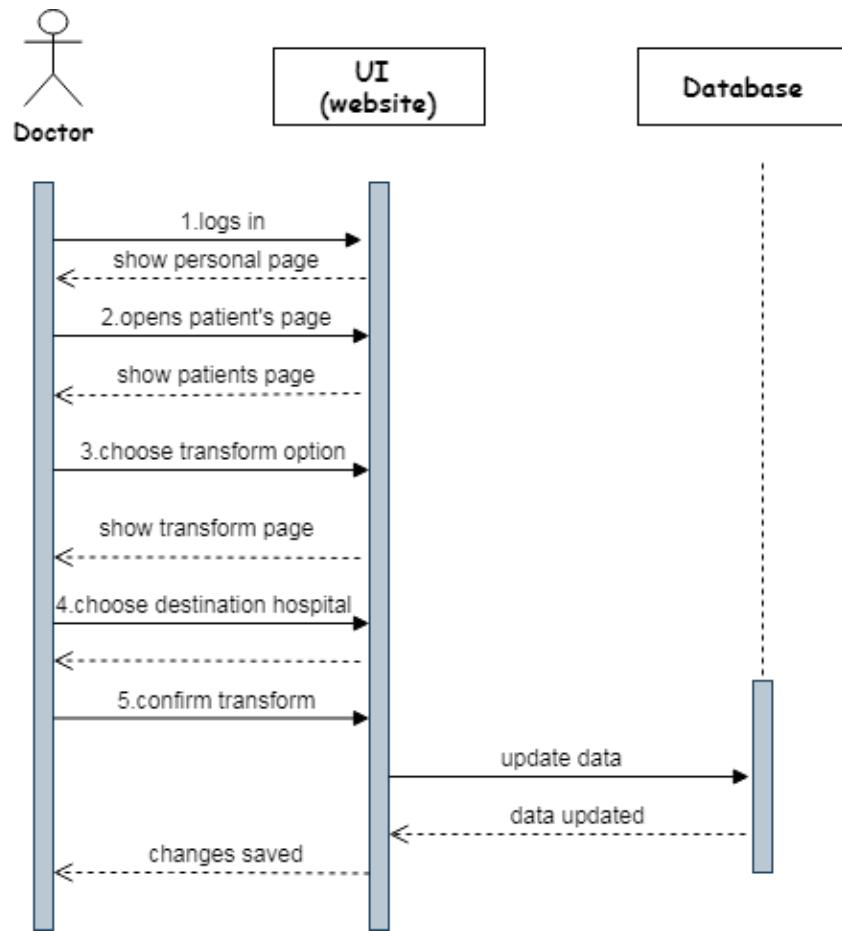


Figure 26 Use Case 10: Transform patient to another hospital/doctor

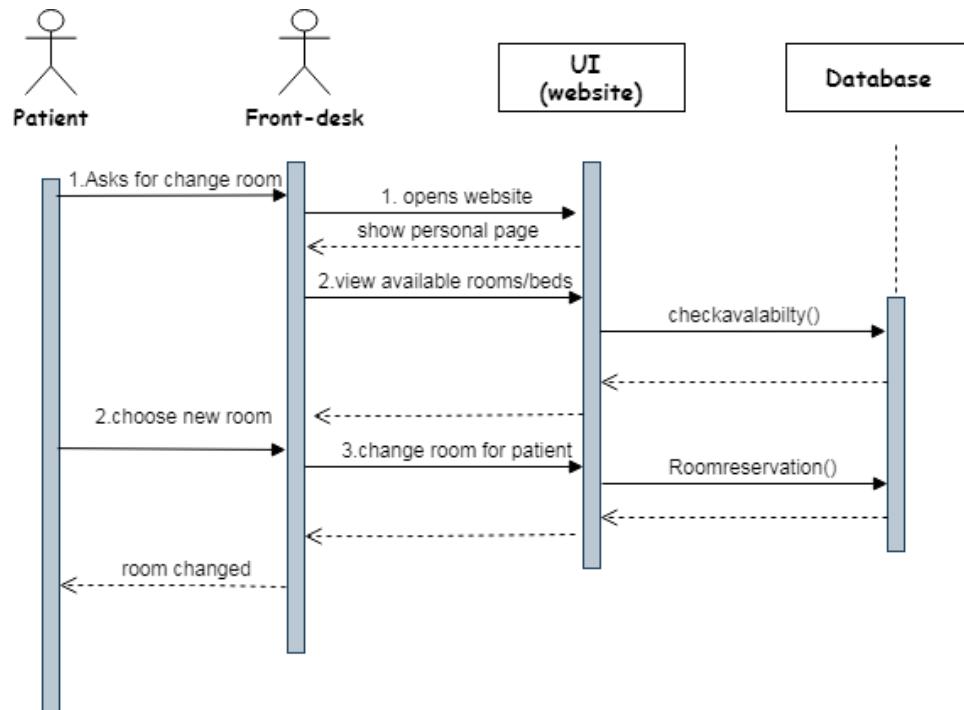


Figure 27 Use Case 11: Transform patient to another room

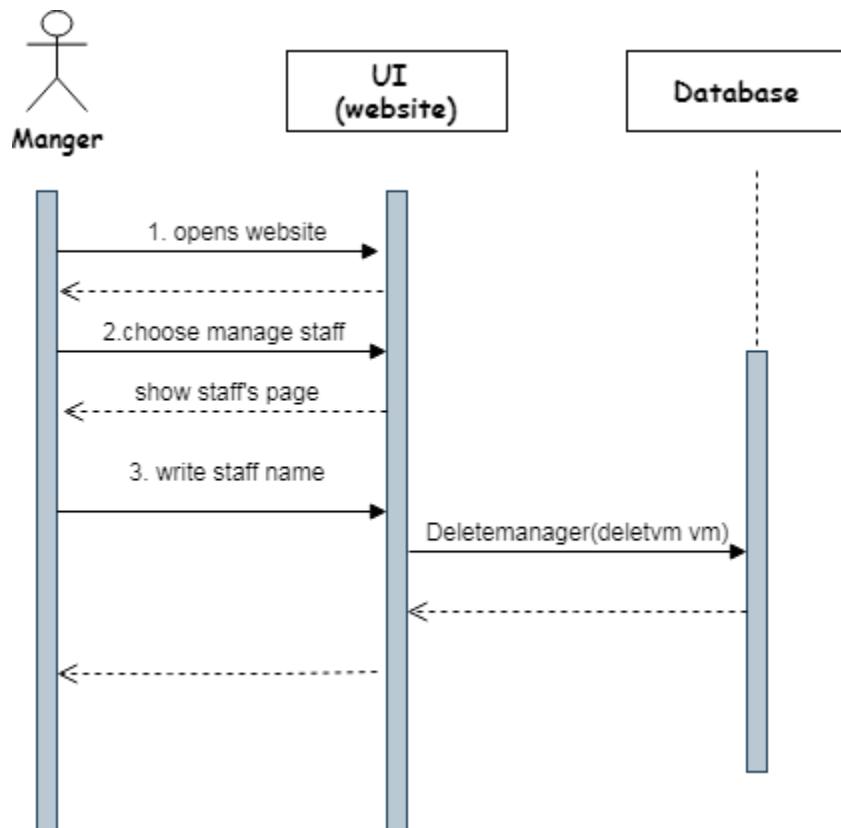


Figure 28 Use Case 12: Manage the staff

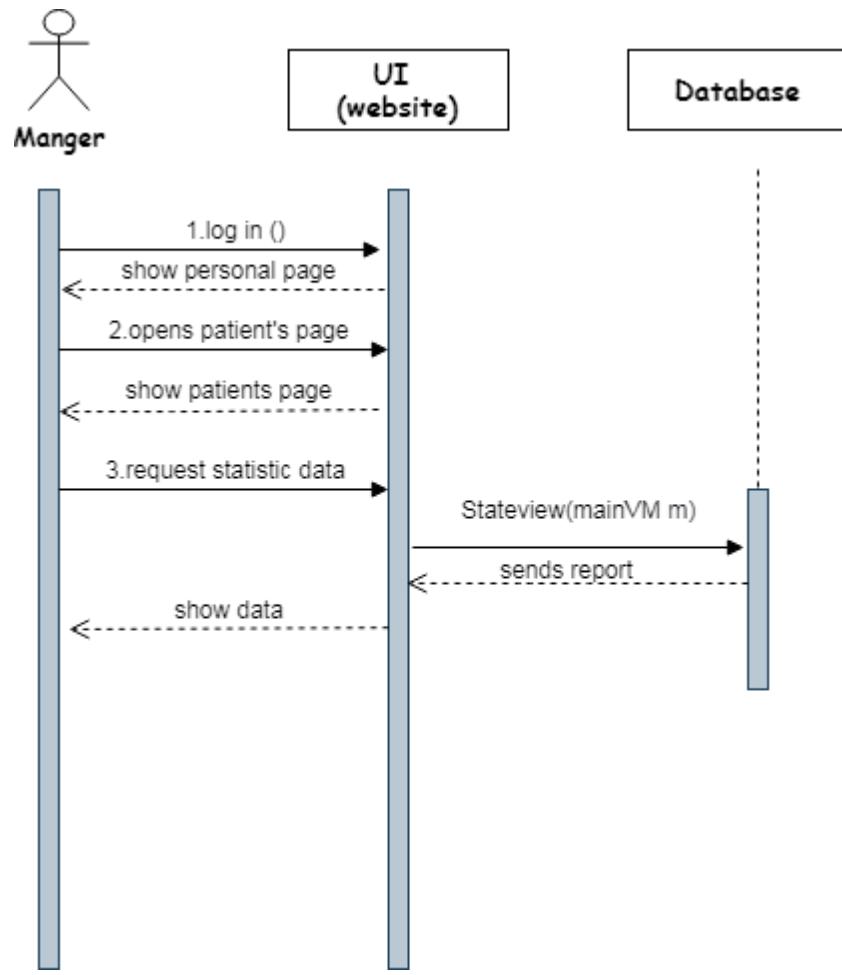


Figure 29 Use Case 13: Collect statistics about the patient

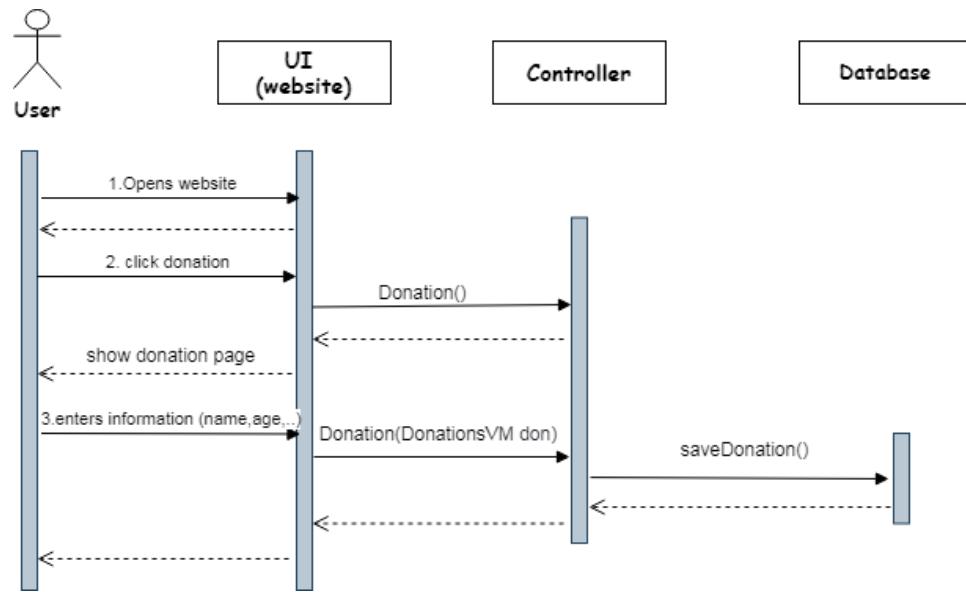


Figure 30 Use Case 14: Donation request

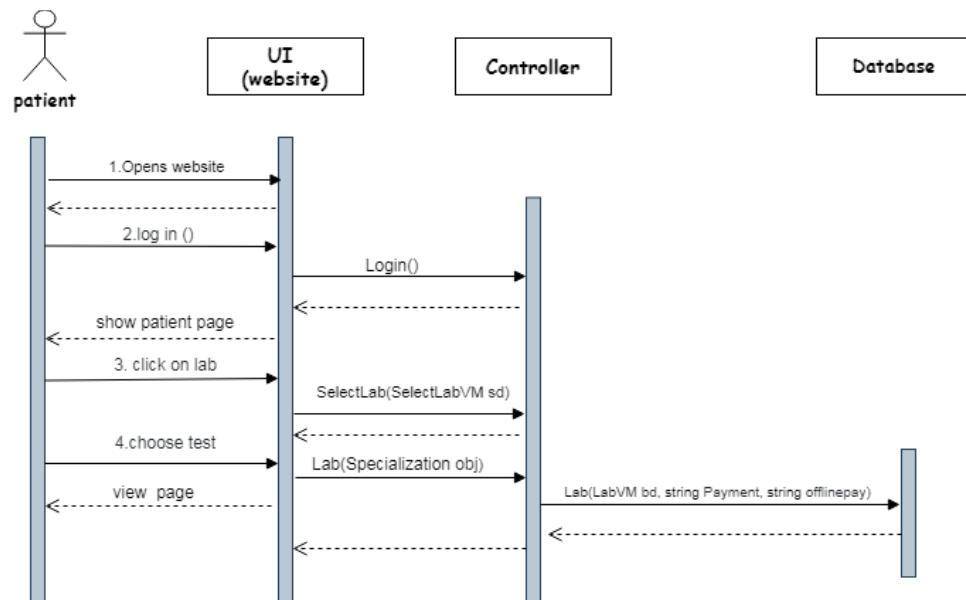


Figure 31 Use Case 15: Add lab appointment

11.Detailed class diagram:

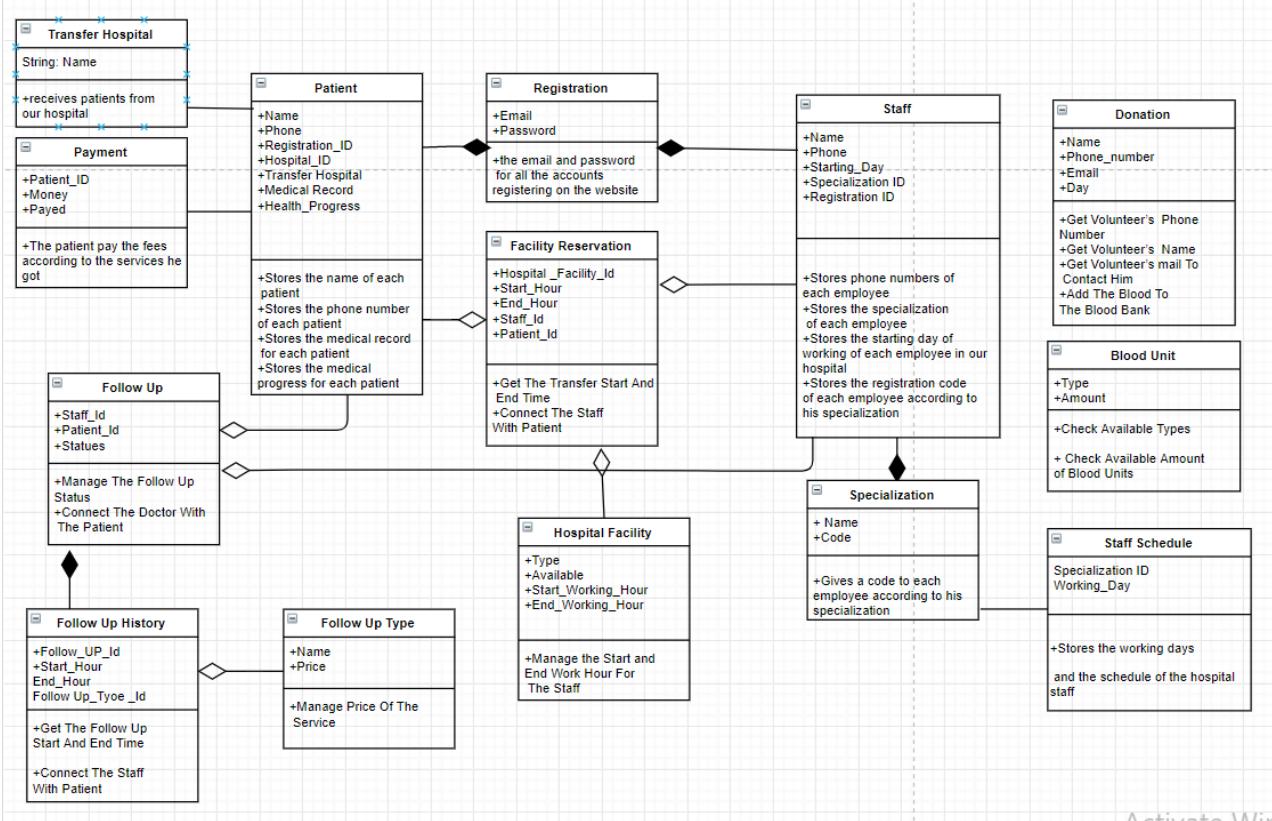


Figure 32 Detailed class diagram

12. Client-object relation diagram:

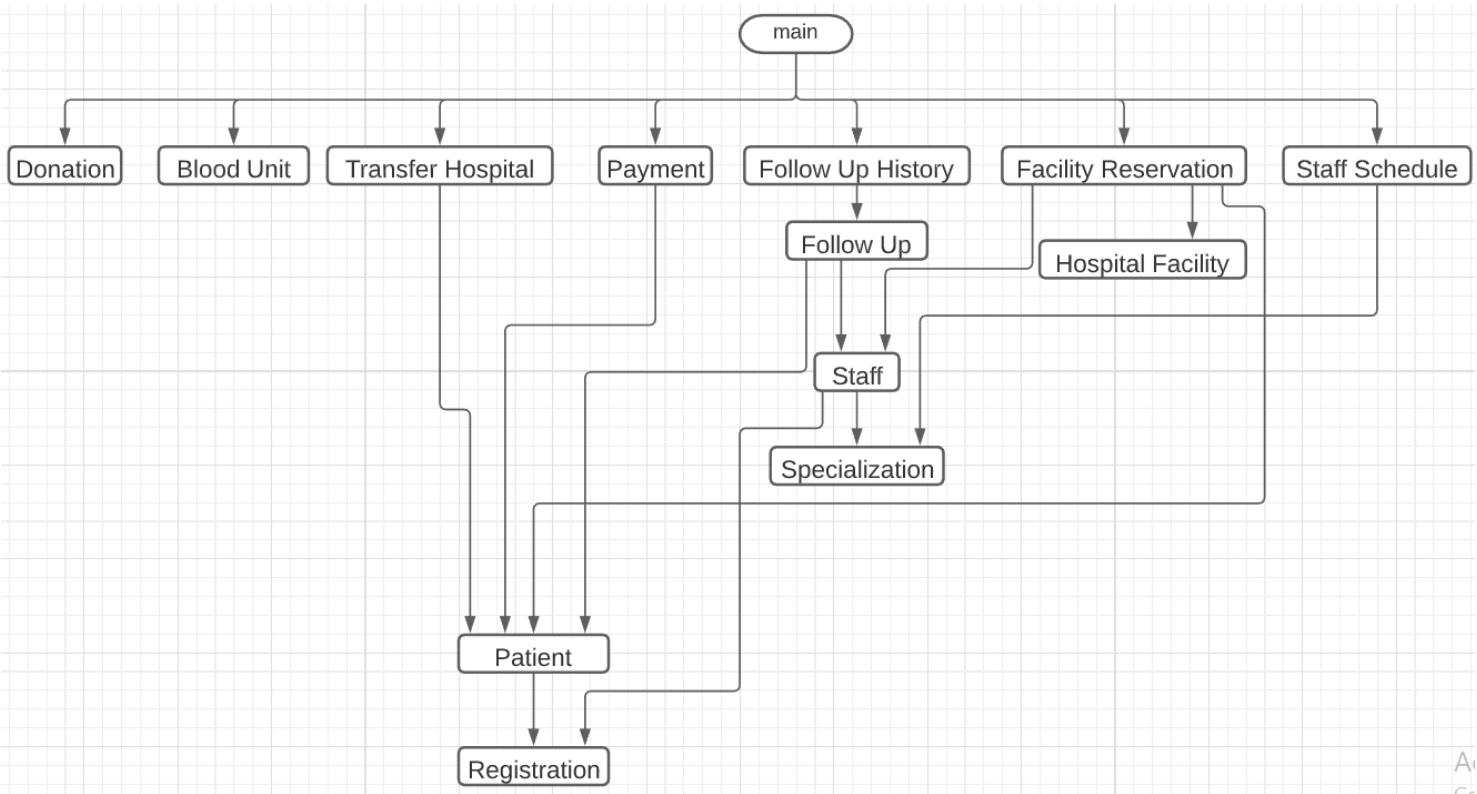


Figure 33 Client-Object relation diagram

13. Architectural model:

13.1. Model (MVC):

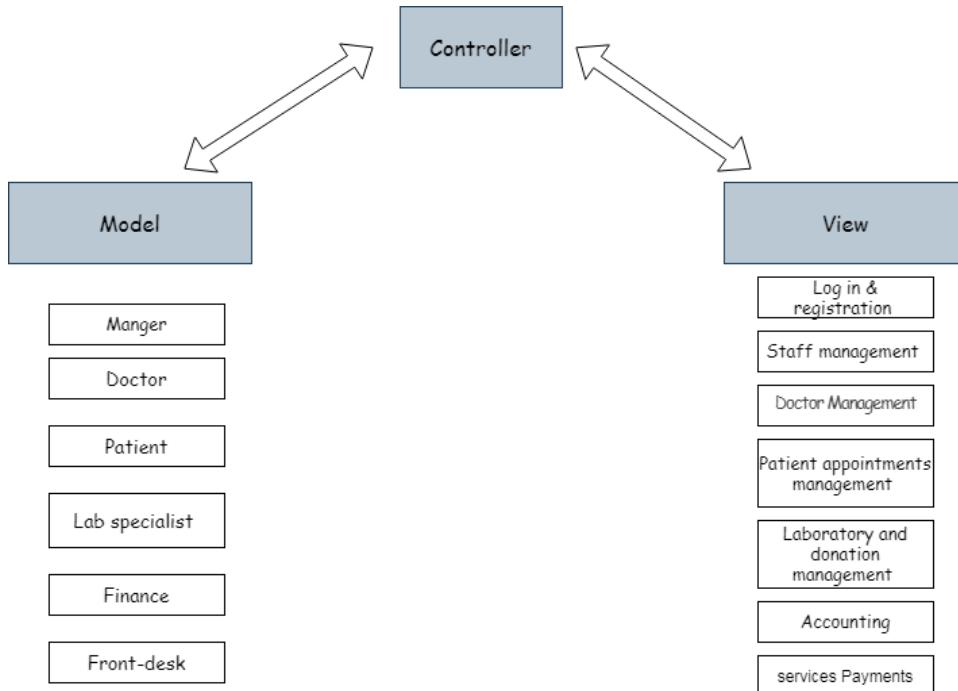


Figure 34 Architectural model (MVC)

13.2. Justifications:

In MVC pattern, application and its development are divided into three interconnected parts.

The advantage of this is it helps in focusing on a specific part of the application name; the ways information is presented to and accepted from, the user. It helps in allowing for efficient code reuse and the parallel development of the application.

13.2.1. Advantages of MVC architecture:

1. Development becomes faster than other models.
2. Easy for multiple developers to collaborate and work together.
3. Easier to update the application without need to change the model.
4. Easier to Debug as we have multiple levels properly written in the application.
5. Ability to Provide Multiple Views:

13.2.2. Disadvantages of MVC architecture:

1. Hard to understand the MVC architecture.
2. Must have strict rules on methods.

14. Component diagram:

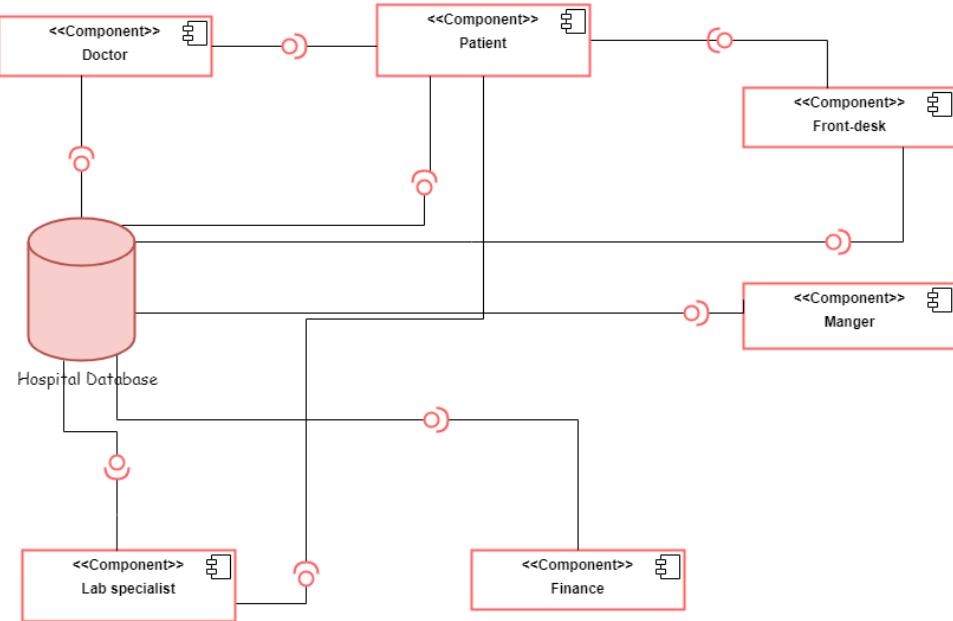


Figure 35 Component diagram

15. User Interface design:

1. Home page:

The home page is attractive for the user and there is no difficulty to find any information about the hospital or move to other pages: signup, log in, about, etc..

User interaction style "Direct manipulation"

1.1. Create new account page:

- Form fill-in style.
- User should enter his "activation code" then press "Create" to go to registration form.

The screenshot shows the homepage of Ain Shams Hospital. At the top, there is a navigation bar with the hospital's logo (a red cross inside a blue circle with 'A.S.H.' below it), the text 'AIN SHAMS HOSPITAL', and a phone number '19056'. Below the navigation bar, there are links for 'Home', 'About', and 'Contact'. The main title 'Welcome to Ain Shams Hospital' is displayed prominently, followed by a red link 'Covid Precautions!'. To the left, there is a large logo for 'AIN SHAMS HOSPITAL' featuring a red cross inside a blue swoosh with 'A.S.H.' written across it. To the right, there is a statement: "'We Care About Your Health.'" and 'We provide you with expert doctors & best services.' Below the text are two images: one showing the exterior of the hospital building and another showing the interior of a hospital hallway with medical equipment. At the bottom, there is a section titled 'YOU CAN DONATE FROM HERE' with a red 'DONATION' button. Below it, a message says 'We are happy to serve you.' with a 'Login' button and a green 'Create New Account' button. A copyright notice at the very bottom states '© 2020 - Ain Shams Hospital - Since 1900'.

Figure 36 User Interface: Home page

The screenshot shows a web page for 'AIN SHAMS HOSPITAL'. At the top right is the number '19056'. The header includes the hospital's logo and name. Below the header, there are links for 'Home', 'About', and 'Contact'. A main title 'Please enter your Activation Code' is displayed, followed by a hint: 'Hintif you are a patient enter "0000"'. There is a text input field labeled 'Activation' containing '1234', a 'Create' button, and a 'Login' link. At the bottom right, there are links for 'Activate Windows' and 'Go to Settings to activate Windows'. The footer contains the copyright notice '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 37 User Interface: Activation code page

The screenshot shows a web page for 'AIN SHAMS HOSPITAL'. At the top right is the number '19056'. The header includes the hospital's logo and name. Below the header, there are links for 'Home', 'About', and 'Contact'. The main title is 'RegistrationStaff'. The form fields include: 'Name' (input: Ahmed Ali Mohamed), 'Phone' (input: 0111223344), 'Starting_Day' (input: 03/31/2006), 'Email' (input: ahmed@yahoo.com), 'Password' (input: ****), and 'ConfirmPassword' (input: ****). There is a 'Create' button and a 'Back to List' link. At the bottom right, there are links for 'Activate Windows' and 'Go to Settings to activate Windows'. The footer contains the copyright notice '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 38 User Interface: Registration for Staff page


19056

[Home](#) [About](#) [Contact](#)

RegistrationPatient

Name

Phone

Email

Password

ConfirmPassword

[Create](#)

[Back](#)

© 2020 - Ain Shams Hospital - Since 1980

Figure 39 User Interface: Registration for patient page

1.2. log in page:

- Form fill-in style.
- Fill the right email & password then click "Login".


19056

[Home](#) [About](#) [Contact](#)

Login

Registration

Email

Password

[Login](#)

[Back to List](#)

Activate Windows
 Go to Settings to activate Windows

© 2020 - Ain Shams Hospital - Since 1980

Figure 40 User Interface: login page

1.3. Donation:

- User fills all his information then press "send".
- User can go back to main page by click on "back".
- Interaction style "Form fill".

The screenshot shows a web browser window with a red header bar. The title bar says 'Donation' and the address bar shows 'localhost:44359/IHome/Donation'. The main content area is titled 'Donation Form'. It contains four input fields: 'Enter Your Full Name' (with an empty input box), 'Enter Your Email' (with an empty input box), 'PhoneNumber' (with an empty input box), and 'Choose Day' (with a date input field showing 'mm/dd/yyyy' and a calendar icon). Below these fields is a blue 'Send' button and a black 'Back' button.

Figure 41 User Interface: Donation page

1.4. Contact us:

- This is a form to contact with hospital.
- Form fills.

The screenshot shows a 'Contact' window with a red title bar. The address bar indicates the URL is 'localhost:44359/Home/Contact'. The main content area has a title 'Contact Us'. It contains four input fields: 'Your name:' with a placeholder 'Your name', 'Email:' with a placeholder 'Email', 'Subject:' with a placeholder 'Subject', and a larger 'Message:' text area. Below these fields is a blue 'send message' button.

Figure 42 User Interface: Contact us page

1.5. About:

- A short overview about the hospital appears in this page.

The screenshot shows a 'About' window with a red title bar. The address bar indicates the URL is 'localhost:44359/Home/About'. The header features the Ain Shams Hospital logo and the year '19056'. The main content includes a 'Home' link, a 'About' link (which is active), and a 'Contact' link. Below this is a section titled 'About Ain Shams Hospital' with a paragraph about the hospital's experience and team. To the right is a graphic of hands assembling puzzle pieces, one of which is pink and labeled 'MISSION'. Another piece is labeled 'VISION'. The bottom left lists 'Our Vision' and 'Our Mission' with their respective descriptions. The bottom right lists 'Core Values' with three items: Quality, Safety, and Empathy.

Figure 43 User Interface: About page

2.Patient page:

2.1 Patient can clearly get hospital clinics in "menu selection style" he can choose easily and free to move between them.

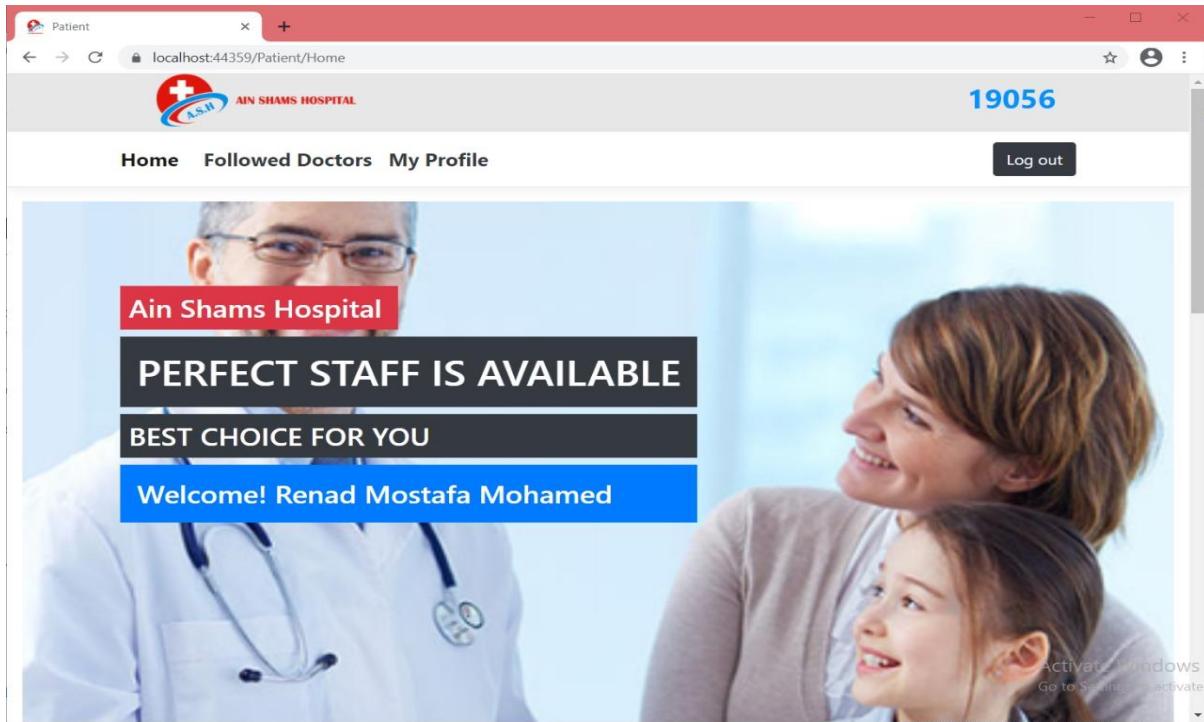


Figure 44 User Interface: patient home page

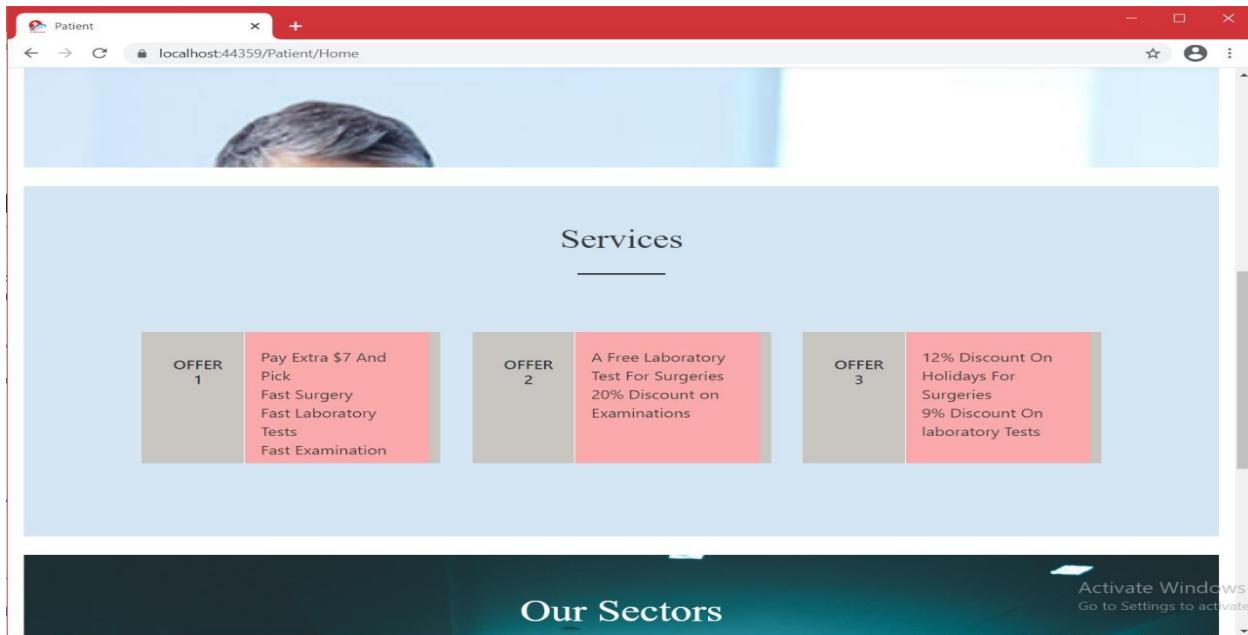


Figure 45 User Interface: patient home page services section

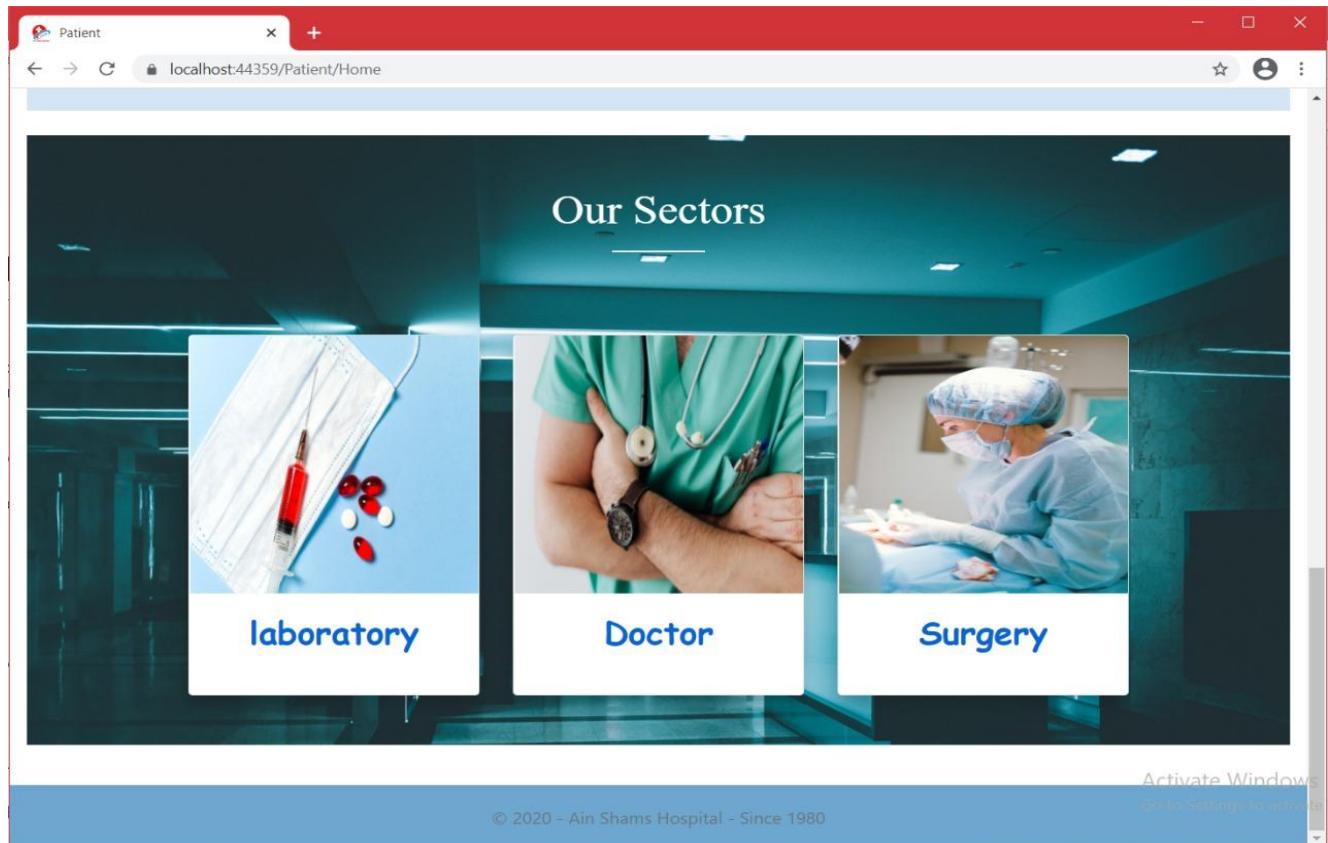


Figure 46 User Interface: patient home page sectors section

In this page the interaction style is "Direct Manipulation"

User should click on the different choices to move for the next page.

2.2 Add appointment with doctor:

In this page the interaction style is "Menu selection"

User should click on next to complete Adding.

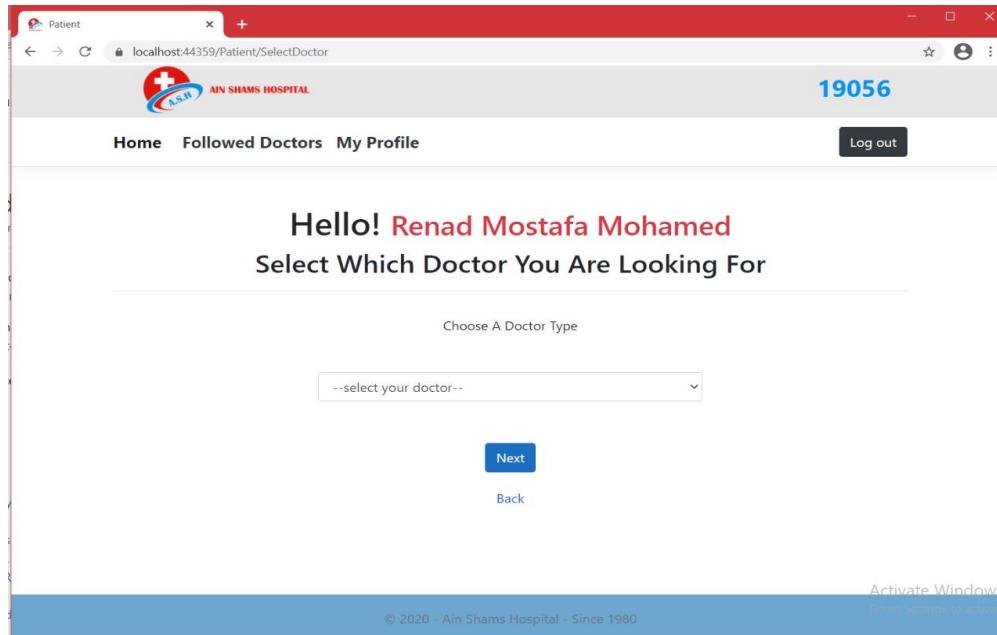


Figure 47 User Interface: add appointment with doctor page

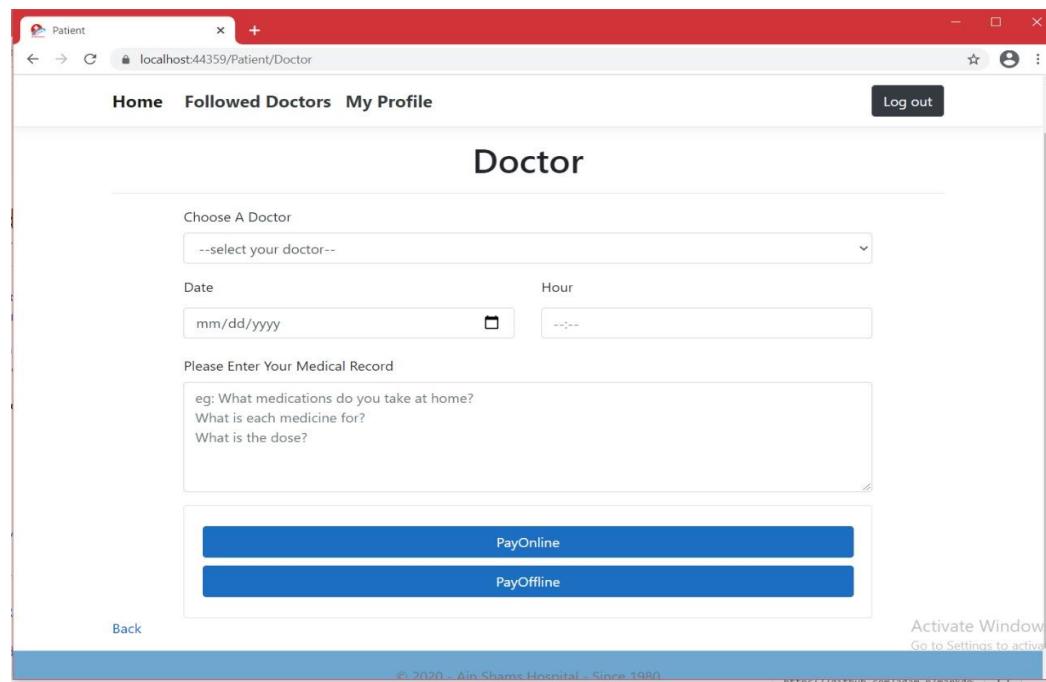


Figure 48 User Interface: filling appointment data with doctor page

2.3 Add appointment in the lab:

The screenshot shows a web browser window titled "Patient" with the URL "localhost:44359/Patient/labSpecialist". The page has a red header bar with the text "Home Followed Doctors My Profile" and a "Log out" button. Below the header, the word "laboratory" is displayed in large, bold, black letters. A section titled "Choose Your Test" contains a dropdown menu with the placeholder "-Select Your Test--". Below this are fields for "Date" (a date input field) and "Hour" (a time input field). A section titled "Please Enter Your Medical Record" contains a text area with placeholder text: "eg: What medications do you take at home? What is each medicine for? What is the dose?". At the bottom of the page are two blue buttons labeled "PayOnline" and "PayOffline". On the left side, there is a "Back" link. On the right side, there is a watermark-like text: "Activate Windows Go to Settings to activate".

Figure 49 User Interface: add appointment in the lab page

2.4 Add surgery with a doctor:

The screenshot shows a web browser window titled 'Patient' with the URL 'localhost:44359/Patient/Surgeon'. The page has a header with 'Home', 'Followed Doctors', 'My Profile', and a 'Log out' button. The main content area is titled 'Surgeon' and contains a form for scheduling a surgery. It includes fields for 'Choose A Doctor' (a dropdown menu with the placeholder '--select your doctor--'), 'Date' (a date input field with placeholder 'mm/dd/yyyy') and 'Hour' (a time input field with placeholder '---:--'). Below these is a text area labeled 'Please Enter Your Medical Record' with placeholder text: 'eg: What medications do you take at home? What is each medicine for? What is the dose?'. At the bottom right of the form is a blue button labeled 'PayOnline' and below it another blue button labeled 'PayOffline'. At the very bottom of the page is a blue footer bar with the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 50 User Interface: add surgery with a doctor page

2.5 Pay online :

The screenshot shows a web browser window titled 'Patient' with the URL 'localhost:44359/Patient/Payment'. The page has a header with 'Home', 'Followed Doctors', 'My Profile', and a 'Log out' button. The main content area is titled 'Pay Invoice' and contains a form for payment. It includes fields for 'Payment amount' (\$120), 'Card Number' (input field), 'Name on card' (input field), 'Expiry Date' (date input field with placeholder 'mm/dd/yyyy') and 'Security Code' (input field). Below these fields is a green button with '\$120'. At the bottom left of the form is a 'Back' link. At the very bottom of the page is a blue footer bar with the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 51 User Interface: pay online page

In all these previous pages the interaction style is "Form fill in"

2.6 Follow doctors:

Interaction style "Menu selection"

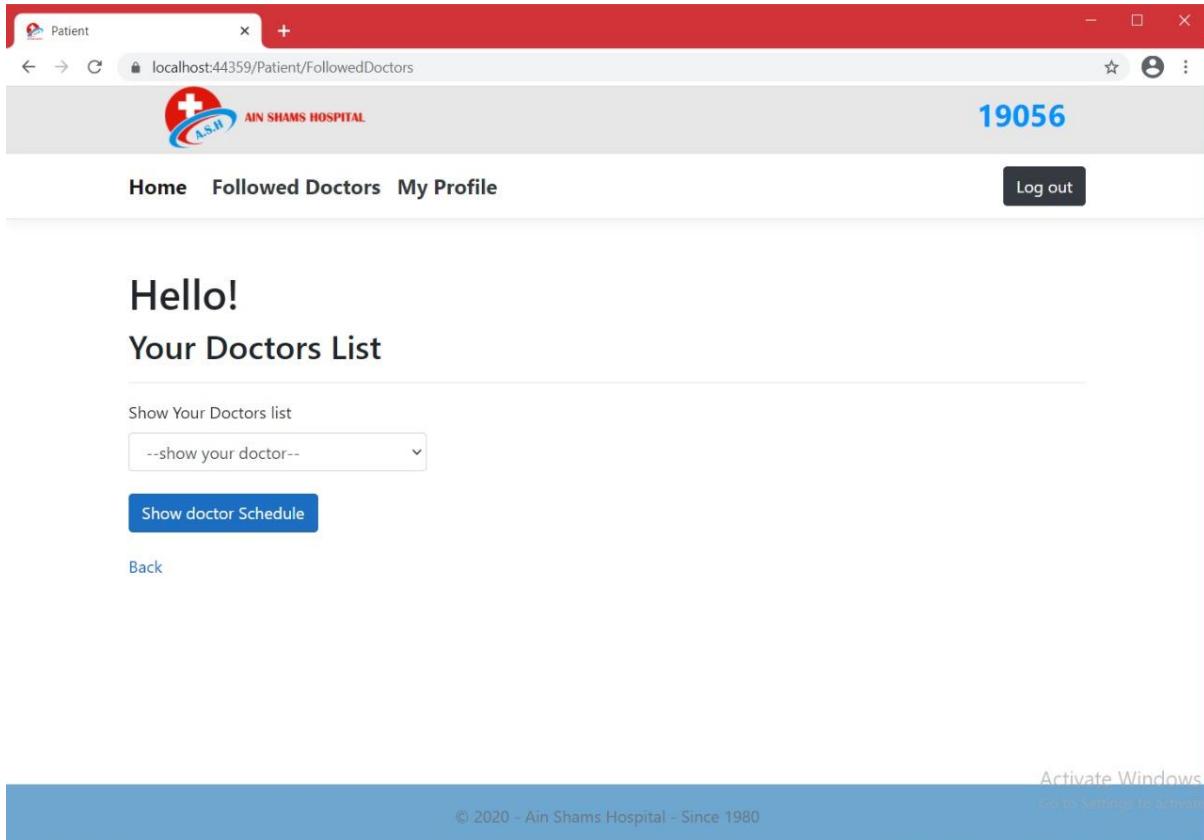


Figure 52 User Interface: followed doctors page

2.7 View your profile:

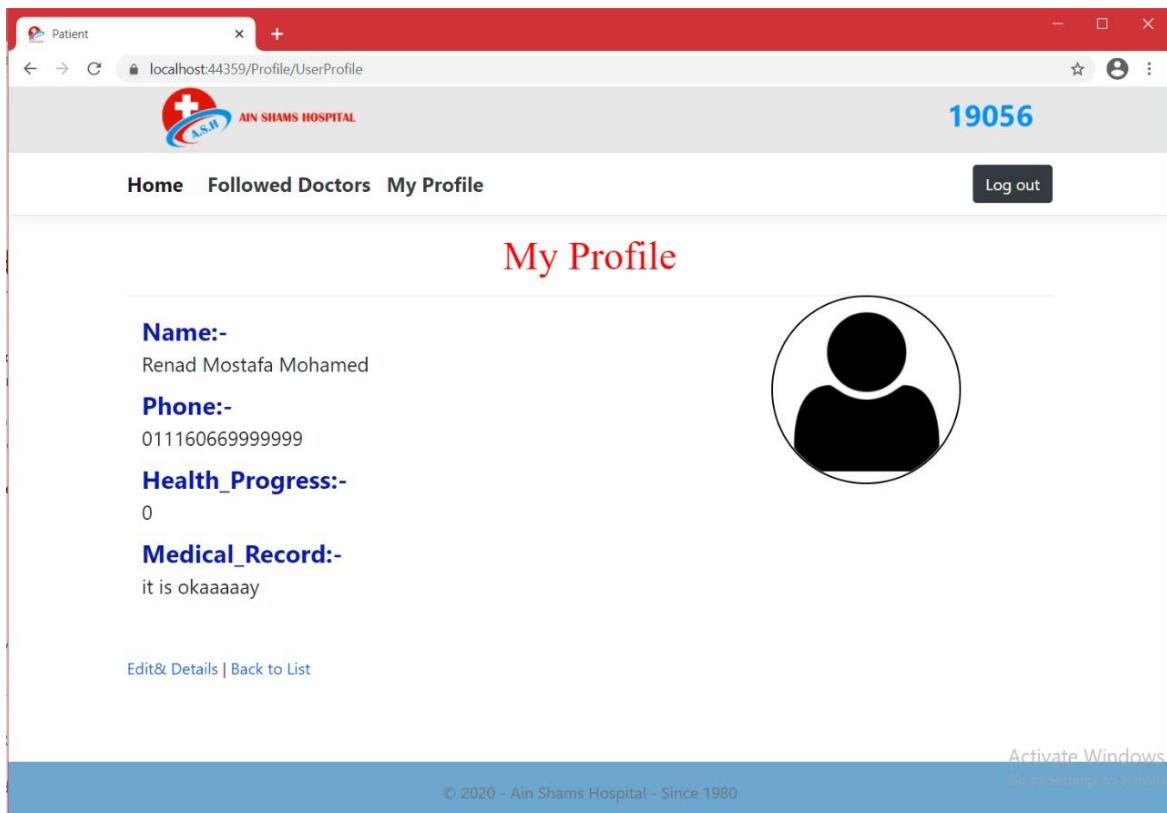


Figure 53 User Interface: patient profile page

Interaction style "Direct Manipulation"

If you want to edit your information press "Edit Details"

2.8 Edit profile:

Interaction style "Form fill in"

The screenshot shows a user interface for editing a patient profile. At the top, there is a header with the hospital logo (Ain Shams Hospital) and a phone number (19056). Below the header, there is a navigation bar with links for Home, About, Contact, and Log out. The main title is 'Edit Your Profile'. There are three input fields: 'Name' (containing 'nour'), 'Phone' (containing '01111122344'), and 'Medical_Record' (containing 'rrrrrrrrrrccccc'). Below these fields is a section titled 'Your Progress' with a progress bar consisting of a blue striped segment followed by a grey segment. Underneath the progress bar is a section titled 'Your Current Hospital' with the name 'Hod El Marsoud'. At the bottom left is a 'SaveChanges' button, and at the very bottom left is a 'Back' link.

Figure 54 User Interface: Edit profile of patient page

3. Doctor page:

3.1 View Doctor's page, schedule, patients and profile:

Doctor can easily view his schedule & profile.

Interaction style "menu selection"

Also doctor can search for patient from a search bar.

The screenshot shows the Ain Shams Hospital doctor home page. At the top right, the doctor's ID '19056' is displayed. The header includes the hospital logo and name 'AIN SHAMS HOSPITAL'. Navigation links for 'Home', 'My Schedules', and 'Profile' are at the top left, along with 'Log out' and 'Emergency' buttons. A search bar with a 'Go' button is located above a table of patient profiles. The table has two rows, each containing a patient's number, name, follow-up status, and a note about recent transfer. The first row is for 'Patient No. 15' (Name: Soha Hany, Follow Up: Examination, Status: Pending, Date: 2021-03-12T12:00). The second row is for 'Patient No. 16' (Name: Manar Hany, Follow Up: Examination, Status: Pending, Recently Transferred to You). Below the table, a copyright notice reads '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 55 User Interface: doctor home page

The screenshot shows the doctor's schedule page. At the top right, the doctor's ID '19056' is displayed. The header includes the hospital logo and name 'AIN SHAMS HOSPITAL'. Navigation links for 'Home', 'My Schedules', and 'Profile' are at the top left, along with 'Log out' and 'Emergency' buttons. The main content area is titled 'My schedule' and contains a table of scheduled appointments. The table has two columns: 'Date' and 'Follow Up'. There is one entry: '2021-03-12T12:00' under 'Date' and 'Examination' under 'Follow Up'. Below the table, a copyright notice reads '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 56 User Interface: doctor schedule page

- Doctor can change his information and save changes.
- Form fills style.

The screenshot shows a user interface for a doctor's profile. At the top, there is a logo for 'AIN SHAMS HOSPITAL' and a patient ID '19056'. Below the logo are navigation links: 'Home', 'My Schedules', and 'Profile'. On the right side of the header are 'Log out' and 'Emergency' buttons. The main content area is titled 'Profile' and displays a box with the text 'Hi DR. Ahmed'. Inside the box, there are input fields for 'Full name' (Ahmed Ali Mohamed), 'Phone number' (0111223344), 'You work in our hospital since' (2006-03-31), 'Email' (ahmed@yahoo.com), and 'Your specialization is: Internist'. Below these fields is a note: 'If you want to save any changes to your personal data click on save changes'. At the bottom of the box is a blue 'Save changes' button. At the very bottom of the page, there is a copyright notice: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 57 User Interface: doctor profile page

3.2 View & update medical record:

The screenshot shows a user interface for updating patient medical data. At the top, there is a logo for 'AIN SHAMS HOSPITAL' and a patient ID '19056'. Below the logo are navigation links: 'Home', 'My Schedules', and 'Profile'. On the right side of the header are 'Log out' and 'Emergency' buttons. The main content area is titled 'Patient personal informations'. It contains a table with rows for 'Name: Manar Hany', 'Phone: 0126716', 'Email: manar@gmail.com', 'Health Progress: 0', and 'Status: Pending'. Below this table is a section titled 'Patient Medical_Record' with a text input field. Underneath is a section titled 'Add Medical_Record:' with a text input field and an 'Edit Medical Record' button. Another section titled 'Add Health_Progress:' has a slider and a text input field. At the bottom of the page, there is a copyright notice: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 58 User Interface: updating patient's medical data page part1

Health Progress: 0 Status: Pending

Patient Medical_Record

Add Medical_Record:

Add Health Progress:

0

Add Status: Status:

If you need to transfer patient.

Transfer

© 2020 - Ain Shams Hospital - Since 1980

Figure 59 User Interface: updating patient's medical data page part2

3.3 Transfer patient to another hospital/doctor:

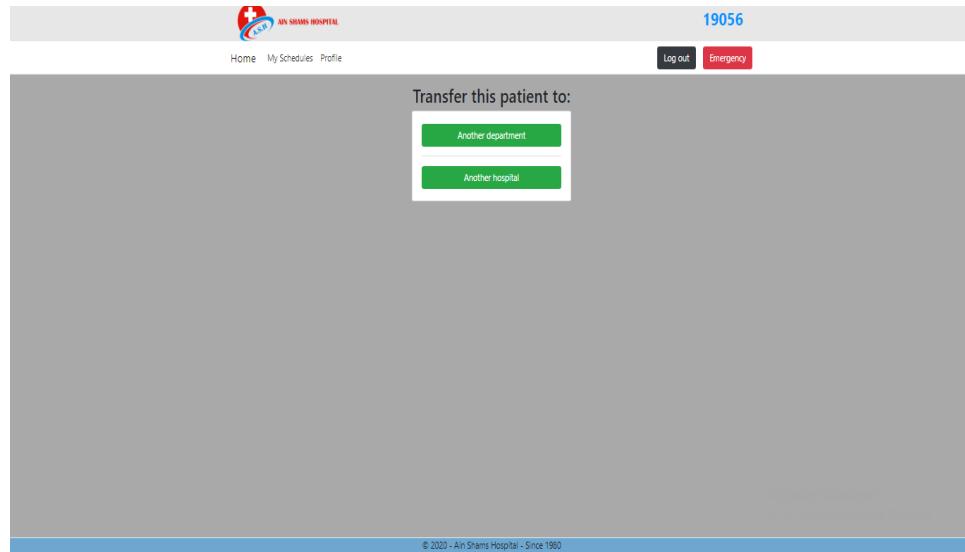


Figure 60 User Interface: transferring patient page

User interaction style "Form fill"

Doctor should fill transfer form then submit the request.

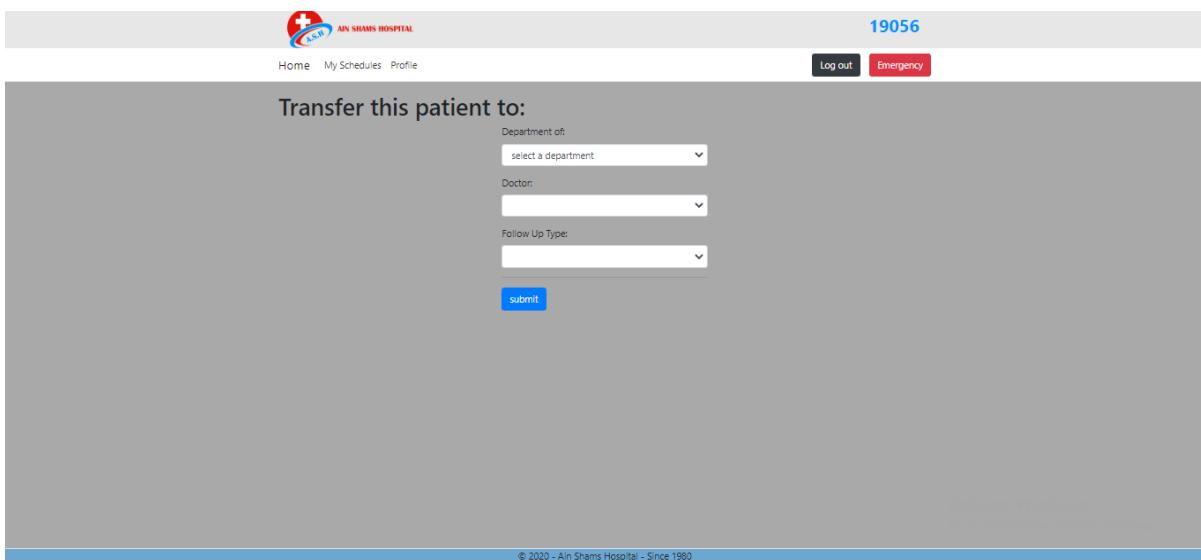


Figure 61 User Interface: transferring patient to another department page

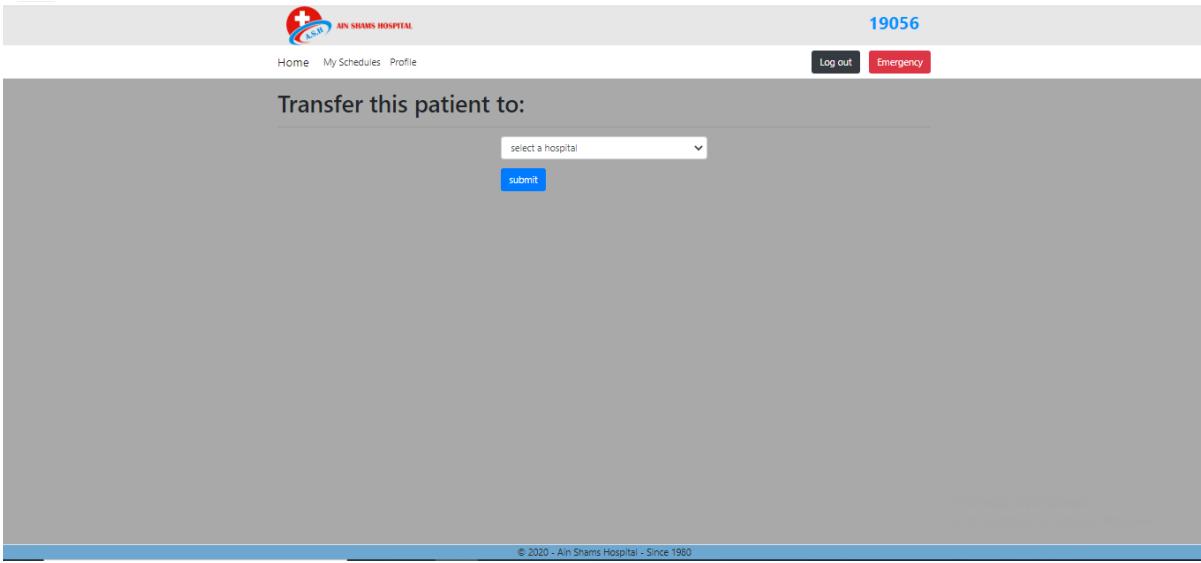


Figure 62 User Interface: transferring patient to another hospital page

4. Manager page:

4.1 Manager home page:

- User interaction style "menu selection".
- Choose the service he wants to check by click on it.

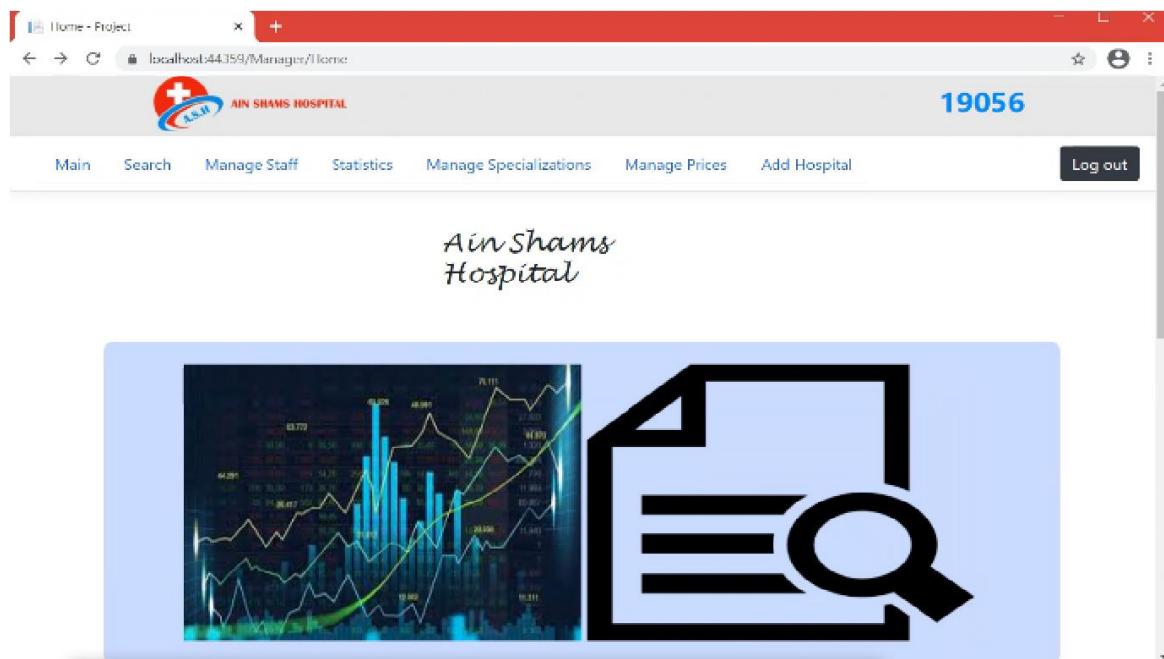


Figure 63 User Interface: manager home page part1



Figure 64 User Interface: manager home page part2

4.2 Manager search for certain staff:

- User interaction style "form fill"
- Manager should write the name in the textbox then press "search" to view staff personal information.

The screenshot shows a web browser window titled 'Manager Home - Project'. The address bar displays 'localhost:44359/Manager/ManagerHome'. The top navigation bar includes links for Main, Search, Manage Staff, Statistics, Manage Specializations, Manage Prices, Add Hospital, and Log out. Below the navigation bar is a search form with a 'Name' input field and a 'Search' button. A modal window titled 'Staff Personal Information' is displayed, containing fields for Name, Phone, Starting Day, Specialization, and Email.

Figure 65 User Interface: search for staff page

4.3. Manage staff:

- Manager writes a certain name and deletes it from the hospital staff.
- User interaction style "form fill"

The screenshot shows a web browser window titled 'Delete manager - Project'. The address bar displays 'localhost:44359/Manager/DeleteManager'. The top navigation bar includes links for Main, Search, Manage Staff, Statistics, Manage Specializations, Manage Prices, Add Hospital, and Log out. The date '19/05/2023' is prominently displayed at the top right. Below the navigation bar is a form with a 'StaffName' input field and a 'Delete Staff Member' button. The footer of the page includes the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 66 User Interface: managing staff page

4.4. Statistics:

- In this part Manger can select a certain period of time and collect statistics for it.
- User interaction style "Direct manipulation" & "form fill"

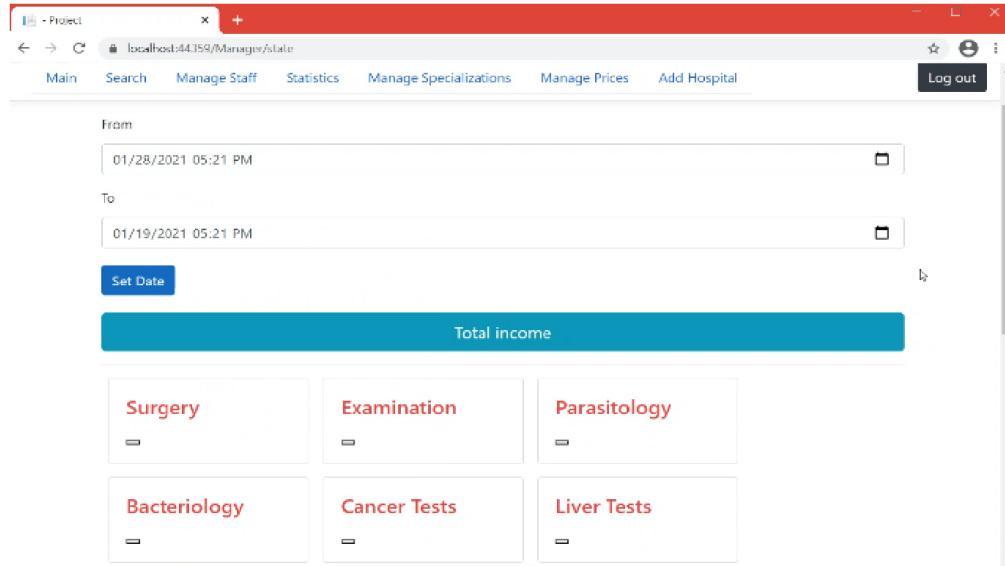


Figure 67 User Interface: statistics page

4.5. Manage specializations:

- Manger can add or delete any specializations.
- User interaction style "Direct manipulation"

The screenshot shows a web browser window titled 'localhost:44359/Manager/ViewSpecializations'. At the top, there are two blue buttons: 'Add Specialization' and 'Delete Specialization'. Below them is a table with two columns: 'Specialization' and 'Code'. The data in the table is as follows:

Specialization	Code
Patient	0000
Internist	1234
cardiologist	1345
Glands specialist	1456
Doctor of Hematology	1567
Doctor of Nephrology	1678
Oncologist	1789
Doctor of Rheumatology	1888
Surgeon	1999

Figure 68 User Interface: managing specializations page

4.6. Manage prices:

- In this part Manager can add new service or delete it, also can edit its price.
- User interaction style "Direct manipulation"

The screenshot shows a web browser window titled 'localhost:44359/Manager/ViewPrices'. At the top, there is a logo for 'AIN SHAMS HOSPITAL' and a user ID '19056'. The navigation menu includes 'Main', 'Search', 'Manage Staff', 'Statistics', 'Manage Specializations', 'Manage Prices', 'Add Hospital', and 'Log out'. Below the menu, there are three blue buttons: 'Add Service', 'Edit Prices' (which is highlighted in grey), and 'Delete Service'. A table below shows a list of services with their names and prices:

Name	Price
Surgery	2600
Examination	500
Parasitology	600
Bacteriology	160
Cancer Tests	120

Figure 69 User Interface: manage prices page

4.7. Add hospital:

- Manager can add hospital.
- User interaction style "Direct manipulation"

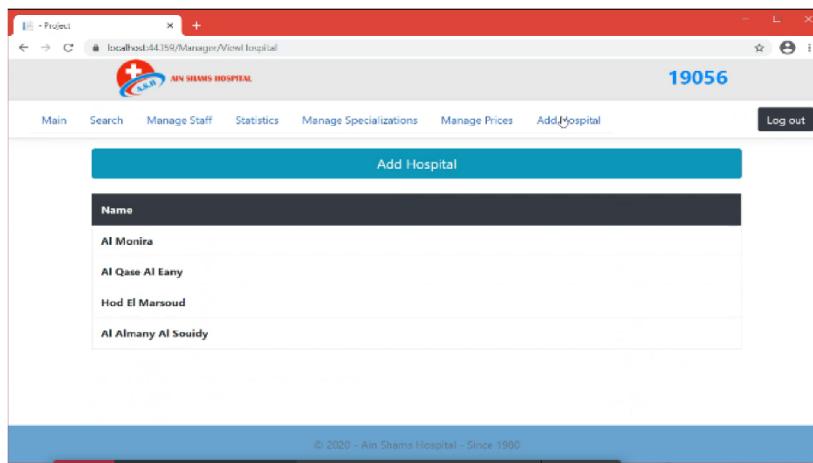


Figure 70 User Interface: adding hospital page

5. Front-desk page:

- Front-desk home page.
- He can apply different services.
- Interaction style "Menu selection"

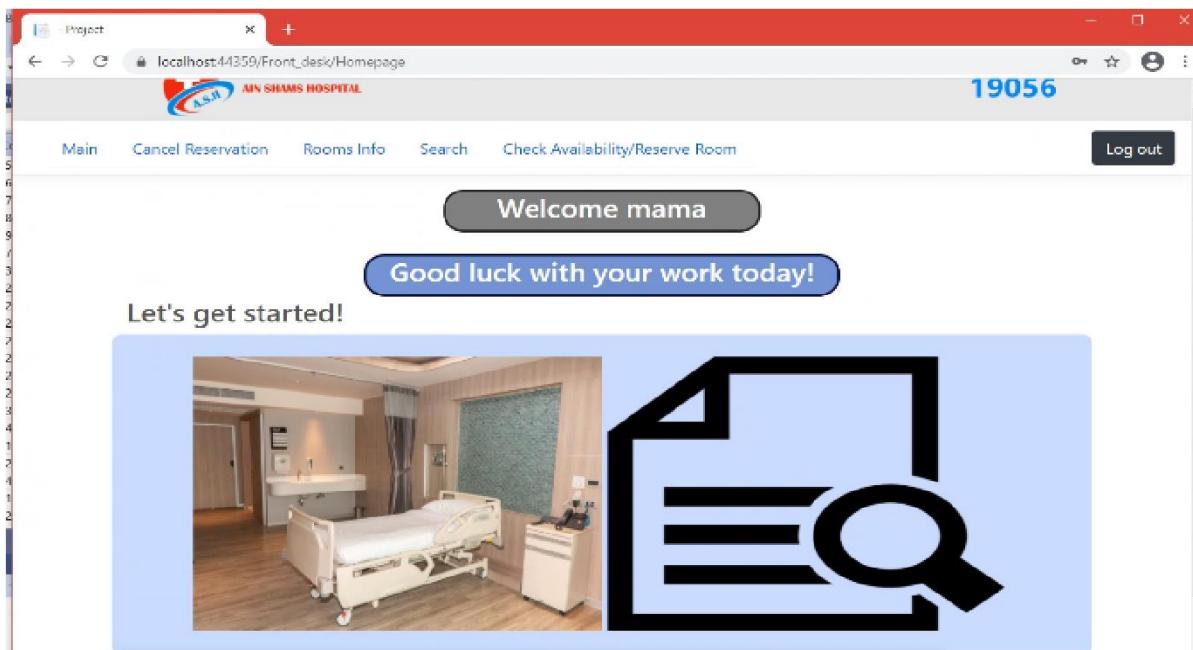


Figure 71 User Interface: front desk home page

5.1 Check Availability/Reserve Room:

Interaction Style "form fill"

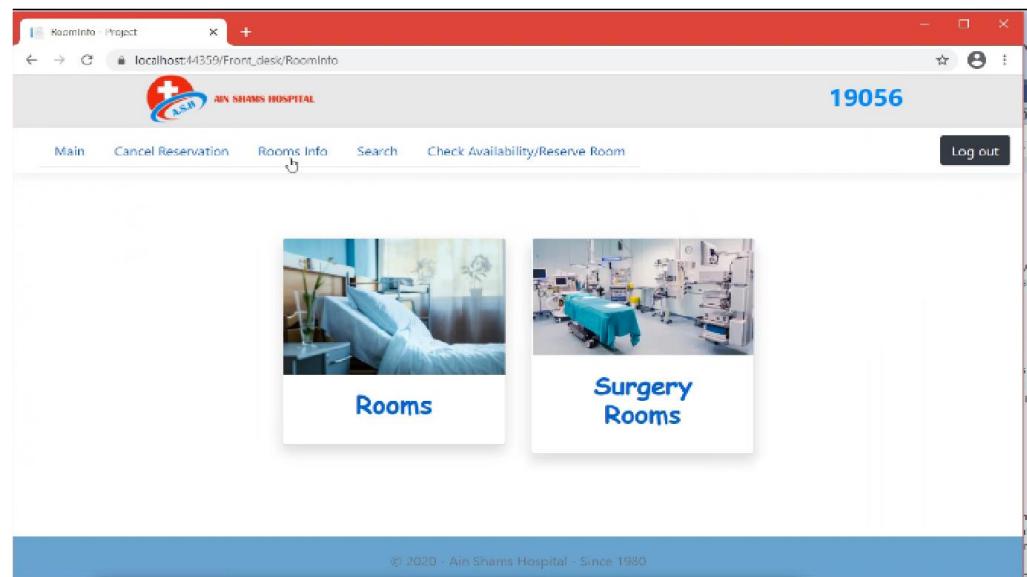


Figure 72 User Interface: room info page

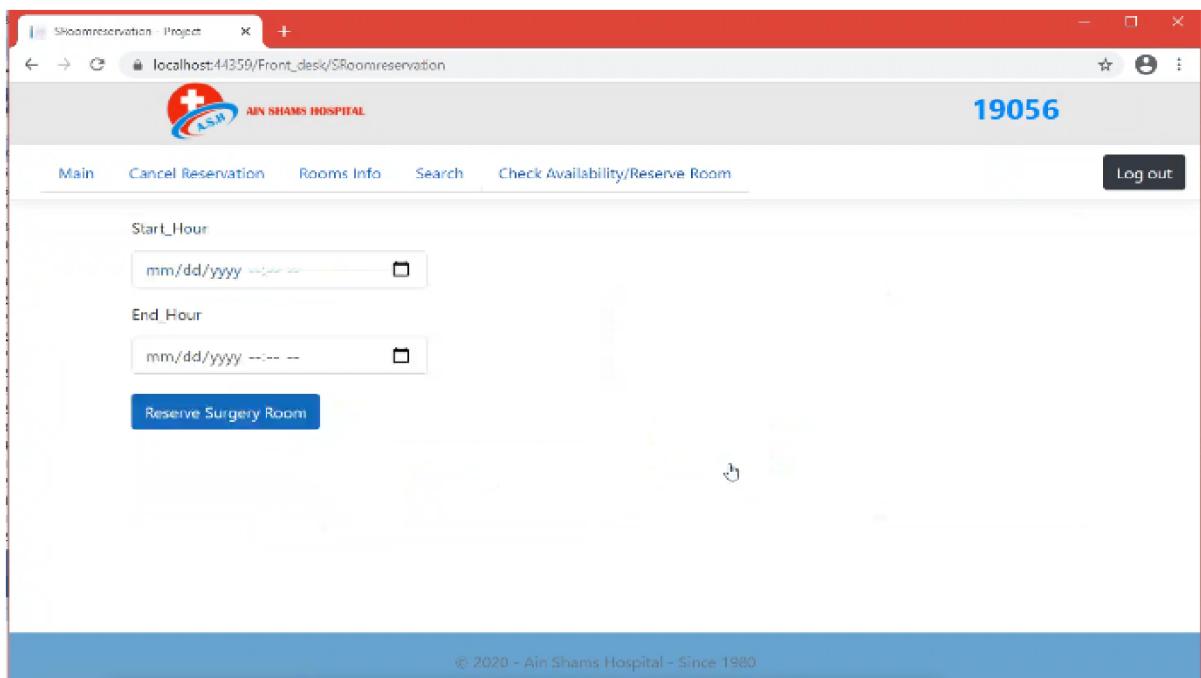


Figure 73 User Interface: show room reservation page

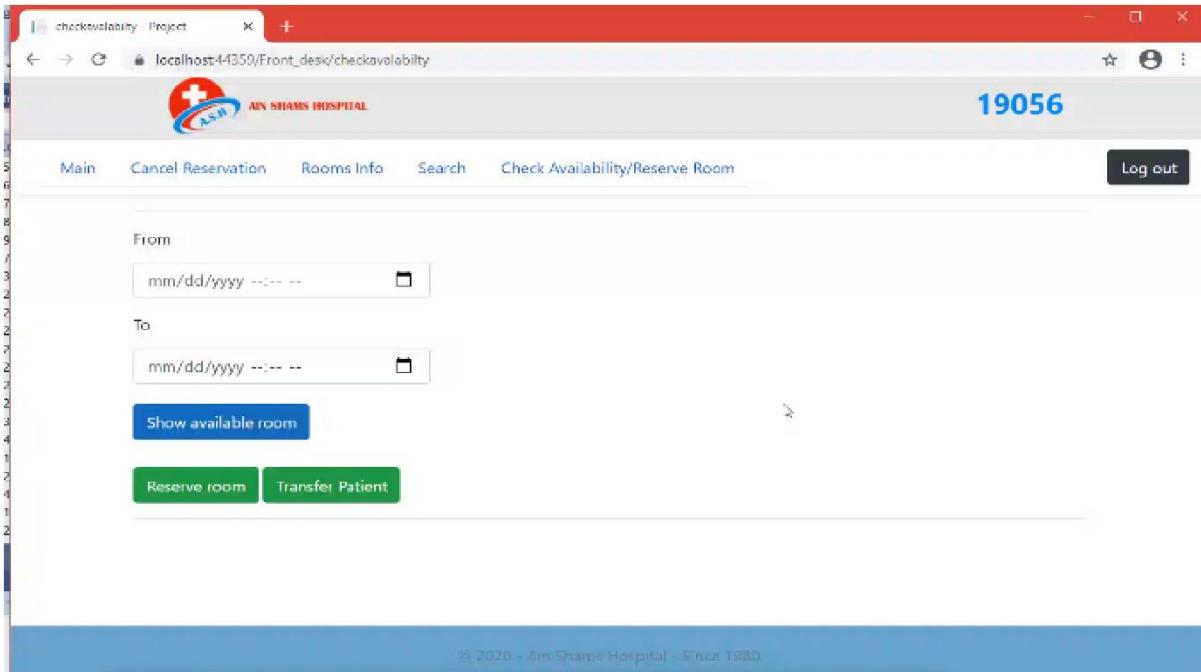


Figure 74 User Interface: check available room page

5.2 Search:

- User can write patient name and date to check if the patient is in the room or not.
- Interaction style "Form fill"

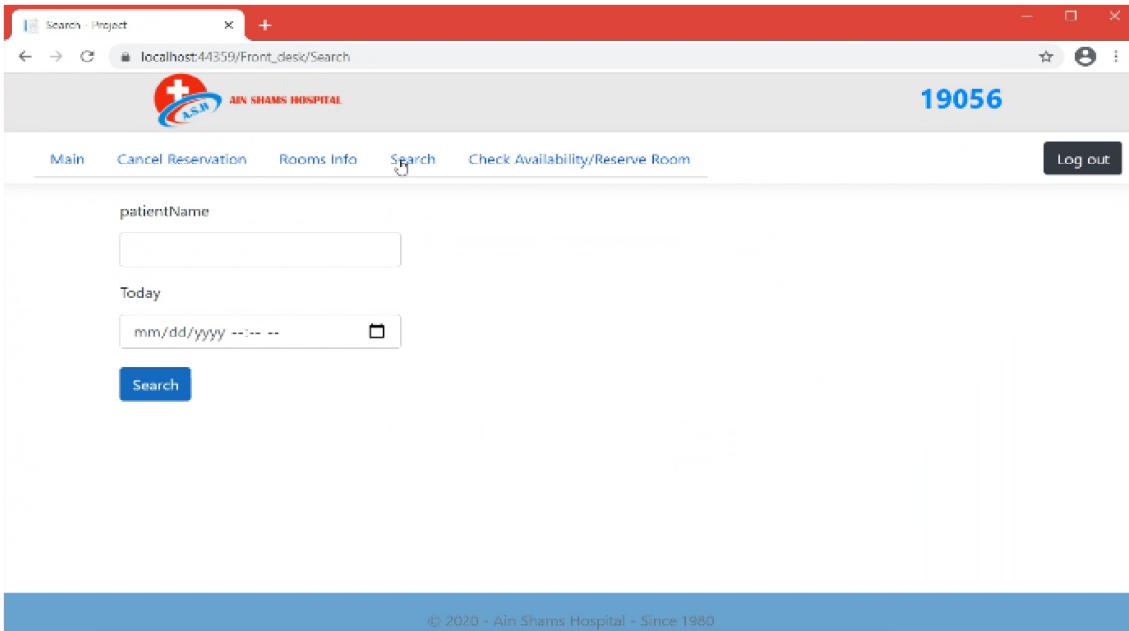


Figure 75 User Interface: Search page

5.3 Rooms info:

Interaction style "Menu selection"

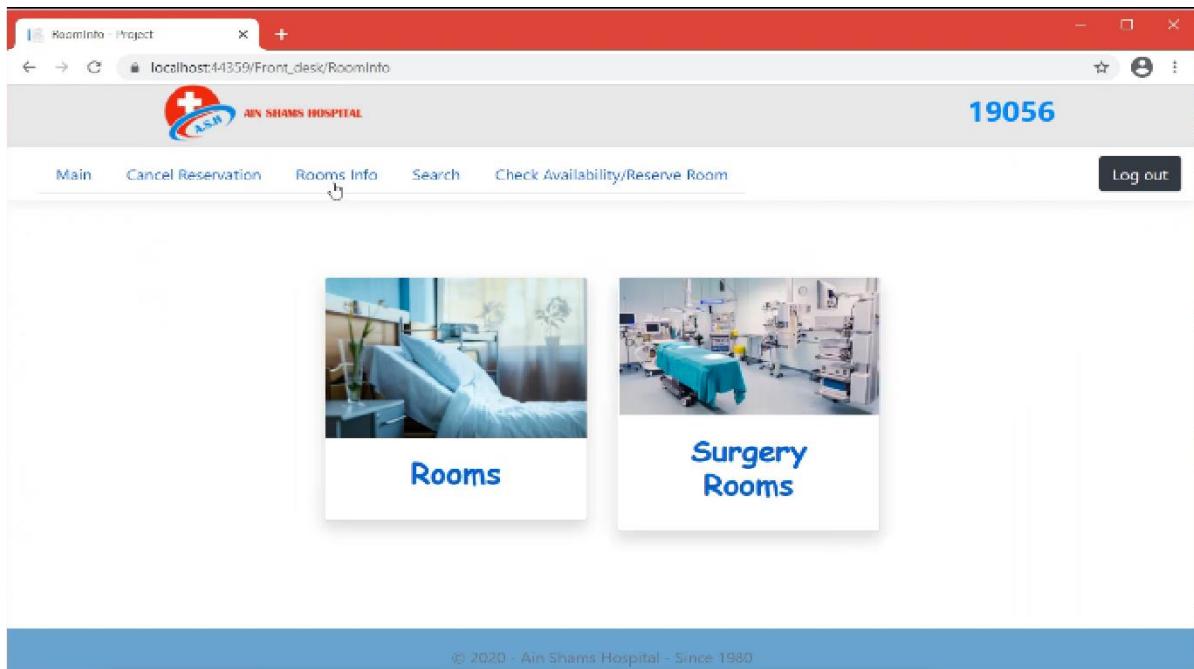


Figure 76 User Interface: Room info page

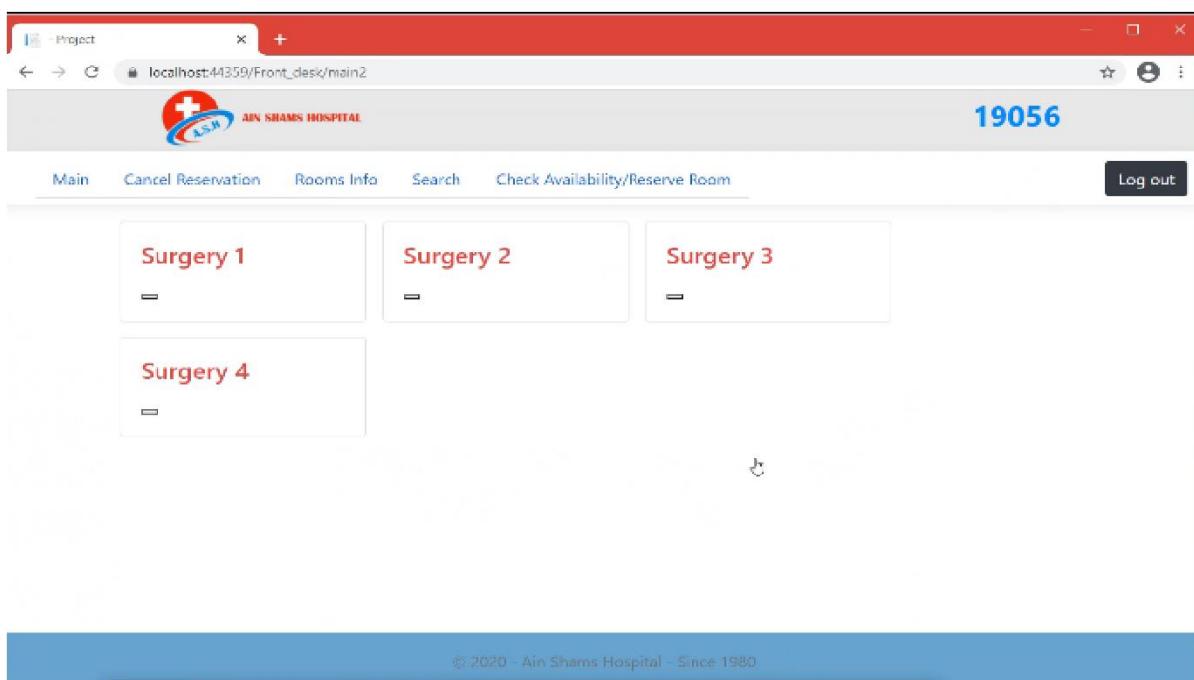


Figure 77 User Interface: surgery rooms page

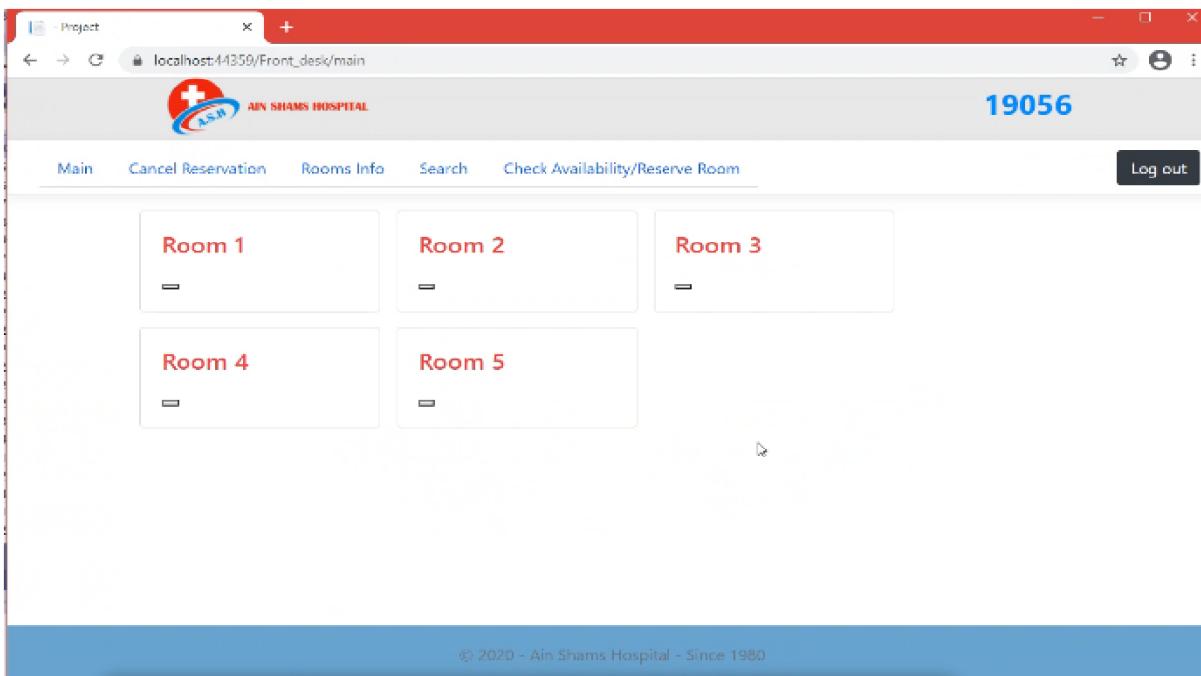


Figure 78 User Interface: rooms page

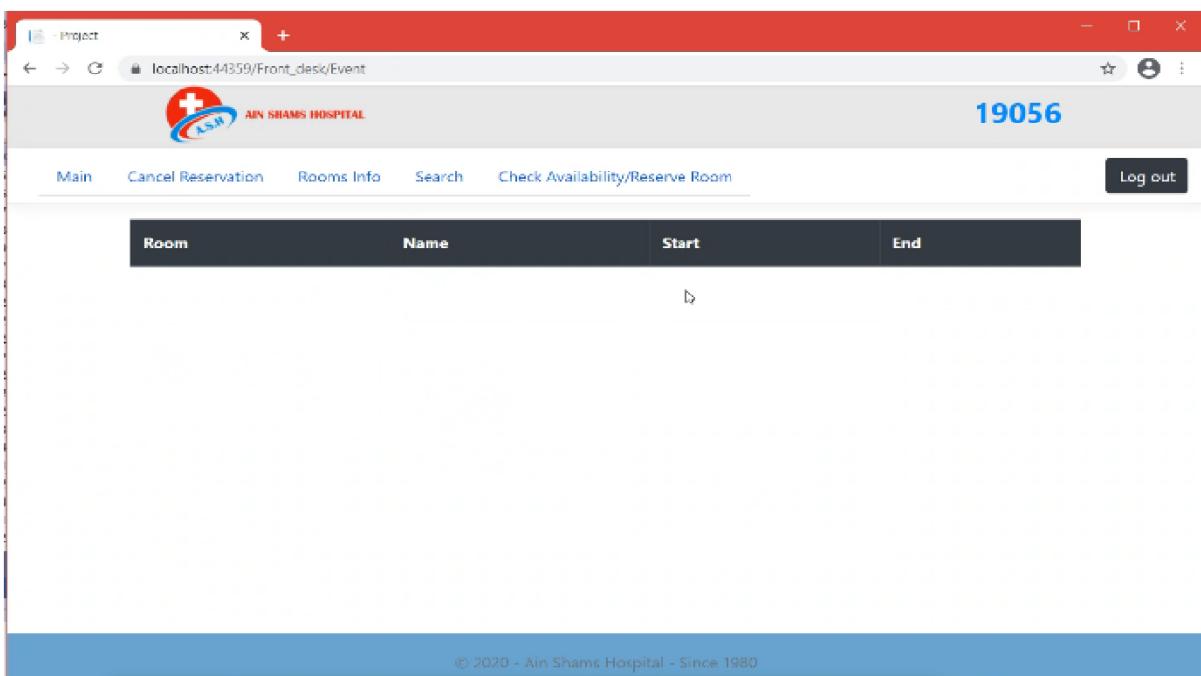


Figure 79 User Interface: Room details page

5.4 Cancel Reservation:

- Interaction style "Form fill"

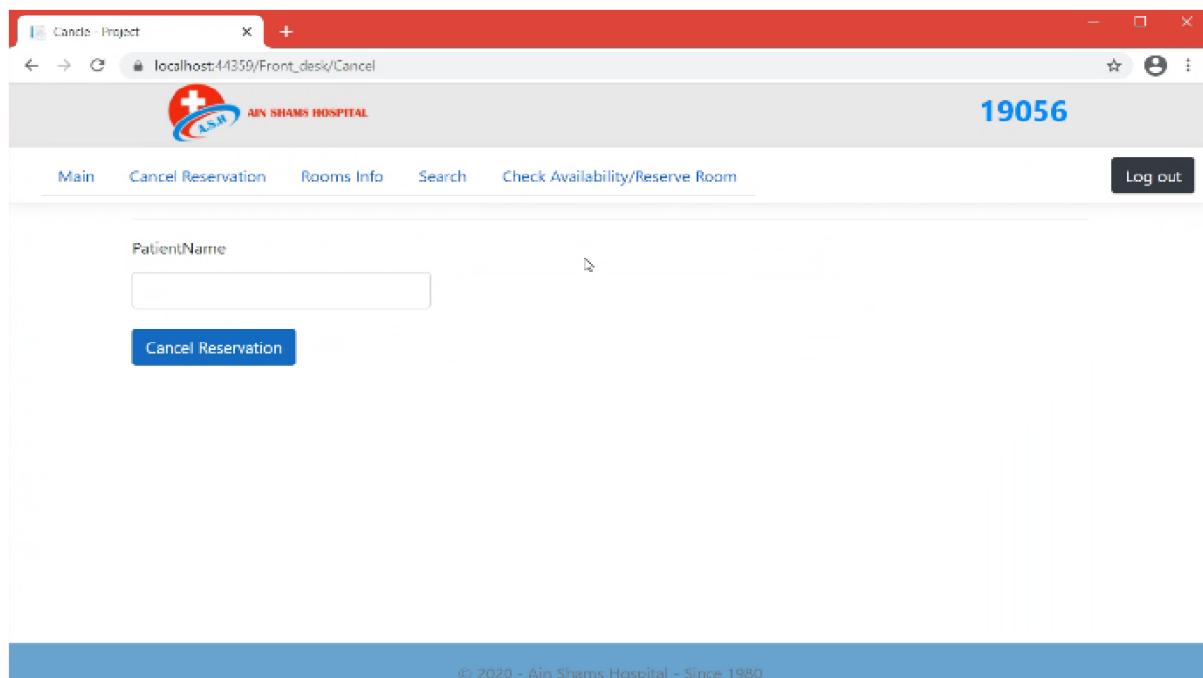


Figure 80 User Interface: cancel reservation page

6. Finance page:

- Finance home page.
- He can apply different services.
- Interaction style "Menu selection"

The screenshot shows a user interface for a hospital's finance system. At the top left is the hospital's logo with the text 'AIN SHAMS HOSPITAL'. At the top right is the number '19056'. Below this is a dark header bar with a rounded rectangle containing the text 'Welcome User Amr Mohammed'. The main content area features two images: one on the left showing a complex financial chart with multiple bars and lines, and one on the right showing a blue-themed graphic with the words 'FINANCE BILL' and a small URL 'fincaledexpress.com'. Below these images is a quote: 'A good financial plan is a road map to success.' The background of the page is light blue. At the bottom, there is a footer bar with the text '© 2020 - Ain Shams Hospital - Since 1960'.

Figure 81 User Interface: finance home page

6.1 View outcome of the hospital:

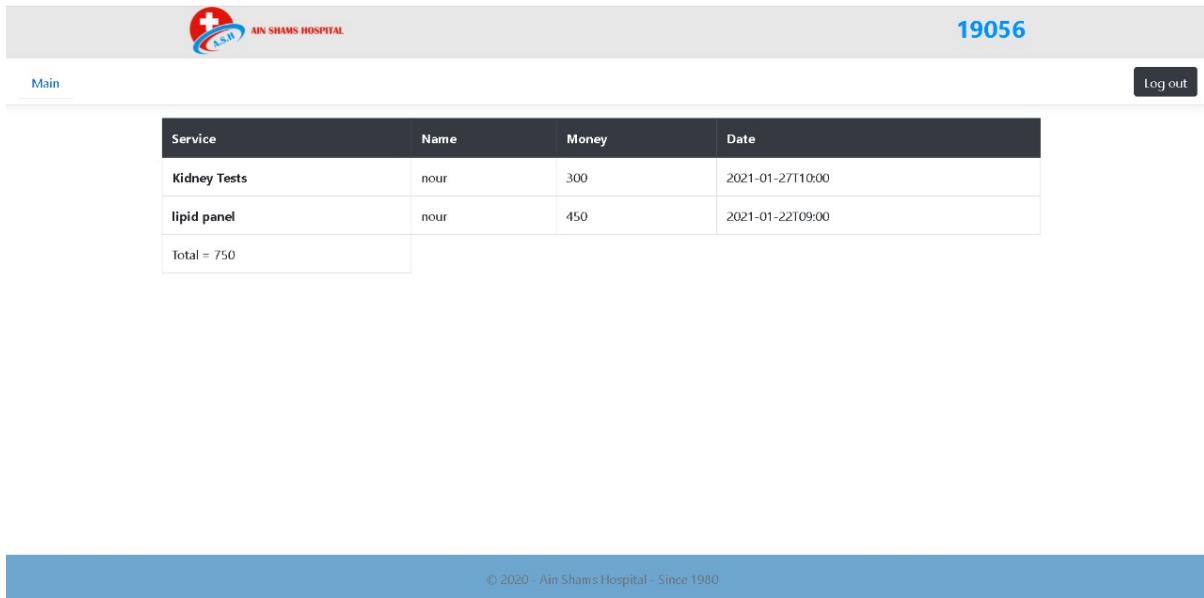
- Finance can select a certain period of time and show the outcome in it.
- Interaction style "Form fill"

The screenshot shows a user interface for viewing financial statistics. At the top right, the number '19056' is displayed. On the left, there is a logo for 'AIN SHAMS HOSPITAL' with a red cross symbol. Below the logo are two date input fields labeled 'From' and 'To', each with a calendar icon. A blue button labeled 'Set Date' is positioned between them. A teal bar at the top has the text 'Total income'. Below this, a grid of 12 boxes displays various services and their counts:

Surgery	Examination	Parasitology
—	—	—
Bacteriology	Cancer Tests	Liver Tests
—	—	—
Corona Tests	Urinalysis	Kidney Tests
—	—	—
Magnetic Resonance Imaging	Mammography	Fluoroscopy
—	—	—
Ultrasound	X-ray Lab	Glucose tests
—	—	—
Complete Blood Count	lipid panel	Basic metabolic panel
—	—	—

At the bottom, a blue footer bar contains the text '© 2020 - Ain Shams Hospital - Since 1900'.

Figure 82 User Interface: financial statistics page

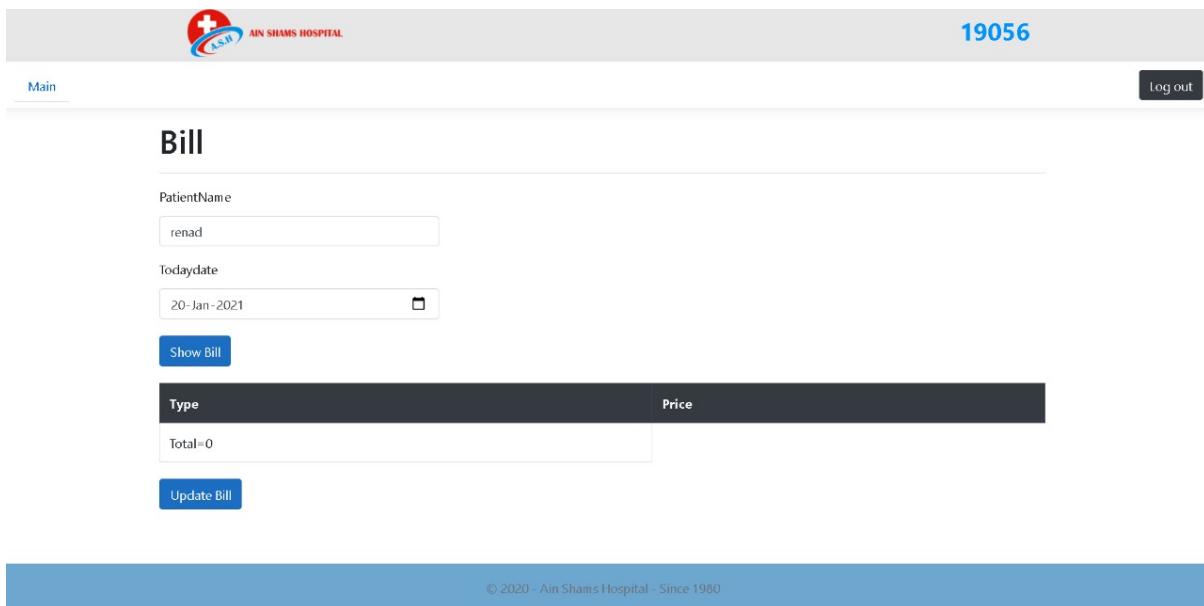


The screenshot shows a user interface for managing financial details of services. At the top right, the number '19056' is displayed. On the left, there's a logo for 'AIN SHAMS HOSPITAL' with a red cross and the letters 'A.S.H'. On the right, a 'Log out' button is visible. Below the header, a table lists service details:

Service	Name	Money	Date
Kidney Tests	nour	300	2021-01-27T10:00
lipid panel	nour	450	2021-01-22T09:00
Total = 750			

At the bottom of the page, a blue footer bar contains the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 83 User Interface: services financial details page



The screenshot shows a user interface for generating a patient bill. At the top right, the number '19056' is displayed. On the left, there's a logo for 'AIN SHAMS HOSPITAL' with a red cross and the letters 'A.S.H'. On the right, a 'Log out' button is visible. The page title is 'Bill'. It includes fields for 'PatientName' (containing 'renad') and 'Todaydate' (containing '20-Jan-2021'). A 'Show Bill' button is present. Below these, a table shows bill details:

Type	Price
Total=0	

A 'Update Bill' button is located at the bottom left of the table area. A blue footer bar at the bottom contains the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 84 User Interface: patient bill page

16.Detailed design:

I. Actions in Registration controller:

In Activation Action():

Select the list of specializations code

If (entered code Exist in the list)

{

If (code is 0000)

 redirect to patient

Else

{

 store the specialization id of this activation code and redirect to staff or front
 desk

}

Else

{

 redirect to the same view with a note to check your input

}

In About Action():

Return a view that show our vision, mission and core values

In Registration Patient Action():

If>Email Exists on the registered emails)

{

return a view to show that you have already signed up

}

Else

{

Add the entered email and password to the database in the registration table and add name ,phone entered by user in patient table in database and redirect to login

}

In Registration Staff Action:

If>Email Exists on the registered emails)

{

return a view to show that you have already signed up

}

Else

{

Add the entered email and password to the database in the registration table and add name , phone entered by user in staff table in database and get the code first number to know if the user is patient , staff , front desk , manager , lab ..

}

In Not Log Action():

Return view that show the user that his login failed

In Login Action():

If(the entered email exist)

{

get the entered password

if(the entered password after bcryption equals the bcripted password in database)

{

we check the code of the specialization of the entered email

if(code is null)

{

redirect to patient view

}

else

{

redirect according to the entered email code of specialization for lab
,doctor, front desk ...}

redirect to staff

}

}

In Log Out Action:

Redirect to view to show the user that he is not logged in

II. Actions in Patient controller:

In Home Action():

Get the registered patient id

Get his name to view it in the main page

In Select Lab Action():

Get the registered patient id

Get his name to view it in the main page

List Specializations for lab which is (Radiology Specialist, Blood Bank Specialist, lab specialist) in a dropdown list to allow the patient to choose which one to select .

Set session for the selected choice Id (sp_Id)

Redirect to lab action

In Lab Action():

Get the Id of the selected choice Using session (sp_Id)

List Follow Up Types (Which tests the patient want which is Radiology Specialist Tests, Blood Bank Specialist Tests, lab specialist Tests) according to the selected lab in a dropdown list to allow the patient to choose which one to select .

If (lab specialist)

{

 List of Follow up Types Contains Lab Specialist Tests

}

Else If (Blood Bank)

{

 List of Follow up Types Contains Blood Bank Tests

}

If (Radiology Specialist)

{

 List of Follow up Types Contains Radiology Specialist Tests

}

Get the registered patient id and sp_Id from session

Get the staff id where specialization id equals sp_Id

Add the medical record entered by the patient on database on patients table where id equals the current patient id

Add Data (Patient id , status="Pending", staff id) to Follow up table in database where patient id is the current patient id and staff id is the selected staff id

Add Data (Date , follow up type id , follow up id) to follow up history table in database where date equals date entered by the patient concatenated with "T" then concatenated with Hour entered by patient

Set session choose _test to get the id of the selected test

If (payment Online)

{

 redirect to Payment action

}

Else

{

 redirect to offline pay action

}

In Select Doctor Action():

Get Patient registration id and then get his name to view it in a view page

List Specializations for Doctor which is (Eyes Doctors , Baby's Doctor, Oncologist , Doctor of Rheumatology ... etc.) in a dropdown list to allow the patient to choose which one to select .

Set session for the selected choice Id (Doctor_sp_Id)

Redirect to Doctor action

In Doctor Action():

Get the Id of the selected choice Using session (Doctor_sp_Id)

List Staff according to the selected Doctor specialization (where specialization Id equals Doctor_sp_Id) in a dropdown list to allow the patient to choose which one to select .

Get the registered patient id and current patient id

Add the medical record entered by the patient on database on patients table where id equals the current patient id

Add Data (Patient id , status="Pending", staff id) to Follow up table in database where patient id is the current patient id and staff id is the selected staff id

Add Data (Date , follow up type id , follow up id) to follow up history table in database where date equals date entered by the patient concatenated with “T” then concatenated with Hour entered by patient

Set session choose_test to get the id of the selected test

If (payment Online)

{

redirect to Payment action

}

Else

{

redirect to offline pay action

}

In Surgeon Action():

List Staff of the surgeon (where specialization Id equals surgeon specialization id) in a dropdown list to allow the patient to choose which one to select

Get the registered patient id and current patient id

Add the medical record entered by the patient on database on patients table where id equals the current patient id

Add Data (Patient id , status=“Pending”, staff id) to Follow up table in database where patient id is the current patient id and staff id is the selected staff id

Add Data (Date , follow up type id , follow up id) to follow up history table in database where date equals date entered by the patient concatenated with “T” then concatenated with Hour entered by patient

Set session choose_test to get the id of the selected test

If (payment Online)

{

 redirect to Payment action

}

Else

{

 redirect to offline pay action

}

In Services Action():

Return the view services

In Followed Doctors Action():

Get the registered patient id and current patient id

Get the staff ids of the doctor this patient follow in a variable called ids

Get the follow up history id where date is empty that indicates that this patient is transferred to another hospital in a variable called (fuph_id)

Get follow up id where date is empty and follow up history id equals (fuph_id) variable in a variable called (id_folup)

Get the first empty date from follow up history table in (Dat) Variable

Get Staff id where follow up id equals id_folup and patient id equals current patient id

If (Date is Empty)

{

Set session named st_ID that contains staff id

Set session named Followup_Id That contains id_folup

Set session named Fuph_Id that contains fuph_id

}

Else

{

List Staff of the followed doctors (where staff Id equals ids) in a dropdown list to allow the patient to choose which one to select and view his schedule

Set Session called sel_Id contains the selected doctor id

Redirect to Doctor Schedule Action

}

In Date Edit Action():

Get session of st_ID and doctor name to show it in the view

Get session of Followup_Id and Fuph_Id

Add date to follow up history table and update database

Redirect to followed doctors

In Doctor Schedules Action():

Get the registered patient id and current patient name

In this action we get the working days according to the specialization of the doctor and show the schedule of the followed doctors in a table to the patient

In Offline pay Action():

Get the registered patient id and current patient id

We add the data (patient_Id ,Money,follow_up_type_Id,Date,Online) in the payment table if the patient will pay offline.

Note: Online is a Boolean variable which will be false in this case

In Payment Action():

Get the registered patient id and current patient id

We add the data (patient_Id ,Money,follow_up_type_Id,Date,Online) in the payment table if the patient will pay online.

Note: Online is a Boolean variable which will be true in this case

In Save Pay Action:

This view the page that show the patient the his payment is successfully recorded to the database in online payment

III. Actions in Front desk Controller:

In main Action():

Go to Hospital_Facilities table in database and get the Type rows which contains the first 4 letters = "Room" and save it in list (NameRexist)
Return (NameRexist) list in cards form

In main2 Action():

Go to Hospital_Facilities table in database and get the Type rows which contains the first 7 letters = "Surgery" and save it in list (NameSexist)
Return (NameSexist) list in cards form

Event Action(mainVM m):

In this action the Inputs are : m is an object from view model class mainVM
Go to Hospital_Facilities table in database and compare (m.Room_Id)with Column Type and get the Id of the row and save it in variable (ID)
Go to Facility_Reservations table in database and compare ID with Column Hospital_Facility_Id and put those rows in list arrange in descending order according to the End_Hour saved in the table and save it in variable(h)
Return table contains room details

Cancel Action(delete vm):

In this action the Inputs are : vm is an object from view model class delete
The user enter the name of the patient who want to transfer him from one room to another (vm.PatientName)
Go to table patients in the database and compare all the rows of column Name with (vm.PatientName) and see if it is exist in variable (NameExist)
If(NameExist) is true:

{

 Go to table patients in the database and compare all the rows of column Name with (vm.PatientName) and get its Id and save it in variable (PatientId)

Go to table facility reservation in the database and compare all the rows of column Patient_Id with (PatientId) and see if it is exist in variable (NameExistreservation)

If(NameExistreservation == true)

{

 Go to table facility reservation in the database and compare all the rows of column Patient_Id with (PatientId) and get all the rows for this Patient_Id and order them from the latest reservation to the oldest (descending order according to the facility reservation table Id) then select the first item from that list and save it in variable (facilityreservationId)

 Go to table Follow_up_Type in the database and compare all the rows of column Name with "Surgery" and get it's Id and save it in variable (surgeryId)

 Go to table Payments in the database and(compare all the rows of column Follow_up_Type_Id with (surgeryId) compare all the rows of column Patient_Id with (PatientId))and get all the rows for this conditions and order them (descending order according to thePayment table Id) then select the first item from that list and save it in variable (surgeryfollowuptypeId)

 Go to table Payments in the database and find the row with Id= (surgeryfollowuptypeId) and save all the row in variable(model3) and remove the row (model3)

 Go to table facility reservation in the database and find the row with Id= (facilityreservationId)and save all the row in variable(model1) and remove the row (model1)

 Go to table facility reservation in the database and compare all the rows of column Patient_Id with (PatientId) and get all the rows for this Patient_Id and order them from the latest reservation to the oldest (descending order according to the facility reservation table Id) then select the first item from that list and save it in variable (FacilityreservationId)

Go to table Follow_up_Type in the database and compare all the rows of column Name with “Room” and get it’s Id and save it in variable (RoompriceId)

Go to table Payments in the database and(compare all the rows of column Follow_up_Type_Id with(RoompriceId) compare all the rows of column Patient_Id with (PatientId))and get all the rows for this conditions and order them (descending order according to thePayment table Id) then select the first item from that list and save it in variable (followuptypeId)

Go to table facility reservation in the database and find the row with Id= (FacilityreservationId)and save all the row in variable(model) and remove the row (model)

Go to table Payments in the database and find the row with Id= (followuptypeId) and save all the row in variable(model2) and remove the row (model2)

Then a message will show to the user says “Reservation is canceled successfully”

}

If(NameExistreservation== false)

{

A message will be returned ”This Patient doesn’t have reservation”

}

}

If(NameExist == false)

{

A message will be returned ”This Patient isn’t in our hospital”

}

SRoomreservation Action(SRoomReservationVM ob):

In this action the Inputs are : ob is an object from view model class SRoomReservationVM
The User enters ob.Start_Hour and ob.End_Hour
Go to Hospital_Facilities table in database and get the Type rows which contains the first 7 letters = "Surgery" and save it in list (NameSurgeryRexist)
Declare an inter flag f=0
Declare variable x which Lop in each value in list (NameSurgeryRexist)
{
 Go to Hospital_Facilities table in database and compare x.Type with Column Type and get the Id of the row and save it in variable (SHID)
 Go to Faciity_Reservations table in database and compare SHID with Column Hospital_Facility_Id and select from those rows the (Start_Hour and save it in date1 and the End_Hour and save it in date) save it in list (Savailableroom)
 Declare variable V which Lop in each value in (Savailableroom)list
 {
 Change ob.Start_Hour , ob.End_Hour,V.date and V.date1 from string type to date time type and save them in parse1, parse2, parse3 and parse4 relatively
 If (parse1,>parse3)OR(parse2 < parse4)
 {
 Make f=1
 }
 Else
 {
 Make f=2}
 }
 If (f=0 Or f=1)

```

{
    Go to Hospital_Facilities table in database and compare SHID with
    Column Id and get the Type of the row and save it in variable
    (Sroomname)
    Save (Sroomname) in string session (SRoomname)
    Save (ob.Start_Hour) in string session (ENDSurgery)
    Save (ob.End_Hour) in string session (STARTSurgery)

    Go to action check availability

}
Change the flag f to 0 }

Return Message "This Surgery room is not available"

```

Checkavalibilty Action(checkVM ch):

In this action the Inputs are : ch is an object from view model class checkVM,
string SURGERYROOMName ,string surgerystart ,string surgeryend(from
SRoomreservation action)

The User enters ch.From and ch.To

Go to Hospital_Facilities table in database and get the Type rows which contains the
first 4 letters = "Room" and save it in list (NameRexist)

Go to Hospital_Facilities table in database and get the Type rows which contains the
first 7 letters = "Surgery" and save it in list (NameSexist)

Declare an inter flag =y=0

Declare variable x which Lop in each value in (NameRexist) list

```

{
    Go to Hospital_Facilities table in database and compare x.Type with Column
    Type and get the Id of the row and save it in variable (HID)
}

```

Go to Facility_Reservations table in database and compare HID with Column Hospital_Facility_Id and select from those rows the (Start_Hour and save it in date and the End_Hour and save it in date1) save it in list (availableroom)
 Declare variable V which Lop in each value in (availableroom)list
 {
 Change ch.From ,ch.To ,V.date and V.date1 from string type to date time type and save them in parse1, parse2, parse3 and parse4 relatively

 If (parse1,>parse3)OR(parse2 < parse4)
 {
 Make y=1
 }
 Else :
 {
 Make y=2
 }
 If (y=0 Or y=1)
 {
 Go to Hospital_Facilities table in database and compare HID with Column Id and get the Type of the row and save it in variable (roomname)

 Save (roomname) in string session (Roomname)

 Save (ch.To) in string session (END)

 Save (ch.From) in string session (START)
 Return (roomname)
 }

Change the flag y to 0
}

Return Message "This room is not available "

Roomreservation(roomVM ob , checkVM ch)

In this action the Inputs are : ch is an object from view model class checkVM, ob is an object from view model class roomVM,

string SURGERYROOMName ,string surgerystart ,string surgeryend(from SRoomreservation action)

string ROOMName ,string start ,string end(from checkavalability action)

The user enter the name of the patient who is reserving room to (ob.PatientName) and the doctor who is following up with (ob.DoctorName)

Initialize two objects from class Facility reservation (FR)&(F)

initialize two objects from class Payment (P)&(P1)

Go to table patients in the database and compare all the rows of column Name with (ob.PatientName) and see if it is exist in variable (NamePEexist)

Go to table staffs in the database and compare all the rows of column Name with (ob.DoctorName) and see if it is exist in variable (NameSEexist)

Go to table Follow _Up_Types in the database and compare all the rows of column Name with "Surgery" and select the Price for this row and save it in variable (surgeryprice)

Go to table Follow _Up_Types in the database and compare all the rows of column Name with “Room” and select the Price for this row and save it in variable (roomprice)

Go to table Follow _Up_Types in the database and compare all the rows of column Name with “Surgery” and select the Id for this row and save it in variable (surgerypriceld)

Go to table Follow _Up_Types in the database and compare all the rows of column Name with “Room” and select the Id for this row and save it in variable (roompriceld)

Change start and end from string type to date time type and save them in parse1 and parse2 relatively

Make variable Duration = parse2- parse1

Integer days =days in (Duration)+1

Integer Totalroomprice = days*(roomprice)

If(NamePExist) is true:

{

Go to table patients in the database and compare all the rows of column Name with (ob.PatientName) and get its Id and save it in TempData (Patient_Id)

If(NameSExist) is true:

{

Go to table Staffs in the database and compare all the rows of column Name with (ob.DoctorName) and get its Id and save it in TempData (Staff_Id)

if SURGERYROOMName is null

{

Change TempData (Staff_Id) & TempData (Patient_Id) into integer

Save in object of class facility reservation (FR) Patient information in new row in facility reservation table

Start_Hour=start(input), End_Hour=end(input)

,Hospital_Facility_Id=HID, Patient_Id= TempData (Patient_Id) after casting, Staff_Id= TempData (Staff_Id) after casting

Add this row in database and save

Save in object of class Payment (P1) Patient information in new row in Payments table

Patient_Id= TempData (Patient_Id) after casting ,

Staff_Id= TempData (Staff_Id) after casting,

Money= Totalroomprice, Follow_Up_Id=(roomprice)

Add this row in database and save

Return to the user message" Done"

}

Else

{

Go to Hospital_Facilities table in database and compare SURGERYROOMName with Column Type and get the Id of the row and save it in variable (SHID)

Save in object of class facility reservation (F) Patient information in new row in facility reservation table

Start_Hour= surgerystart, End_Hour=surgeryend,
Hospital_Facility_Id=SHID,

Patient_Id= TempData (Patient_Id) after casting,
Staff_Id= TempData (Staff_Id) after casting

Add this row in database and save

Save in object of class facility reservation (F) Patient information in new row in facility reservation table

Start_Hour=start(input), End_Hour=end(input)
,Hospital_Facility_Id=HID, Patient_Id= TempData (Patient_Id) after casting, Staff_Id= TempData (Staff_Id) after casting

Add this row in database and save

Return to the user message " Done"

}

}

Else

{

Return to the user message " This Doctor is not in our hospital"

```

        }

    Else
    {

        Return to the user message" This Doctor is not in our hospital"

    }

}

```

In Delete Action(delete vm,checkVM ch):

In this action the Inputs are : vm is an object from view model class delete, ch is an object from view model class checkVM,

string ROOMName ,string start ,string end(from checkavailability action)

The user enter the name of the patient who want to transfer him from one room to another
(vm.PatientName)

So the system will go to table patients in the database and compare all the rows of column Name with (vm.PatientName) and see if it is exist in variable (NameExist)

If(NameExist) is true:

```

{
    Go to table patients in the database and compare all the rows of column Name with
    (vm.PatientName) and get its Id and save it in variable (PatientName)
}

```

Go to table facility reservation in the database and compare all the rows of column Patient_Id with (PatientName) and see if it is exist in variable (NameExistreservation)

If(NameExistreservation) is true:

{

Go to table facility reservation in the database and compare all the rows of column Patient_Id with (PatientName) and get all the rows for this Patient_Id and order them from the latest reservation to the oldest (descending order according to the facility reservation table Id) then select the first item from that list and save it in variable (h)

The system will go to table Hospital_Facilities in the database and compare all the rows of column Type with (ROOMName)and get that row Id and save it in variable (HID)

Make object of class facility reservation (FR) to save that patient information in new row in facility reservation table

Start_Hour=start(input), End_Hour=end(input) ,Hospital_Facility_Id=HID,
Patient_Id=PatientName,
Get staff_Id(Doctor Id) from the row h and save it in Staff_Id

Add FR object to the database facility reservation table and save

From h get the row Id and save it in Integer x

Go to table facility reservation in the database and find the row with Id=x and save all the row in variable(model)

The system will go to table facility reservation in the database and remove the row (model)

Then a message will show to the user says “Patient is transferred successfully”

}

If(NameExistreservation) is false:

{

A message will show to the user says “This Patient doesn’t have reservation”

}

}

If(NameExist) is false:

{

A message will show to the user says “This Patient doesn’t have reservation”

}

Search Action(SearchVm ob):

In this action the Inputs are : ob is an object from view model class SearchVM

The user enter the name of the patient who want to transfer him from one room to another
(ob.PatientName)

Go to table patients in the database and compare all the rows of column Name with
(ob.PatientName) and see if it is exist in variable (NameExist)

If(NameExist) is true:

{

Go to table patients in the database and compare all the rows of column Name
with (ob.PatientName) and get its Id and save it in variable (ID)

Go to table facility reservation in the database and compare all the rows of column Patient_Id with (ID) and see if it is exist in variable (NamePExist)

If(NamePExist) is true:

{

Go to table facility reservation in the database and compare all the rows of column Patient_Id with (ID) and get all the rows for this Patient_Id and order them from the latest reservation to the oldest (descending order according to the facility reservation table End_Hour) then select the first item from that list and save it in list (Endavailable)

Change Start_Hour and End_hour from string to DateTime Parse3 &parse4 respectfully

If (Parse3 <= ob.Today) And (ob.Today<=parse4)

{

Go to table facility reservation in the database and compare all the rows of column Id with (Endavailable.name)and select the Hospital_Facility_Id save it in variable (roomnumber)

Go to table Hospital_Facilities in the database and compare all the rows of column Id with (roomnumber)and select the Type save it in TempData[“room”]}

Else

{ Return Message to the user “This Patient is not in our hospital”} }

Return Message to the user “This Patient is not in our hospital”}

Return Message to the user “This Patient is not in our hospital” }

IV. Actions in Manager Controller:

Action ManagerHome (StaffName)

The user (Manager) enters the name of the staff member he/she wants to search for their information (StaffName) in ViewModel (ManagerSearch).

Enter [Staff Table] and compare all values in column “Name” to (StaffName)

If (NameExist)

{

Enter [Staff Table], select Name, Phone, Starting_Day, Specialization_Id,

Registration_Id and store them in variables (member), (membrer1),(member2), (x)

And (registration_id) respectively.

Enter [Specializations Table] and compare all values in column “Id”

where Id=(x) and select corresponding Name and store it in (member3)

Enter [Registrations Table] and compare all values in column “Id”

where Id=(registration_id) and select corresponding Email and store it in (mail)

Display (member), (membrer1), (member2), (member3), (mail)

}

Else {

Display “Staff member doesn’t exist”

}

Action DeleteManager (StaffName)

The user (Manager) enters the name of the staff member he/she wants to search for their information (StaffName) in ViewModel (deletevm).

Enter [Staff Table] and compare all values in column “Name” to (StaffName)

If (NameExist)

{

Enter [Staff Table], select corresponding Id, Registration_Id and store them in

variables (StaffName), (registrationId) respectively.

Enter [Follow_Ups Table] and compare all values in column “Staff_Id” to (StaffName)

and select them all in a list and store it in (followupid)

foreach (every selected value in the (followupid) stored in (x)){

Enter [Follow_Ups_History Table] and compare all values in column “Follow_Up_Id” to (x)

and select the corresponding Id and store it in (followuphistory)

Find every row in [Follow_Ups_History Table] whose Id=(followuphistory) and delete them.

Find every row in [Follow_Ups] whose Id=(x) and delete them.

}

Find the row in [Registrations] whose Id=(registration id) and delete it.

Find the row in [Staff] whose Id=(StaffName) and delete it .

Display “Staff member is deleted successfully”

```
 }

Else {

    Display "This staff member doesn't exist" }
```

Action DeleteSpecialization (SpecializationName):

The user (Manager) enters the name of the specialization he/she wants to delete (SpecializationName) in ViewModel (AddSpectalization).

Enter [Specializations Table] and compare all values in column “Name” to (SpecializationName)

If (NameExist)

{

Enter [Registrations Table], select corresponding Id, and store them in variable (SpecializationName).

Find the row in [Registrations] whose Id=(SpecializationName) and delete it.

Display “Specialization code is deleted successfully”

}

Else {

Display “This specialization doesn't exist”

}

Action DeletePrice (ServiceName)

The user (Manager) enters the name of the service he/she wants to delete (ServiceName) in ViewModel (AddPrice).

Enter [Follow_Ups_Types Table] and compare all values in column “Name” to (SpecializationName)

If (NameExist)

{

Enter [Follow_Ups_Types Table], select corresponding Id, and store them in variable (ServiceName).

Find the row in [Follow_Ups_Types] whose Id=(ServiceName) and delete it.

Display “This service is deleted successfully”

}

Else {

Display “This service doesn’t exist”

}

Action ViewPrice ()

Enter [Follow_Ups_Types Table] and select every “Name” with its corresponding “Price” and store the list in variable (s)

Display the list (s)

Action ManagerEdit (Name,Price)

The user (Manager) enters the name of the service he/she wants to edit (Name) and the new price

(Price) in ViewModel (ManagerEdit).

(v.Name) = (Name), (v.Price) = (Price)

Enter [Follow_Ups_Types Table] and compare all values in column “Name” to (v.Name)

If (NameExist)

{

Enter [Follow_Ups_Types Table], select corresponding Id, and store them in variable (m)

Select Price of corresponding Id and change it with (Price).

Display “The price is edited successfully”

}

Else {

Display “This name doesn’t exist”

}

Action ViewSpecializations ()

Enter [Specializations Table] and select every “Name” with its corresponding “Code” and store the list in variable (s)

Display the list (s)

Action ViewHospital ()

Enter [Transfer_Hospitals Table] and select every “Name” and store the list in variable (s)

Display the list (s)

Action AddHospital (HospitalName)

The user (Manager) enters the name of the hospital he/she wants to add (HospitalName) in ViewModel (AddHospital).

(b.Name) = (HospitalName)

Enter [Transfer_Hospitals Table] and compare all values in column “Name” to (b.Name)

If (NameExist)

{

Display “This hospital already exists”

}

Else {

Variable (name) = (b.name)

Add (b.name) to [Transfer_Hospitals Table] and save changes

Display “Hospital is added successfully”

}

Action AddPrice (ServiceName,Price)

The user (Manager) enters the name of the service he/she wants to add (ServiceName) and its price

(Price) in ViewModel (AddPrice).

Variable (b.name) = ServiceName

Enter [Follow_Ups_Types Table] and compare all values in column “Name” to (b.Name)

If (NameExist)

{

Display “This service already exists”

}

Else {

Variable (b.Price) = (Price)

Add (b.name) and (b.Price) to [Follow_Ups_Types Table] and save changes

Display “Service is added successfully”

}

Action AddSpecializations (Specialization,Code)

The user (Manager) enters the name of the specialization he/she wants to add (Specialization) and its

code (Code) in ViewModel (AddSpecialization).

Variable (b.Name) = (Specialization), variable (b.Code) = (Code)

Enter [Specializations Table] and compare all values in column “Name” to (b.Name)

Enter [Specializations Table] and compare all values in column “Code” to (b.Code)

If (CodeExist) {

 Display “This code is already taken”

}

Else if (NameExist) {

 Display “This specialization already exists”

}

Else {

 Variable (code) = (b.Code)

 Integer variable (_index) = first number in (b.code)

 If (_index == 1) {

 Add (b.Name), (b.code) to [Specializations Table]

 Save changes

 Display “Specialization is added successfully”}

 Else {

 Display “Your code must start with 1”

}

}

Action State (From,To)

The user (Manager) enters the time interval (From), (To) in ViewModel (FinanceInterval).

Enter [Follow_Ups_Types] and select all values and store them in list (Type)

Display list (Type)

Get (From) and save it in (FIRST)

Return (FIRST)

Get (To) and save it in (LAST)

Return (LAST)

Action StateView (Room_Id)

The user (Manager) _Id) sets the type name (Room_Id) by clicking on a card

Enter [Follow_Ups_Types Table] and compare all values in column “Name” to (Room_Id) and select

the corresponding Id and store it in variable (ID)

String variable (from) = Get returned value (FIRST) from “Action State”

String variable (to) = Get returned value (LAST) from “Action State”

If (User entered time interval)

{

Enter [Payments Table], include [Patient] and [Follow_Ups_Types], select all rows that fulfill the condition [(Follow_Up_Type_Id)=(ID) and (Payed)=(True)] in the three tables and store the list in variable (h)

Convert string (from) to DateTime and store it in variable (start)

Convert string (to) to DateTime and store it in variable (end)

Define new list (payment) of class payment

Initialize (total) = 0

Foreach (item in list (h) stored in variable (V)) {

 Convert column value in column “Date” to DateTime and store it in variable (date)

 If ((date) lies between (start) and (end)) {

 Add (total) to previous total and value in column “Money” of [Payment] and store it in (total)

 Add variable (V) in list (payment)} }

 Display (Room_Id), (total), (payment) in a table }

Else {

 Return [Error] view which displays “You have to enter the dates From and To”

}

Action StateView1 ()

String variable (from) = Get returned value (FIRST) from “Action State”

String variable (to) = Get returned value (LAST) from “Action State”

If (User entered time interval)

{

Enter [Payments Table], include [Patient] and [Follow_Ups_Types], select all rows that fulfill

the condition [(Payed)=(True)] in the three tables and store the list in variable (h)

Convert string (from) to DateTime and store it in variable (start)

Convert string (to) to DateTime and store it in variable (end)

Define new list (payment1) of class payment

Initialize (total) = 0

Foreach (item in list (h) stored in variable (V)) {

 Convert column value in column “Date” to DateTime and store it in variable (date)

 If ((date) lies between (start) and (end)) {

 Add (total) to previous total and value in column “Money” of [Payment] and store it in (total)

 Add variable (V) in list (payment1)}

 }

 Display (total), (payment1) in a table

}

Else {

 Return [Error] view which displays “You have to enter the dates From and To” }

V. Actions in Finance Controller

Action State (From,To)

The user (Finance) enters the time interval (From), (To) in ViewModel (FinanceInterval).

Enter [Follow_Ups_Types] and select all values and store them in list (Type)

Display list (Type)

Get (From) and save it in (FIRST)

Return (FIRST)

Get (To) and save it in (LAST)

Return (LAST)

Action StateView (Room_Id)

The user (Manager) _Id) sets the type name (Room_Id) by clicking on a card

Enter [Follow_Ups_Types Table] and compare all values in column “Name” to (Room_Id) and select the corresponding Id and store it in variable (ID)

String variable (from) = Get returned value (FIRST) from “Action State”

String variable (to) = Get returned value (LAST) from “Action State”

If (User entered time interval)

{

Enter [Payments Table], include [Patient] and [Follow_Ups_Types], select all rows that fulfill the condition [(Follow_Up_Type_Id)=(ID) and (Payed)=(True)] in the three tables and store the list in variable (h)

Convert string (from) to DateTime and store it in variable (start)

Convert string (to) to DateTime and store it in variable (end)
 Define new list (payment) of class payment
 Initialize (total) = 0
 Foreach (item in list (h) stored in variable (V)) {
 Convert column value in column “Date” to DateTime and store it in variable (date)
 If ((date) lies between (start) and (end)) {
 Add (total) to previous total and value in column “Money” of [Payment] and store it
 in (total)
 Add variable (V) in list (payment)
 }
 }
 Display (Room_Id), (total), (payment) in a table
}
Else {
 Return [Error] view which displays “You have to enter the dates From and To”
}

Action StateView1 ()

String variable (from) = Get returned value (FIRST) from “Action State”

String variable (to) = Get returned value (LAST) from “Action State”

If (User entered time interval)

{

Enter [Payments Table], include [Patient] and [Follow_Ups_Types], select all rows that fulfill the condition [(Payed)=(True)] in the three tables and store the list in variable (h)

Convert string (from) to DateTime and store it in variable (start)

Convert string (to) to DateTime and store it in variable (end)

Define new list (payment1) of class payment

Initialize (total) = 0

Foreach (item in list (h) stored in variable (V)) {

 Convert column value in column “Date” to DateTime and store it in variable (date)

 If ((date) lies between (start) and (end)) {

 Add (total) to previous total and value in column “Money” of [Payment] and store it in (total)

 Add variable (V) in list (payment1)

 }

}

 Display (total), (payment1) in a table}

Else {Return [Error] view which displays “You have to enter the dates From and To”}

Bill (PatientName)

The user (Finance) enters patient’s name (PatientName) in ViewModel (payed).

Enter [Patients Table] and compare all values in column “Name” to (PatientName)

 If (NameExist)

{

Enter [Patients Table] and compare all values in column “Name” to (PatientName) and select the corresponding Id and store it in variable (patientid)

Enter [Payments Table], include [Patient] and [Follow_Ups_Types], select all rows that fulfill the condition [(Patient_Id)=(patientid) and (Payed)=(false)] in the three tables and store the list in variable (patientbill)

Initialize int variable (total) = 0

Foreach (item in list (patientbill) stored in variable (y)) {

Add (total) to previous total and value in column “Money” of [patientbill] and store it in (total)

}

Get (Todaydate) and save it in (Today)

Return (TODAY)

}

Else {

Display “Patient doesn’t exist”

}

Billresult (Checkbox)

Finance member enters Boolean value by checking the checkbox list in View model (payed) to return int list (isactive)

String variable (Date) = Get returned value (Today) from “Action Bill”

Foreach (item in list (isactive) stored in variable (mn)) {

 Enter [Payments Table] and compare all values in column “Id” to (mn) and store row in
 variable (payed)

 Value in column “Payed”=true

 Value in column “Date”= (Date)

 Save changes

 Display “Done”

}

VI. Actions in Doctor Controller:

Action Main ():

Open Staff database table and search for Registration_Id==Doc_Reg_Id

If(search Doc_Reg_Id)

{

 Return Staff_Id;

 Open Follow_Ups database table and return all patient_ids whose rows have
 Staff_Id== Staff_Id in variable (Patients_Follow_Ups);

 Create a list of patients called (patient);

 Create a list of Follow_Ups_History called (follow_up);

 Merge these 2 lists together In 1 list (patient_join_folow_up);

 Loop around all elements in Patients_Follow_Ups

{

 Add to patient list the patient whose (Id==Patient_Id in
 Patients_Follow_Ups)

 Add to follow_up list the Follow_Up whose (Follow_Up Id == Id in
 Patients_Follow_Ups);

 Add these new elements in patient_join_follow_up list;

}

 Return the value in SearchItem sent from (LabMainVM) in variable (SearchString)

 If (searchString !=Null)

```

{

    Select from the list patient_follow_up the columns from
    (Name,Id,Follow_Up_Type_Name) the column which is == SearchString
    and send it in ViewBag.follow_ups to the View;

}

Sort items in patient_join_follow_up in descending order according to Date;

Return the value in SortItem sent from (LabMainVM) in variable (SortString)

If (SortString!=Null)

{

    If (OrderItem sent from post action ==Null)

    {

        Return “DSC” in variable (OrderString);

    }

    Else

    {

        Return the value of OrderItem sent from post action in variable
        OrderString;

    }

    If (sortString==Date && OrderString==DSC)

    {

        Sort items in patient_join_follow_up in descending order
        according to Date;
    }
}

```

```
    }

    Else if (sortString==Date && OrderString ==ASC)

    {

        Sort items in patient_join_follow_up in ascending order according
        to Date;

    }

    Else if (sortString==Status && OrderString ==DSC)

    {

        Sort items in patient_join_follow_up in descending order
        according to Status;

    }

    Else if (sortString==Status && OrderString ==ASC)

    {

        Sort items in patient_join_follow_up in ascending order according
        to Status;

    }

    Else if (sortString==Follow_Up && OrderString ==DSC){

        Sort items in patient_join_follow_up in descending order
        according to Follow_Up;}

    Else if (sortString==Follow_Up && OrderString ==DSC){

        Sort items in patient_join_follow_up in descending order
        according to Follow_Up;}}}
```

Send patient_join_follow_up in ViewBag to the View;

Action Patient_Follow_Up():

Open Follow_Ups database table and search for a row with Id= Follow_Up_Id

If(search Follow_Up_Id)

{

 Return Patient_Id;

Opening Patients database table and search for a row with Id== Patient_Id;

If (searching Patient_Id)

{

 Return the entire row in variable (result);

 Return the entire row in variable (patient)

}

Include registrations table in Patients table to find the email of the patient whose Id==Patient_Id and return it in variable(Mail);

Return Status in variable (status);

Send variables mail, status and result in ViewBags to (Follow_Up)View;

Return the entire row (follow_Up)

}

If (Edit_status),(Edit Medical Record)or (Edit Health Progress) buttons are pressed

{a post action is sent from (Follow_UpVM) ViewModel to the controller to edit in database;}

```
If (Status sent from (Follow_UpVM) !=Null)
{
    Select Status in variable follow_Up and Status to input status;

}

If (Medical_Record sent from (Follow_UpVM) !=Null)
{
    Select Medical_record in variable patient and Status to input record;

}

If (Health_Progress sent from (Follow_UpVM) !=0)
{
    Select Health_Progress in variable patient and Status to input Health_Progress;
}
```

Action AnotherHospital:

Convert Transfer_Hospitals database table into a list and send it in ViewBag to
(AnotherHospital)View

Open Follow_Ups database table and search for a row with Id= Follow_Up_Id

If(search Follow_Up_Id)

{

Return Patient_Id;

Return the entire row in variable (follow)

Opening Patients database table and search for row with Id==Patient_Id

If (searching Patient_Id)

{

Return the entire row in variable (patient);

If (Hospital_Id = NULL)

{

Edit Hospital_Id to the new_hospital_Id;

Change follow_up.Status into "Transferred";

}

Else

{

Return;

}}}

Action AnotherDepartment():

Open Specializations database table and return names of specializations whose code starts with digit 1 in variable (OtherDep).

Open Follow_Ups database table and search for a row with Id= Follow_Up_Id

If(search Follow_Up_Id)

{

Return patient_Id;

Return the whole row in variable entity;

Select Status from entity to “Pending”

Select Staff_Id from entity and change it to new input Staff_Id;

Open Follow_Up_History database table and search for Follow_Up_Id

If (search Follow_Up_Id)

{

Change Date to “NULL”:

Change Follow_Up_Type to new input Follow_Up_Type;

}

}

Action Emergency():

Converting the data in Blood_Units database table to a list to send it in ViewBag to (Emergency) View;

Open Hospital_Facilities database table and search for (Type=="Emergency Bed")

If (search Emergency Bed)

{

Return the entire row (facility);

Send facility in ViewBag to (Emergency) View;

}

Action Schedule():

Open Staff database table and search for Registration_Id==Doc_Reg_Id

If(search Doc_Reg_Id)

{

Return Staff_Id;

Open Staff database table and search for a row with Id==Staff_Id

If(search Staff_Id)

{

Open Follow_Ups database table and search for Staff_Id

If(search Staff_Id)

{

Return all Ids of rows with Staff_Id in (followlist);

Open Follow_Ups_History database table and search for all Ids in followlist and send their date and Follow_up_Ids in ViewBag to

(Schedule) View;

}

}

}

Action Profile():

Open Staff database table and search for Registration_Id==Doc_Reg_Id

If(search Doc_Reg_Id)

{

Return Staff_Id;

Open Staff database table and search for a row with Id==Staff_Id

If(search Staff_Id)

{

Return Name ,Phone number and Starting_Day from that row (info);

Send info as ViewBag to (Profile) View;

Cuts the first name only from Name and send it ViewBag.firstName;

Return Specialization_Id from that row(y);

Open Specializations database table and search for a row with Id==y

If (search y)

{

Return Name in (x);

Send x as ViewBag to (Profile) View;

}

}

}

Open Registrations database table and search for Registration_Id==Doc_Reg_Id

If(search Doc_Reg_Id)

{

Send Email to (Profile) View in ViewBag.email;

}

If the doctor changed any of his data and pressed save changes button

{

Open Staff database table and search for Registration_Id==Doc_Reg_Id

If(search Doc_Reg_Id)

{

Return Doctor_Id;

Return the whole row in variable entity;

}

Open Registrations database table and return the email of the row whose Id==

Doc_Reg_Id in variable email;

Select Name from entity and change it to new input Name;

Select Phone from entity and change it to new input Phone;

Select Startding_Day from entity and change it to new input Startding_Day;

Change email to new input email;

}

VII. Profile controller:

In this controller “profile controller” showing the profile of patient

In Action UserProfile:

In this controller we call a specific table from Database “Patient Table” and

Get the patient Id and his Information To showing it in the profile view

```
return View(return patient information to view);
```

In Action Edit(int id =0)

By using session, we get the patient Registration Id;

By registration id to get the patient Id.

Here we show all the patient information using View bag if it existing

```
if (if information is null)
```

```
{
```

Will return to the patient did not find any information

```
return NotFound();
```

```
}
```

```
return View(pa).
```

```
}
```

VIII. Actions in Lab controller:

Action BloodBank():

Converts Blood_Units database table into a list and send it in ViewBag into BloodUnit View;

if Add button is pressed in a row in blood types table in BloodBank View

{

 Return the number of that row in variable (index);

 Open Blood_Units database table and return the row of Type called in index in variable (bloodunit);

 Select Amount from bloodunit and add the original value stored in it to the input amount sent from BloodBankVM;

}

If Delete button is pressed in a row in blood types table in BloodBank View

{

 Open Blood_Units database table and return the row of Type called in index in variable (bloodunit);

 Remove this row (bloodunit) from Blood_Units table in database;

}

If Add button is pressed in a row in blood types table in BloodBank View

{

 Create a new row in Blood_Unit database table and Blood_Unit.Type== input Type from Blood_BankVM and Amount == input amount from Blood_BankVM;

}

Action LabMain():

Open Staff database table and search for Registration_Id==Doc_Reg_Id
If(search Doc_Reg_Id)
{
 Return Staff_Id;

 Open Follow_Ups database table and return all patient_ids whose rows have
 Staff_Id== Staff_Id in variable (Patients_Follow_Ups);

 Create a list of patients called (patient);

 Create a list of Follow_Ups_History called (follow_up);

 Merge these 2 lists together In 1 list (patient_join_folow_up);

 Loop around all elements in Patients_Follow_Ups

 {
 Add to patient list the patient whose (Id==Patient_Id in
 Patients_Follow_Ups)

 Add to follow_up list the Follow_Up whose (Follow_Up Id == Id in
 Patients_Follow_Ups);

 Add these new elements in patient_join_follow_up list;

 }

 Return the value in SearchItem sent from (LabMainVM) in variable (SearchString)

 If (searchString !=Null)
 {

Select from the list patient_follow_up the columns from
(Name,Id,Follow_Up_Type_Name) the column which is == SearchString
and send it in ViewBag.follow_ups to the View;

}

Sort items in patient_join_follow_up in descending order according to Date;

Return the value in SortItem sent from (LabMainVM) in variable (SortString)

If (SortString!=Null)

{

If (OrderItem sent from post action ==Null)

{

Return “DSC” in variable (OrderString);

}

Else

{

Return the value of OrderItem sent from post action in variable
OrderString;

}

If (sortString==Date && OrderString==DSC)

{

Sort items in patient_join_follow_up in descending order
according to Date;

}

```
Else if (sortString==Date && OrderString ==ASC)

{

    Sort items in patient_join_follow_up in ascending order according
    to Date;

}

Else if (sortString==Status && OrderString ==DSC)

{

    Sort items in patient_join_follow_up in descending order
    according to Status;

}

Else if (sortString==Status && OrderString ==ASC)

{

    Sort items in patient_join_follow_up in ascending order according
    to Status;

}

Else if (sortString==Follow_Up && OrderString ==DSC)

{

    Sort items in patient_join_follow_up in descending order
    according to Follow_Up;

}

Else if (sortString==Follow_Up && OrderString ==DSC)

{
```

Sort items in patient_join_follow_up in descending order
according to Follow_Up;

}

}

}

Send patient_join_follow_up in ViewBag to the View;

Action LabPatient():

Open Follow_Ups database table and search for a row with Id= Follow_Up_Id
If(search Follow_Up_Id)
{
 Return Patient_Id;

 Opening Patients database table and search for a row with Id== Patient_Id;
 If (searching Patient_Id)
 {
 Return the entire row in variable (result)
 }

 Include registrations table in Patients table to find the email of the
 patient whose Id==Patient_Id and return it in variable(Mail);

 Return Status in variable (status);

 Return the entire row (follow_Up);

 Send variables mail, status and result in ViewBags to (Follow_Up)View;
}

If (Edit_status) button is pressed
{
 A post action is sent from (LabMainVM) ViewModel to the controller to edit in
 database;

 If(Status!="pending")

```
{  
    Select Status in variable follow_Up and Status to input status;  
}  
  
}
```

17.Testing:

Testing for Patient:

- 1) Sign up as patient.

The screenshot shows a web page for 'AIN SHAMS HOSPITAL'. At the top right is a blue button labeled '19056'. Below it is a navigation bar with links 'Home', 'About', and 'Contact'. The main content area has a heading 'Please enter your Activation Code' and a sub-instruction 'Hint:if you are a patient enter "0000"'. There is an input field containing '0000' and a blue 'Create' button below it. A 'Login' link is also present. At the bottom of the page is a copyright notice: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 85 Testing: Sign up as patient (Activation code)

First we enter patient activation code which is 0000 then click create

The screenshot shows a web page for 'AIN SHAMS HOSPITAL'. At the top right is a blue button labeled '19056'. Below it is a navigation bar with links 'Home', 'About', and 'Contact'. The main content area has a heading 'RegistrationPatient'. It contains several input fields: 'Name' (Nouran Ahmed), 'Phone' (01022299894), 'Email' (nouran@gmail.com), 'Password' (four dots), 'Confirm Password' (four dots). Below these is a blue 'Create' button. A 'Back' link is at the bottom left. At the bottom of the page is a copyright notice: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 86 Testing: Sign up as patient (Filling information)

This view appears ,the patient entered his information and click create.

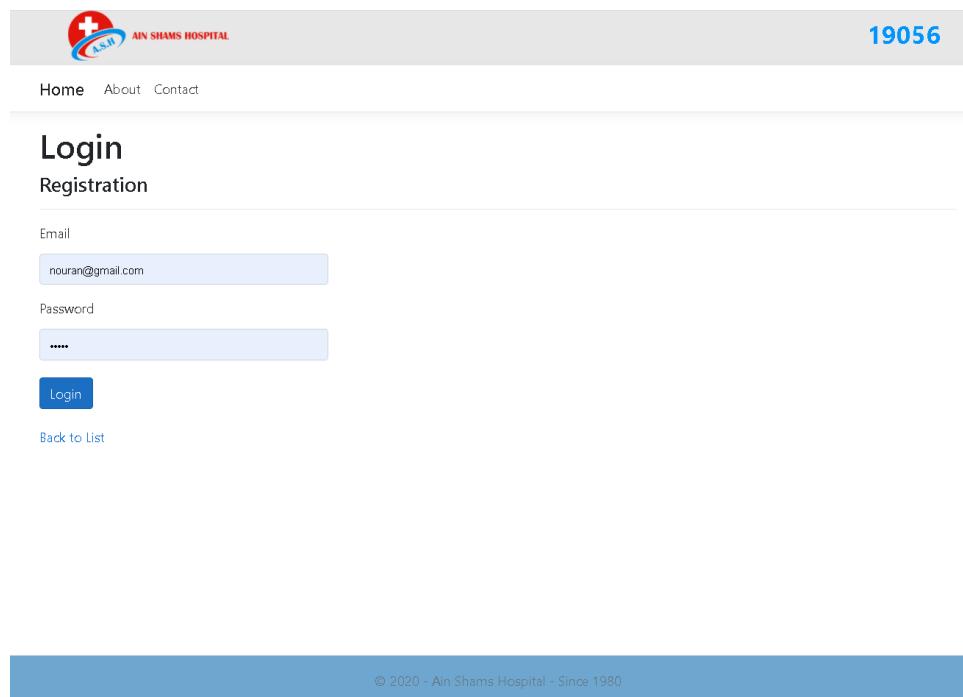


Figure 87 Testing: login as patient

He will be redirected to login with his information set in the fields then click login

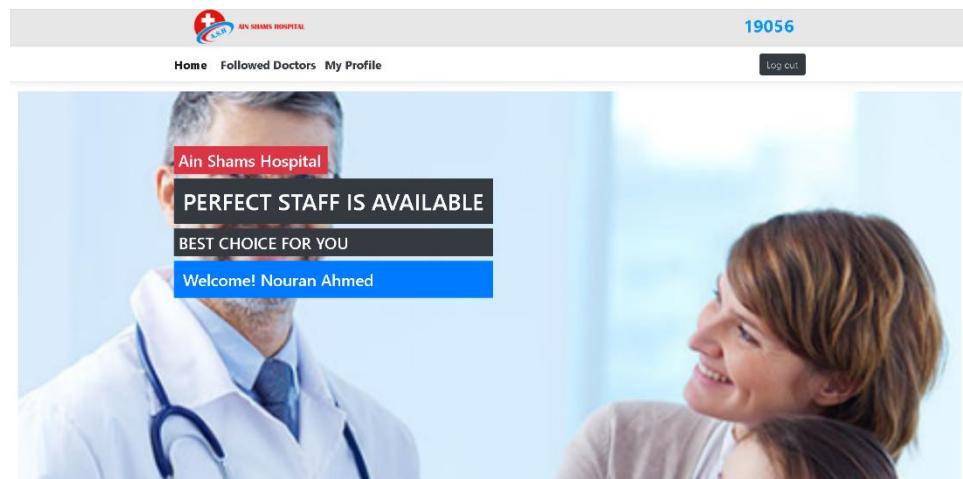


Figure 88 Testing: after patient login

He will be redirected to patient home view.

- 2) Enter wrong email format and password not matching during sign up.

The screenshot shows a registration form for 'RegistrationPatient'. The form fields are as follows:

- Name: Nouran Ahmed
- Phone: 01022299894
- Email: nouranahmed (with validation error: Please enter a valid email address)
- Password: (with validation error: 'ConfirmPassword' and 'Password' do not match.)
- ConfirmPassword: (highlighted in blue)

Below the form are two buttons: a blue 'Create' button and a blue 'Back' button.

Figure 89 Testing: Enter wrong email format and password not matching during sign up

- 3) Login with the wrong email or password.

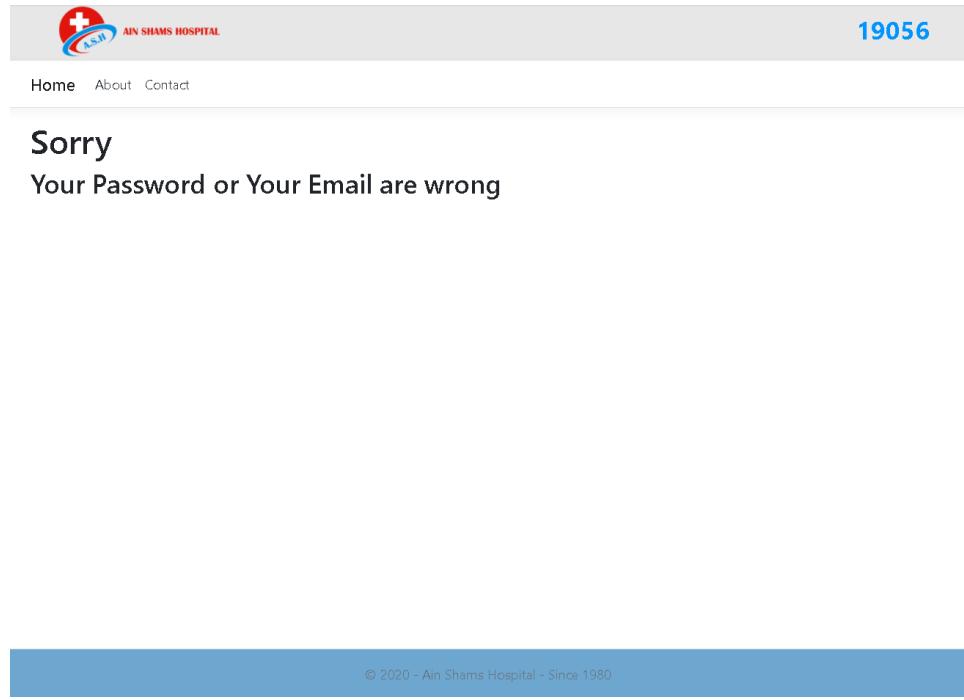


Figure 90 Testing: Login with the wrong email or password.

- 4) Add appointment with a certain doctor and see that the doctor is added in followed doctor list and choose to pay offline.



Figure 91 Testing: Sectors in our hospital

Choose doctor

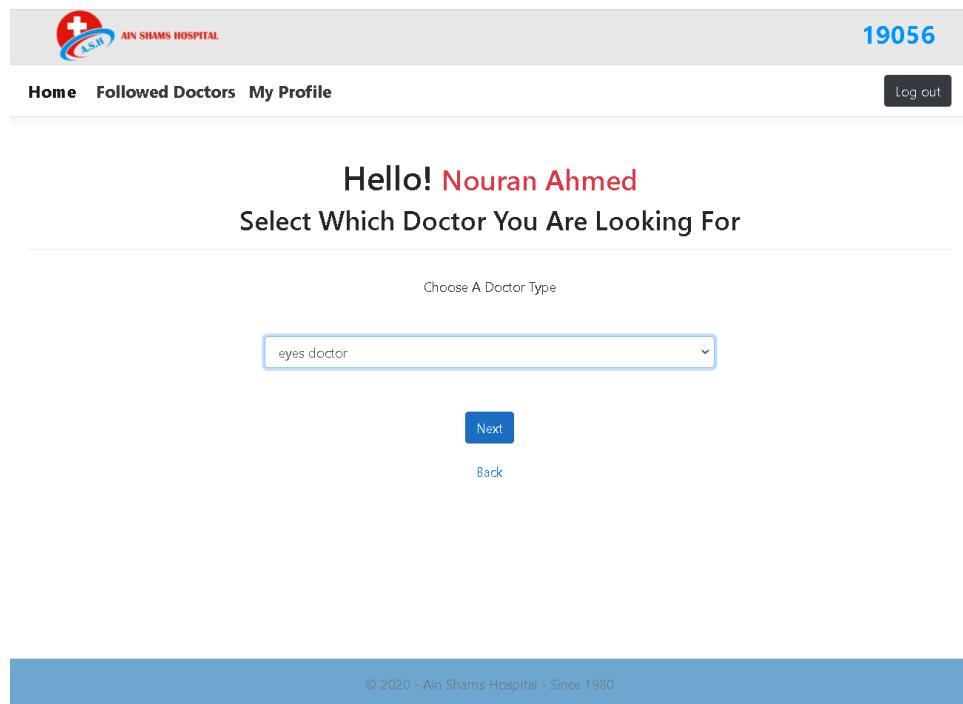


Figure 92 Testing: choose doctor

Select which specialization you are looking for then click next

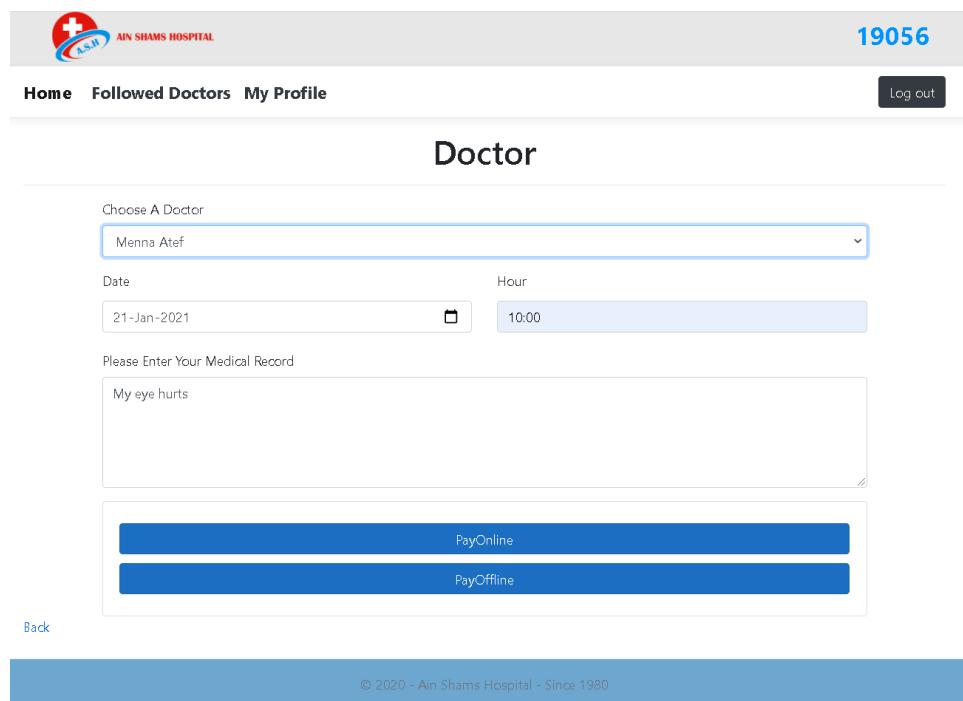


Figure 93 Testing: choose to pay online or offline

Select doctor Then Choose to pay Offline

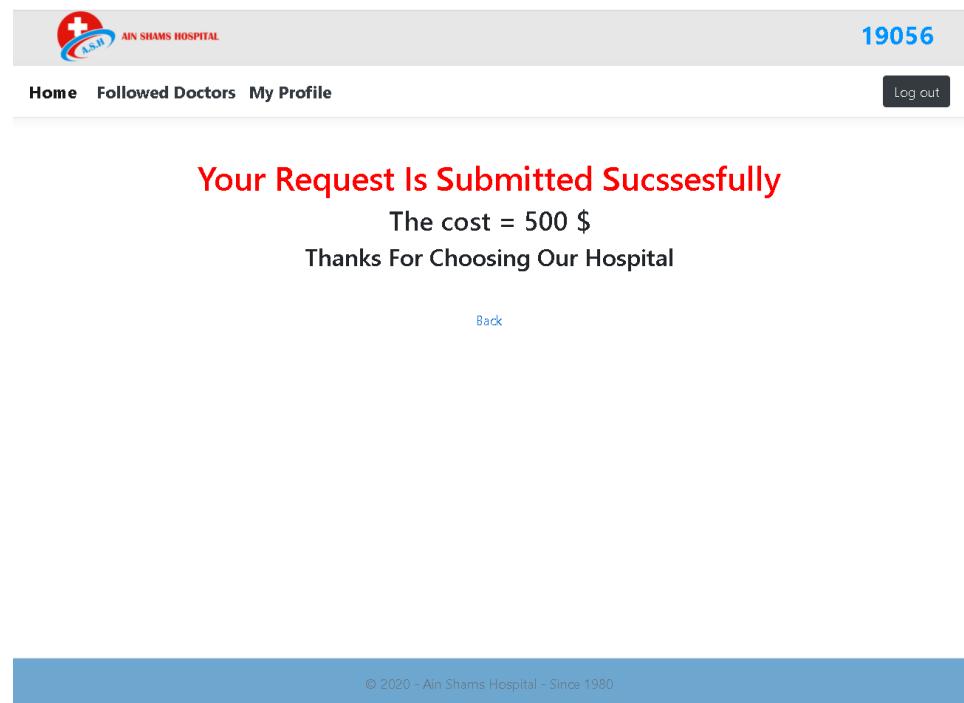


Figure 94 Testing: pay offline

Now your appointment is submitted and the examination costs 500\$

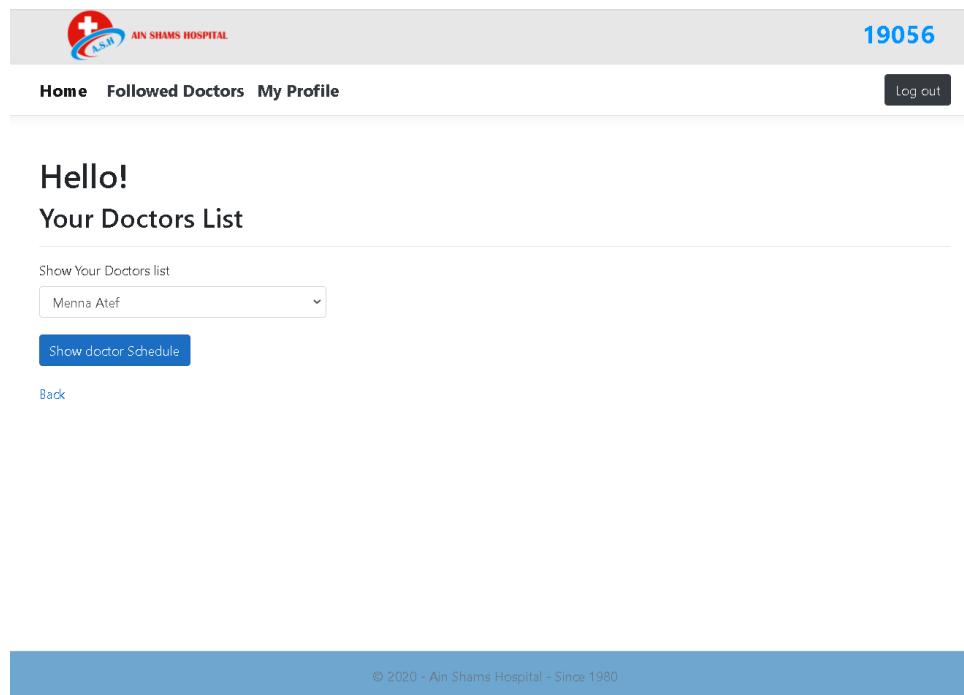


Figure 95 Testing: Doctor's list

Now your New Doctor is added to your doctor list

5) See Your Doctor's Schedule

The screenshot shows a mobile application interface for Ain Shams Hospital. At the top, there is a logo for 'AIN SHAMS HOSPITAL' with a red cross icon. To the right of the logo is the number '19056'. Below the header, there is a navigation bar with links for 'Home', 'Followed Doctors', and 'My Profile'. On the far right of the navigation bar is a 'Log out' button. The main content area starts with a greeting 'Hello! Nouran Ahmed' followed by 'Menna Atef's Schedule'. Below this, there is a table showing working days and their corresponding durations:

Working Day	Duration
Sunday	9AM-4PM
Monday	9AM-4PM
Wednesday	9AM-4PM
Thursday	9AM-4PM

On the right side of the schedule table is a large, dark circular placeholder for a profile picture. Below the table is a small link 'Back'. At the bottom of the screen, there is a blue footer bar with the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 96 Testing: Doctor's Schedule

6) choose another doctor in another specialization and choose to pay online

The screenshot shows a mobile application interface for Ain Shams Hospital. At the top, there is a logo for 'AIN SHAMS HOSPITAL' with a red cross icon. To the right of the logo is the number '19056'. Below the header, there is a navigation bar with links for 'Home', 'Followed Doctors', and 'My Profile'. On the far right of the navigation bar is a 'Log out' button. The main content area starts with the word 'Doctor' in bold. Below it, there is a section titled 'Choose A Doctor' with a dropdown menu containing the name 'Heba Atef Muhammed'. Underneath this, there are fields for 'Date' (set to '21-Jan-2021') and 'Hour' (set to '14:00'). Below these fields is a text input field labeled 'Please Enter Your Medical Record' containing the text 'My Heart is not working properly'. At the bottom of the screen, there are two blue buttons: 'PayOnline' and 'PayOffline'. Below the buttons is a small link 'Back'. At the bottom of the screen, there is a blue footer bar with the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 97 Testing: select doctor and pay online

Choose to pay online

The screenshot shows a web-based payment interface for Ain Shams Hospital. At the top right, the number '19056' is displayed. Below it, there are navigation links: 'Home', 'Followed Doctors', 'My Profile', and a 'Log out' button. The main area is titled 'Pay Invoice'. It includes fields for 'Payment amount' (\$500), 'Card Number' (1333333333333333), 'Name on card' (Nouran Ahmed), 'Expiry Date' (27-Jan-2021), 'Security Code' (222), and 'ZIP/Postal code' (26226). A large green button at the bottom left of the form area contains the text '\$500'.

Figure 98 Testing: online payment

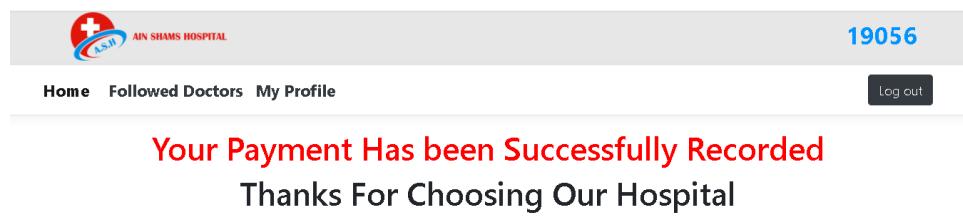


Figure 99 Testing: online payment successfully recorded

7) Patient adds appointment in lab.



19056

[Home](#) [Followed Doctors](#) [My Profile](#) [Log out](#)

Hello! Nouran Ahmed
Select Which Lab You Are Looking For

Choose A Lab Type

Blood Bank Specialist

[Next](#) [Back](#)

© 2020 - Ain Shams Hospital - Since 1980

Figure 100 Testing: choose lab



19056

[Home](#) [Followed Doctors](#) [My Profile](#) [Log out](#)

Lab

Choose Your Test

Liver Tests

Date Hour

27-Jan-2021 11:00

Please Enter Your Medical Record

I Take Panadol

[PayOnline](#) [PayOffline](#)

[Back](#)

© 2020 - Ain Shams Hospital - Since 1980

Figure 101 Testing: Filling lab scan request

8) View Patient Profile and edit it then see the changes

The screenshot shows the 'My Profile' section of the AIN SHAMS HOSPITAL website. At the top, there is a header with the hospital's logo and name, followed by the number '19056'. Below the header, there are navigation links for 'Home', 'Followed Doctors', and 'My Profile'. On the right, there is a 'Log out' button. The main content area is titled 'My Profile' in red. It displays the following information:

- Name:-** Nouran Ahmed
- Phone:-** 01022299894
- Health_Progress:-** 55
- Medical_Record:-** My Heart is not working properly

On the right side, there is a placeholder for a user profile picture, showing a black silhouette of a person inside a circle. Below the profile area, there are two small blue links: 'Edit& Details' and 'Back to List'. At the bottom of the page, a blue footer bar contains the text '© 2020 - AIN SHAMS HOSPITAL - Since 1980'.

Figure 102 Testing: Patient Profile

The screenshot shows the 'Edit Your Profile' page of the AIN SHAMS HOSPITAL website. At the top, there is a header with the hospital's logo and name, followed by the number '19056'. Below the header, there are navigation links for 'Home', 'About', and 'Contact'. On the right, there is a 'Log out' button. The main content area is titled 'Edit Your Profile' in blue. It contains input fields for the following information:

- Name**: Nour
- Phone**: 01022299894
- Medical_Record**: I'm Not Fine

Below the input fields, there is a section titled 'Your Progress' with a progress bar consisting of a blue striped bar and a grey bar. Underneath the progress bar is a section titled 'Your Current Hospital' with the name 'Al Monira'. At the bottom of the page, there is a blue 'SaveChanges' button and a 'Back' link.

Figure 103 Testing: edit patient profile

 AIN SHAMS HOSPITAL

19056

Home Followed Doctors My Profile Log out

My Profile

Name:-
Nour

Phone:-
01022299894

Health_Progress:-
55

Medical_Record:-
I'm Not Fine

[Edit& Details](#) | [Back to List](#)

© 2020 - Ain Shams Hospital - Since 1980

Figure 104 Testing: patient profile

Testing For Doctor:

- 1) Patient adds appointment with doctor and check that the appointment added in the doctor schedule.

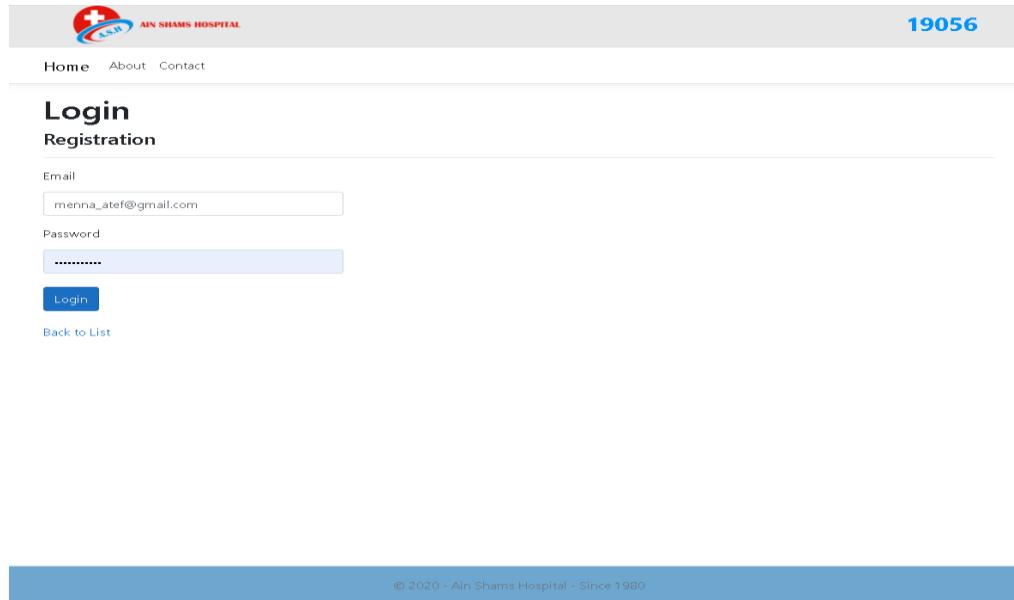


Figure 105 Testing: doctor login

The screenshot shows the doctor's home page. At the top right is a red button labeled '19056'. The header includes the hospital's logo and name, along with 'Log out' and 'Emergency' buttons. Below the header is a search bar with 'Search:' and 'Go' buttons, and dropdown menus for 'Sort By:' and 'Order:' with a 'Sort' button. The main area displays six patient appointment cards in a grid:

Patient No.	Name	Follow Up	Status	Date	Notes
Patient No. 13	Manal Muhammed	Examination	Pending	2021-01-30T09:00	
Patient No. 16	Nour	Examination	Pending	2021-01-21T10:00	Transferred to: Al Monira
Patient No. 2	Atef Ahmed	Examination	Pending	2021-01-17T12:00	
Patient No. 11	Khaled Karim	Examination	Pending	2021-01-14T16:00	
Patient No. 1	Ahmed Nour	Examination	Pending		Recently Transferred to You

At the bottom is a footer with the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 106 Testing: doctor home page

The patient is added to the doctor's schedule

- 2) Doctor checks the medical record and update information of another patient.

The screenshot shows a web-based medical application interface. At the top right, the number '19056' is displayed. On the left, there's a logo for 'AN SHAMS HOSPITAL' with a red cross symbol. The top navigation bar includes links for 'Home', 'My Schedules', 'Profile', 'Log out', and 'Emergency'. The main content area is titled 'Patient personal informations' in green. It displays the following data:

Name: Atef Ahmed	
Phone: 15598885641	Email: atef_ahmed@gmail.com
Health Progress: 0	Status: Pending

Below this, under 'Patient Medical_Record', it says 'Nothing'. There are three input fields for updating medical records:

- 'Add Medical_Record': A text input field containing the placeholder 'you are fine you can leave' with a blue 'Edit Medical Record' button below it.
- 'Add Health_Progress': A progress bar set to 00, with a blue 'Edit Health Progress' button below it.
- 'Add Status': A dropdown menu set to 'In Progress' with a blue 'Edit Status' button next to it.

At the bottom, a white box contains the text 'If you need to trasfer patient.' followed by a large black 'Transfer' button.

Figure 107 Testing: Updating patient's medical data

Patient personal informations

Name: Atef Ahmed

Phone: 15598885641 Email: atef_ahmed@gmail.com

Health Progress: 80 Status: In Progress

Patient Medical_Record

Nothing you are fine you can leave

Add Medical_Record:

Add Health_Progress:

80

Add Status: Status: Edit Status

If you need to trasfer patient.

Transfer

Figure 108 Testing: After updating health progress & status

3) Doctor transfer patient to another doctor/hospital and see the result in patient's page.

Transfer this patient to:

Another department

Another hospital

Figure 109 Testing: Transferring patient

The screenshot shows a web-based application for managing patient transfers. At the top right, the number '19056' is displayed. On the left, the hospital's logo and name 'AIN SHAMS HOSPITAL' are visible. Below the logo, there are navigation links: 'Home', 'My Schedules', and 'Profile'. On the right side, there are two buttons: 'Log out' and 'Emergency'. The main content area is titled 'Transfer this patient to:' and contains several input fields:

- 'Department of:' dropdown menu set to 'cardiologist'
- 'Doctor:' dropdown menu set to 'Heba Atef Muhammed'
- 'Follow Up Type:' dropdown menu set to 'Corona Tests'

A blue 'submit' button is located at the bottom of this form.

Figure 110 Testing: Transferring patient to another doctor

This screenshot shows the 'Followed Doctors' section of the application. At the top right, the number '19056' is shown. On the left, the hospital's logo and name 'AIN SHAMS HOSPITAL' are present. Below the logo, there are navigation links: 'Home', 'Followed Doctors', and 'My Profile'. On the right, there is a 'Log out' button. The main content area starts with a greeting 'Hello!' and the heading 'Your Doctors List'. It includes a dropdown menu labeled 'Show Your Doctors list' which is currently set to 'Heba Atef Muhammed'. Below this is a blue 'Show doctor Schedule' button. A small 'Back' link is also visible. At the bottom of the page, a blue footer bar contains the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 111 Testing: The doctor whom the patient is transferred to is shown in the followed doctor list of the patient

4)show Doctor Schedule

The screenshot shows a web interface for 'AIN SHAMS HOSPITAL'. At the top right is the number '19056'. Below it are navigation links: 'Home', 'My Schedules', and 'Profile'. On the far right are 'Log out' and 'Emergency' buttons. The main title 'My schedule' is centered above a table. The table has two columns: 'Date' and 'Follow Up'. It lists two entries: '2021-01-13T12:00' under 'Follow Up' and 'Examination' under 'Follow Up'. Another entry '2021-01-21T10:00' is listed under 'Follow Up'.

Date	Follow Up
2021-01-13T12:00	Examination
2021-01-21T10:00	Examination

Figure 112 Testing: Show doctor schedule

5) show Doctor Profile

The screenshot shows a 'Profile' page for a doctor named 'DR. Heba'. The page includes a greeting 'Hi DR. Heba' and several input fields: 'Full name' (Heba Atef Muhammed), 'Phone number' (01003393030), 'You work in our hospital since' (2020-12-27), 'Email' (heba.atef.muhammed@gmail.com), and 'Your specialization is: cardiologist'. A note at the bottom says 'If you want to save any changes to your personal data click on save changes' with a 'Save changes' button.

Full name:
Heba Atef Muhammed

Phone number:
01003393030

You work in our hospital since:
2020-12-27

Email:
heba.atef.muhammed@gmail.com

Your specialization is: cardiologist

If you want to save any changes to your personal data click on save changes

Save changes

Figure 113 Testing: show Doctor Profile

6) View Emergency Beds and blood units

The screenshot shows a dashboard for Ain Shams Hospital. At the top right is the number '19056'. Below it are navigation links: 'Home', 'My Schedules', 'Profile', 'Log out', and a red button labeled 'Emergency'. The main area has two tables. The left table, titled 'Blood_Unit', lists blood types and their amounts: A+ (28), A- (25), B+ (32), B- (32), O- (40), AB+ (18), AB- (43), O+ (10), and A+ (2). The right table, titled 'Emergency bed', lists four beds: Emergency bed 1 (Available: True, Start_Working: All, End_Working: All), Emergency bed 2 (Available: True, Start_Working: All, End_Working: All), Emergency bed 3 (Available: True, Start_Working: All, End_Working: All), and Emergency bed 4 (Available: True, Start_Working: All, End_Working: All).

Figure 114 Testing: View Emergency beds and blood units

Testing For Lab:

1) Register as a lab specialist and change patient status

The screenshot shows a login page for Ain Shams Hospital. At the top right is the number '19056'. Below it are navigation links: 'Home', 'About', and 'Contact'. The main title is 'Login' with a 'Registration' link below it. There are fields for 'Email' (containing 'lab@gmail.com') and 'Password' (containing '*****'). A blue 'Login' button is at the bottom left, and a 'Back to List' link is at the bottom right. A footer bar at the bottom contains the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 115 Testing: login as a lab specialist

The screenshot shows a web interface for Ain Shams Hospital. At the top right is the number '19056'. On the left is the hospital's logo and name. Below the logo are links for 'Home', 'About', and 'Contact'. On the right are 'Log out' and search functions. A search bar with a 'Go' button is followed by dropdown menus for 'Sort By:' and 'Order:', and a 'Sort' button. Two patient cards are displayed: 'Patient No. 2' (Name: Atef Ahmed, Follow Up: Corona Tests, Date: 2021-01-26T13:00, Status: Pending) and 'Patient No. 1' (Name: Ahmed Nour, Follow Up: Parasitology, Date: 2021-01-19T11:00, Status: Pending). At the bottom is a copyright notice: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 116 Testing: View lab requests

The screenshot shows a web interface for Ain Shams Hospital. At the top right is the number '19056'. On the left is the hospital's logo and name. Below the logo are links for 'Home', 'About', and 'Contact'. On the right are 'Log out' and other buttons. The main content area has a green header 'Patient personal informations'. Below it are four lines of information: 'Name: Atef Ahmed', 'Phone: 15598885641', 'Email: atef_ahmed@gmail.com', and 'Status: Pending'. Below this is a form with 'Add Status:' and a dropdown menu set to 'Pending', along with a 'edit Status' button. At the bottom is a copyright notice: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 117 Testing: Updating lab request status part1

The screenshot shows a web application interface for Ain Shams Hospital. At the top right, the number "19056" is displayed. The header includes the hospital's logo and name "AIN SHAMS HOSPITAL". Navigation links for "Home", "About", and "Contact" are on the left, and a "Log out" button is on the right. The main content area is titled "Patient personal informations" in green. It lists the following details for a patient:

- Name: Atef Ahmed
- Phone: 15598885641
- Email: atef_ahmed@gmail.com
- Status: In Progress

Below this, there is a "Add Status:" field with a dropdown menu set to "In Progress" and a blue "edit Status" button.

© 2020 - Ain Shams Hospital - Since 1980

Figure 118 Testing: Updating lab request status part2

2)Sort requests by date ascending

The screenshot shows the same web application interface as Figure 118, but with sorting applied. The "Date" dropdown is set to "Ascending" and the "Sort" button is highlighted in green. The results show two entries:

Patient No.	Name	Follow Up	Date	Status
Patient No. 1	Ahmed Nour	Parasitology	2021-01-19T11:00	Pending
Patient No. 2	Atef Ahmed	Corona Tests	2021-01-26T13:00	In Progress

Figure 119 Testing: Sort requests by date ascending

3) Search for a certain request

The screenshot shows a web interface for AIN SHAMS HOSPITAL. At the top right, the number '19056' is displayed. Below it, there are links for 'Home', 'About', and 'Contact', and a 'Log out' button. A search bar contains the text 'Search: Ahmed Nour' with a green 'Go' button next to it. To the right of the search bar are dropdown menus for 'Date' and 'Ascending' status, and a green 'Sort' button. Two search results are shown in boxes:

Patient No.	Name	Follow Up	Date	Status
2	Atef Ahmed	Corona Tests	2021-01-26T13:00	In Progress
1	Ahmed Nour	Parasitology	2021-01-19T11:00	Pending

Figure 120 Testing: Search for a certain request part1

Write his name and click GO

The screenshot shows a web interface for AIN SHAMS HOSPITAL. At the top right, the number '19056' is displayed. Below it, there are links for 'Home', 'About', and 'Contact', and a 'Log out' button. A search bar contains the text 'Search:' with a green 'Go' button next to it. To the right of the search bar are dropdown menus for 'Sort By' and 'Order', and a green 'Sort' button. One search result is shown in a box:

Patient No.	Name	Follow Up	Date	Status
1	Ahmed Nour	Parasitology	2021-01-19T11:00	Pending

Figure 121 Testing: Search for a certain request part2

Testing For blood bank:

1) add new blood units

The screenshot shows a web-based application for managing blood units. At the top right, the number '19056' is displayed. Below it is a navigation bar with links for 'Home', 'About', 'Contact', and 'Log out'. The main title 'Blood_Unit' is centered above a table. The table has columns for 'Type', 'Amount', 'Edit', and 'Delete'. The data rows are as follows:

Type	Amount	Edit	Delete
A+	28	<input type="text" value="0"/>	
A-	25	<input type="text" value="0"/>	
B+	32	<input type="text" value="0"/>	
B-	32	<input type="text" value="0"/>	
O-	40	<input type="text" value="0"/>	
AB+	18	<input type="text" value="0"/>	
AB-	43	<input type="text" value="0"/>	
O+	10	<input type="text" value="0"/>	

Below the table are input fields for 'Type' (A+) and 'Amount' (2), followed by a green 'Add' button. A copyright notice at the bottom states: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 122 Testing: add new blood units part1

This screenshot shows the same application interface as Figure 122, but with a new row added to the table. The new row represents an additional unit of type A+ with an amount of 2. The rest of the data remains the same as in Figure 122. The 'Add Another Blood Unit' button is visible at the bottom of the table.

Type	Amount	Edit	Delete
A+	28	<input type="text" value="0"/>	
A-	25	<input type="text" value="0"/>	
B+	32	<input type="text" value="0"/>	
B-	32	<input type="text" value="0"/>	
O-	40	<input type="text" value="0"/>	
AB+	18	<input type="text" value="0"/>	
AB-	43	<input type="text" value="0"/>	
O+	10	<input type="text" value="0"/>	
A+	2	<input type="text" value="0"/>	

At the bottom, there is a green 'Add Another Blood Unit' button and a copyright notice: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 123 Testing: add new blood units part2

Testing For Finance:

1) Show the bill of the patient in finance

PatientName
Atef Ahmed

Todaydate
20-Jan-2021

Show Bill

Type	Price
Examination	500
Corona Tests	400
Total=900	

Update Bill

© 2020 - Ain Shams Hospital - Since 1980

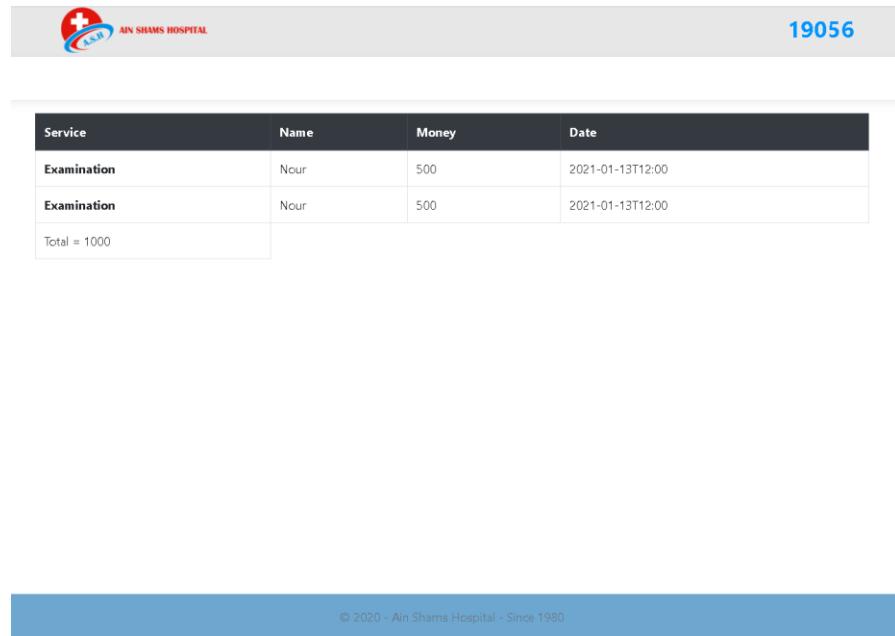
Figure 124 Testing: Show the bill of the patient in finance

2) See the total income for the hospital during a certain period.

Service	Name	Money	Date
Surgery	nermeen	2600	2021-01-02
Room	nermeen	600	2021-01-15
Room	nermeen	600	2021-01-11
Surgery	nermeen	2600	2021-01-13
Room	nermeen	200	2021-01-13
Room	nora	800	2021-01-13
Room	nora	600	2021-01-13
Room	nermeen	2600	2021-01-13

Figure 125: Total income

3) Finance collects the income for the hospital during a certain period for examination service



The screenshot shows a table titled "Finance" with the identifier "19056" at the top right. The table has four columns: "Service", "Name", "Money", and "Date". It lists two entries for "Examination" services provided by patient "Nour" on "2021-01-13T12:00" for a total amount of "500". A footer note states "Total = 1000".

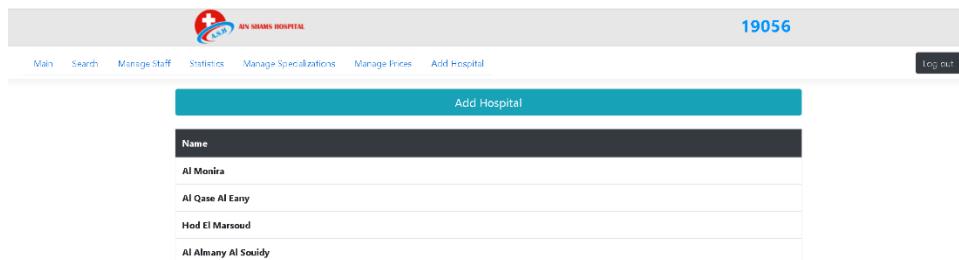
Service	Name	Money	Date
Examination	Nour	500	2021-01-13T12:00
Examination	Nour	500	2021-01-13T12:00
Total = 1000			

© 2020 - Ain Shams Hospital - Since 1980

Figure 126 Testing: Finance collects the outcome for the hospital during a certain period

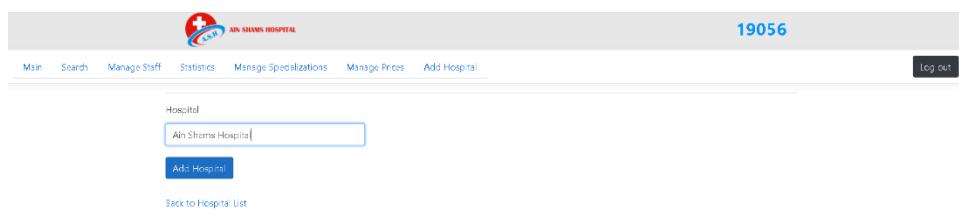
Testing Manager:

1) The manager added hospital



The screenshot shows a list of hospitals under the "Add Hospital" section. The list includes "Al Monira", "Al Qase Al Eany", "Hod El Marsoud", and "Al Almany Al Soudy".

Figure 127 Testing: The manager view transfer hospital



The screenshot shows a form for adding a new hospital. The "Hospital" field contains "Ain Shams Hospital" and the "Add Hospital" button is visible below it.

Figure 128 Testing: The manager adds transfer hospital part1

The screenshot shows a web-based application for managing hospital data. At the top right, the number '19056' is displayed. The header includes the 'AIN SHAMS HOSPITAL' logo and navigation links: Main, Search, Manage Staff, Statistics, Manage Specializations, Manage Prices, Add Hospital, and Log out. Below the header, a form titled 'Hospital' contains a text input field with the value 'Air Shams Hospital'. A blue button labeled 'Add Hospital' is present. A success message 'Hospital is added successfully' is shown above a link 'Back to Hospital List'.

Figure 129 Testing: The manager adds transfer hospital part2

2)The manager added Specialization and see that new specialization is added in Specialization table

The screenshot shows a web-based application for managing hospital data. At the top right, the number '19056' is displayed. The header includes the 'AIN SHAMS HOSPITAL' logo and navigation links: Main, Search, Manage Staff, Statistics, Manage Specializations, Manage Prices, Add Hospital, and Log out. Below the header, a form titled 'Specialization' contains two text input fields: 'Specialization' and 'Code'. A blue button labeled 'Add Specialization' is present. A link 'Back to Specialization List' is visible at the bottom.

Figure 130Testing : Add Specialization

3)The manager added service and see that that new service is added in services table

The screenshot shows a web-based application for managing hospital data. At the top right, the number '19056' is displayed. The header includes the 'AIN SHAMS HOSPITAL' logo and navigation links: Main, Search, Manage Staff, Statistics, Manage Specializations, Manage Prices, Add Hospital, and Log out. Below the header, a form titled 'ServiceName' contains two text input fields: 'ServiceName' and 'ServicePrice'. A blue button labeled 'Add Service' is present. A link 'Back to Service List' is visible at the bottom.

Figure 131 Testing :Add service

4)The manager delete Specialization and see that specialization is removed from the Specialization table

Main Search Manage Staff Statistics Manage Specializations Manage Prices Add Hospital Log out

Specialization

Delete Specialization

[Back to Specialization List](#)

19056

Figure 132: Delete specialization

5)The manager delete service and see that service is removed from the services table

Main Search Manage Staff Statistics Manage Specializations Manage Prices Add Hospital Log out

ServiceName

Delete Service

[Back to Service List](#)

19056

Figure 133:Delete Service

6)The manager edit service price and see that that the price of the service is edited in services table

Main Search Manage Staff Statistics Manage Specializations Manage Prices Add Hospital Log out

Name

Price

Edit Price

[Back to List of Prices](#)

19056

Figure 134: Edit Price

- 1) See the total income for the hospital during a certain period.

The screenshot shows a web-based hospital management system. At the top, there's a header with the AIN SHAMS HOSPITAL logo and a red cross symbol. To the right of the logo is the number '19056'. Below the header, a navigation bar includes links for Main, Search, Manage Staff, Statistics, Manage Specializations, Manage Prices, and Add Hospital. On the far right of the navigation bar is a 'Log out' button. The main content area displays a table titled 'Total Income' with the following data:

Service	Name	Money	Date
Surgery	nermeen	2600	2021-01-02
Room	nermeen	600	2021-01-15
Room	nermeen	600	2021-01-11
Surgery	nermeen	2600	2021-01-13
Room	nermeen	200	2021-01-13
Room	nora	800	2021-01-13
Room	nora	600	2021-01-13
Room	nermeen	2600	2021-01-13

Figure 135 Testing: Total income

8) show the income for each service during a certain period.

This screenshot shows the same hospital management system interface. The table data is as follows:

Service	Name	Money	Date
Examination	Nour	500	2021-01-13T12:00
Examination	Nour	500	2021-01-13T12:00
Total = 1000			

© 2020 - Ain Shams Hospital - Since 1980

Figure 136 Testing :Income for Examination Service

9)The manager search for a staff member and see it's information

Name

Search

Staff Personal Information

Name:

Phone:

Starting Day:

Specialization:

Email:

Figure 137 Testing: Staff Information

Testing Front desk:

1) Login as a front desk and reserve a room for a patient

AIN SHAMS HOSPITAL

19056

Home About Contact

Login

Registration

Email

frontdesk@gmail.com

Password

.....

Login

Back to List

Figure 138 Testing: Login as a front desk

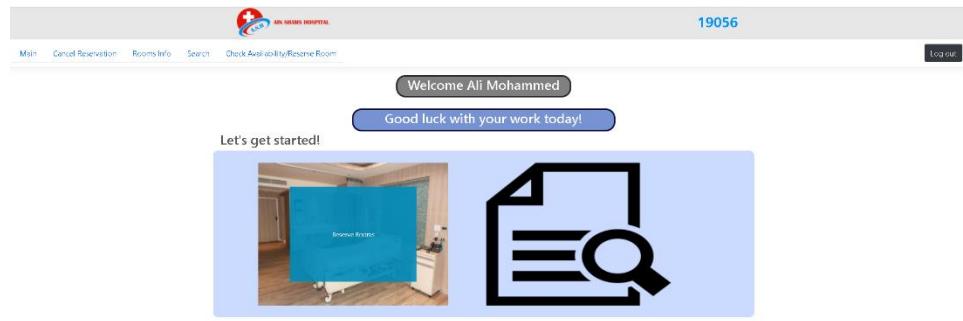


Figure 139 Testing: front desk home page

2)Choose reserve room

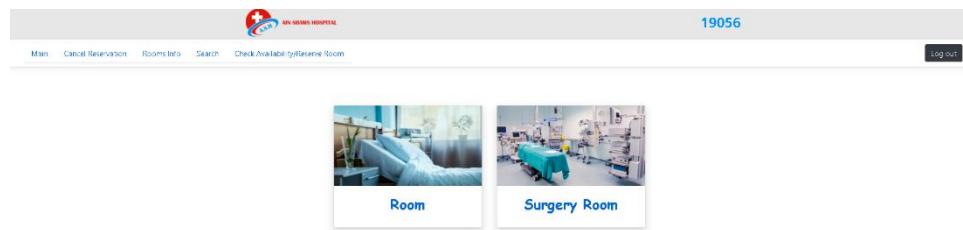


Figure 140 Testing: Choose reserve room part1

Choose room



Figure 141 Testing: Choose reserve room part2

Show available rooms then reserve room

Figure 142 Testing: reserve room part1

3)Enter patient and doctor name to reserve room

Figure 143 Testing: reserve room part2

4)Transfer patient from one room to another if in the hospital any available room

Figure 144 Testing :Transfer Patient

5) search for patient room number if he is in the hospital

- Enter Patient name
- Enter Today date

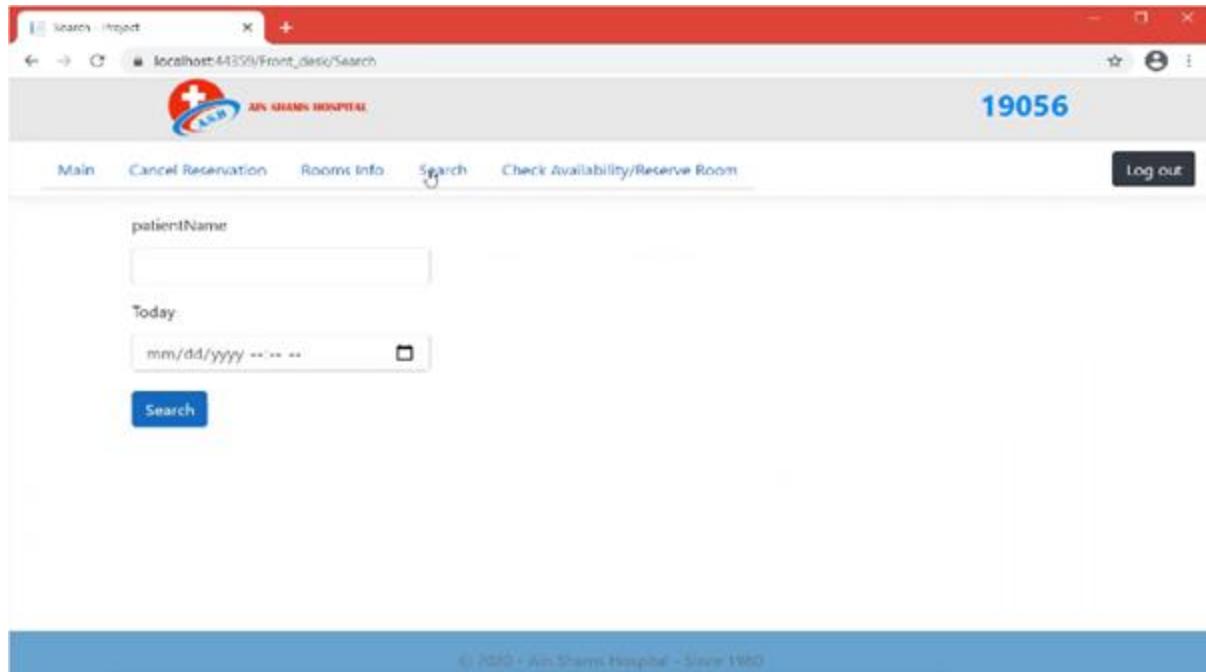


Figure 145 Testing: search for patient

6)cancel surgery reservation

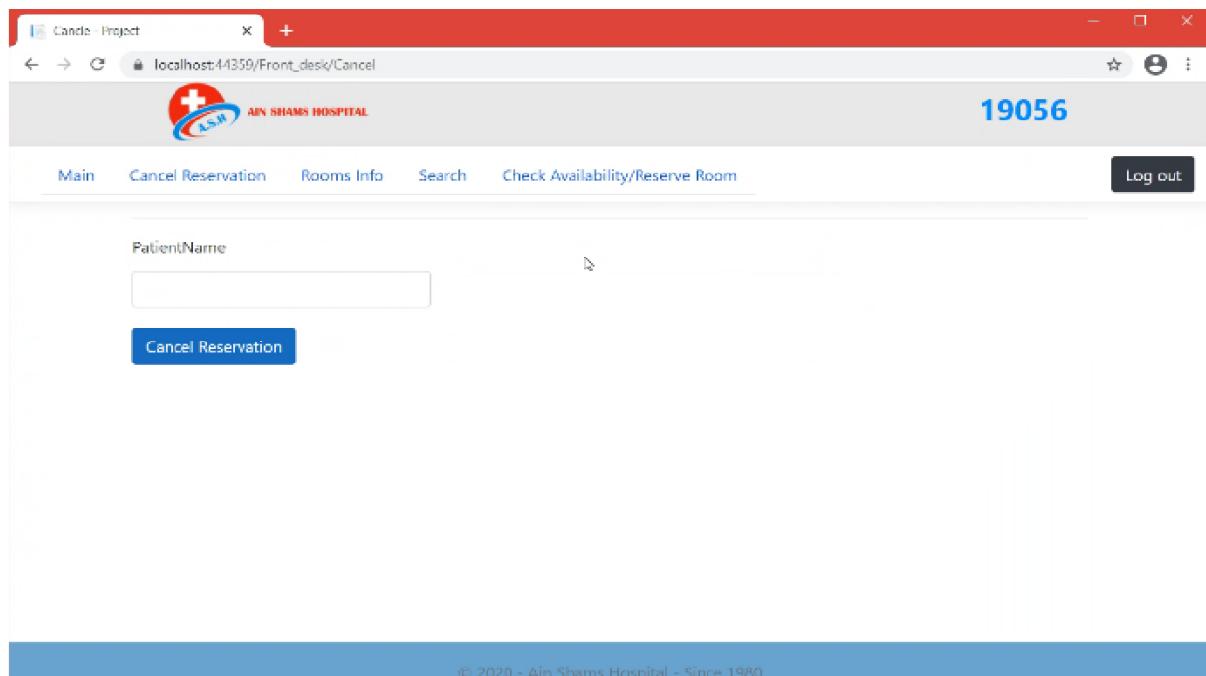


Figure 146 Testing : Cancel

18.Estimated cost:

Function Point Table:

Table 2 Function Point

measurement parameter	Simple	avg	complex
number of user inputs	3	4	6
number of user outputs	4	5	7
number of user inquiries	7	10	15
number of files	5	7	10
number of ext.interfaces	3	4	6

COCOMO-II Model:

$$PM = A \times Size^B \times M$$

Where :

$$M = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED$$

A = 2.94 in initial calibration, Size in Kilo Line Of Code

(KLOC), B varies from 1.1 to 1.24 depending on novelty of
the project, development flexibility, risk management
approaches and the process maturity.

Function Point Estimation (FP->KLOC) table:

Table 3 Function Point Estimation

Name	External user types	Complexity	FP
Doctor	External inquiry type	avg	5
Patient	External inquiry type	avg	5
Front desk	External inquiry type	complex	12
Home	External output type	simple	4
Registration	External input type	complex	6
Manager	External inquiry type	simple	7
Finance	External inquiry type	simple	7
Donation	External input type	simple	3
labs	External inquiry type	avg	5
total			54

FP->LOC:

- 3 Total function points =54
- 4 Published figures for C# show that: – 1 FP = 128 LOC in C#
- 5 Estimated Size – $54 * 128 = 6912 = 6.912$ KLOC

Scale Factor Estimation table:

Table 4 Scale Factor Estimation

Name	Very low (0.05)	Low (0.04)	Nominal (0.03)	High (0.02)	Very High (0.01)	Extra High (0.00)	Assessment	Value
Precedented	Thoroughly unprecede nted	Largely unprecedent ed	Somewhat unpreceden ted	General ly familiar	Largely familiar	Thoroughl y familiar	Very high	0.01
Flexibility	Rigorous	Occasional relaxation	Some relaxation	General conformity	Some conformi ty	General goals	Very high	0.01
Significant risks eliminated	Little (20%)						Nominal	0.03
Team interaction process	Very difficult	Some difficult	Basically cooperative	Largely cooperat ive	Highly cooper ative	Seamless interactio ns	High	0.02
Process maturity	Level 1	Level 2	Level 2+	Level 3	Level 4	Level 5	Low	0.04
Add								1.01
Total								1.13

Effort Adjustment Factors (EAF) table:

Table 5 Effort Adjustment Factors

Identifier	Name	Ranges (VL-EH)	Assessment	Values
			VL/L/N/H/VH/EH	
RCPX	product Reliability and Complexity	0.5-1.5	Low	0.75
RUSE	required reusability	0.5-1.5	Nominal	1.0
PDIF	Platform Difficulty	0.5-1.5	High	1.1
PERS	Personnel capability	1.5-0.5	High	0.75
PREX	Personnel Experience	1.5-0.5	very high	0.65
FCIL	Facilities available	1.5-0.5	Nominal	1.0
SCED	Schedule pressure	1.5-0.5	low	1.2
Product				0.4826

$$\text{Effort} = 2.94 \times (6.912)^{1.13} \times 0.4826 = 12.698 \text{ person-months} == 13.$$

- Total cost = 50000 LE.
- Duration of the project = 3 months.
- Salary of one person -months =1000 LE.
- Total salaries = $13 \times 1000 \times 3 = 39000$ LE.
- Other cost = $50000 - 39000 = 11000$ LE.

Expert Judgement Estimation:

- a) Advantages: Relatively cheap estimation method. Can be accurate if experts have direct experience of similar systems.
- b) Disadvantages: Very inaccurate if there are no experts!
- c) The expert Judgement the total cost of the project is 70000 LE.
- d) Salary of one person -months =1500 LE.
- e) Total salaries = $13 \times 1500 \times 3 = 58500$ LE.
- f) Other cost = $70000 - 58500 = 11500$ LE.

19.User guide:

1. Home Page:

Welcome to our online hospital website page.

- You can know more about our hospital by visiting "[About](#)" page.
- a. If you are visiting our website for the first time press "[Create new account](#)" button to sign up and try our services.
- b. If you already have an account just press on "log in" to go to your page.



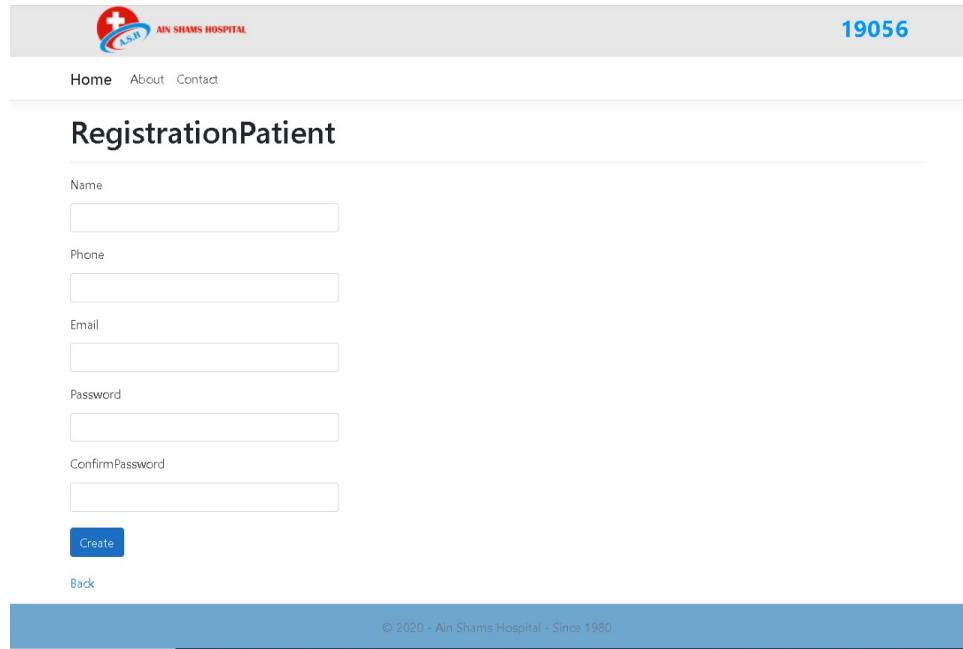
Figure 147 User Guide: Home Page

Figure 148 User Guide: About Page

1. Create new account:

1. Press "Create new account" button in the home page.
2. Activation code page will appear, just enter your code and press "create".
- Hint: enter (0000) if you are a patient; otherwise enter your code from hospital.
3. Fill your information in registration page then press "Create".

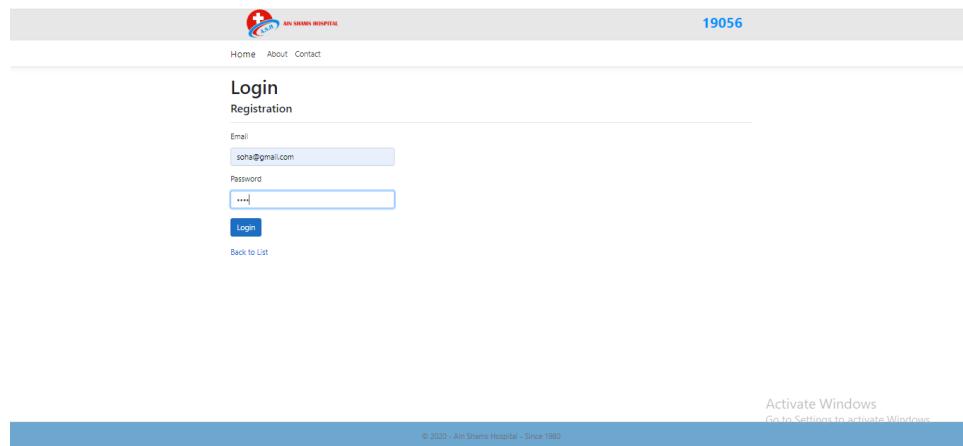
Figure 149 User Guide: Activation Code Page



The screenshot shows the 'RegistrationPatient' page of the Ain Shams Hospital website. At the top right is the number '19056'. Below it are navigation links: 'Home', 'About', and 'Contact'. The main title 'RegistrationPatient' is centered above a form. The form consists of five input fields: 'Name' (empty), 'Phone' (empty), 'Email' (empty), 'Password' (empty), and 'Confirm Password' (empty). Below the fields are two buttons: a blue 'Create' button and a grey 'Back' button. At the bottom of the page is a blue footer bar with the text '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 150 User Guide: Patient Registration Page

2. Log in:
1. To log in press on "Log in" button in the home page.
2. Enter your Email and password.
3. Press log in.



The screenshot shows the 'Login' page of the Ain Shams Hospital website. At the top right is the number '19056'. Below it are navigation links: 'Home', 'About', and 'Contact'. The main title 'Login' is centered above a form. The form has two input fields: 'Email' containing 'sons@gmail.com' and 'Password' containing '****'. Below the fields is a blue 'Login' button. At the bottom of the page is a blue footer bar with the text 'Activate Windows' and 'Go to Settings to activate Windows', along with the copyright notice '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 151 User Guide: Login Page

3. Donation:

1. To fill a donation request press "Donation".
2. Enter your name and fill all the required information.
3. Click "Send".

The screenshot shows a web browser window titled 'Donation' with the URL 'localhost:44359/ICorner/Donation'. The page is titled 'Donation Form' and contains four input fields: 'Enter Your Full Name', 'Enter Your Email', 'PhoneNumber', and 'Choose Day'. Below these fields is a date input field with the placeholder 'mm/dd/yyyy'. At the bottom of the form are two buttons: a blue 'Send' button and a black 'Back' button.

Figure 152 User Guide: Donation Page

2. Patient section:

1. View patient's page :

After successfully logging in patient's home page will appear.

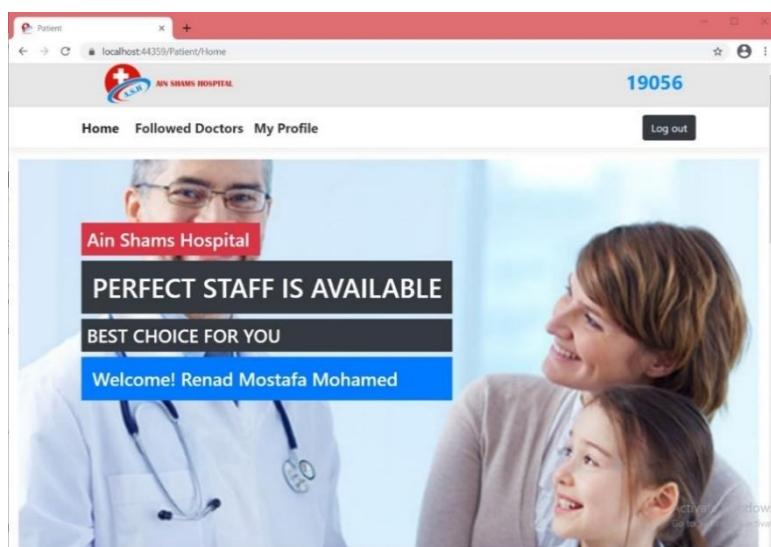


Figure 153 User Guide: Patient Home Page

2. services & offers :

You can find the offers in the main page after logging in.

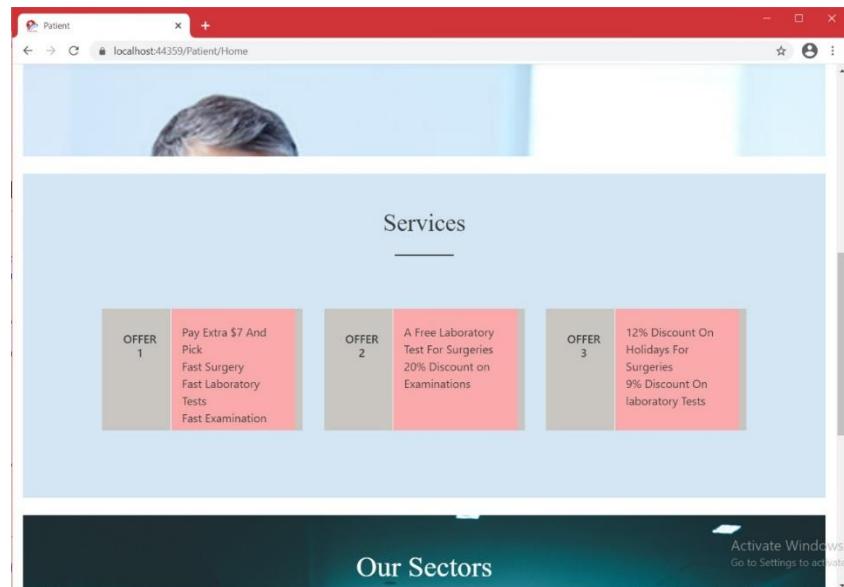


Figure 154 User Guide: Patient Home Page Services Section

3. View different clinics:

Also you can find our different clinics in the hospital.

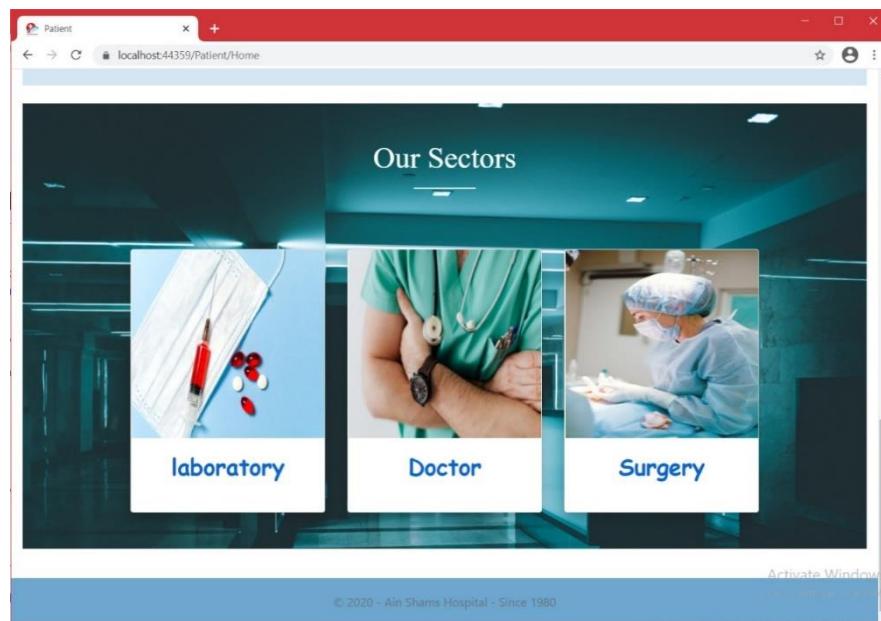


Figure 155 User Guide: Patient Home Page Hospital Sectors Section

4. Add appointment with doctor:
1. Click on "**Doctor**" to view different clinics and doctors.
 2. From the menu select your clinic then click "**Next**".

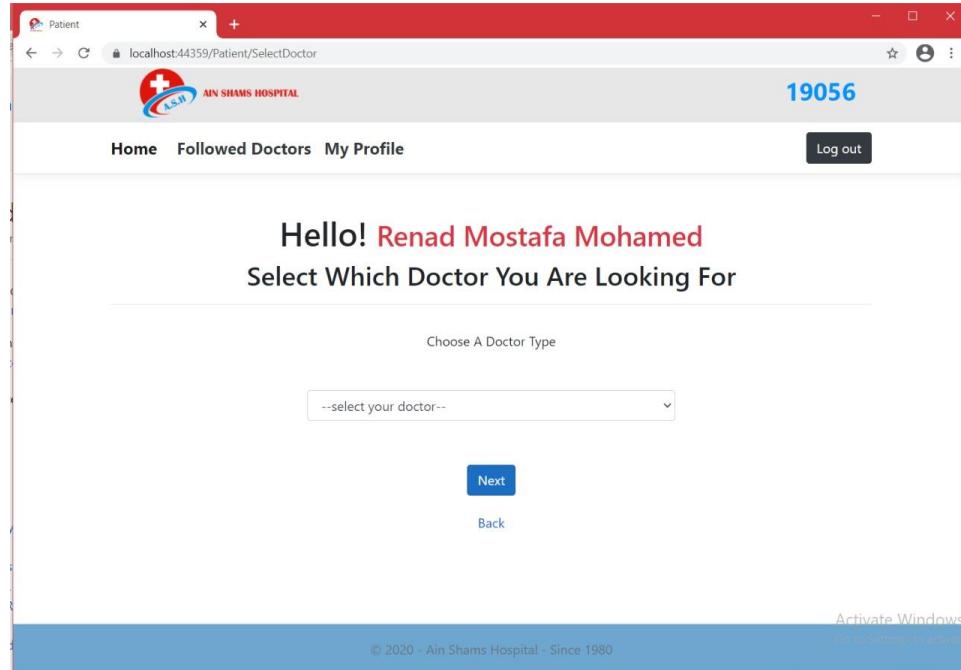


Figure 156 User Guide: Add appointment with doctor Page Part1

3. Select Doctor's name and appointment date and time.
4. Type your previous medicals record.
5. Finally choose your pay method.

The screenshot shows a web browser window titled 'Patient' with the URL 'localhost:44359/Patient/Doctor'. The page has a header with 'Home', 'Followed Doctors', 'My Profile', and a 'Log out' button. Below the header, the word 'Doctor' is centered. A section titled 'Choose A Doctor' contains a dropdown menu with the placeholder '--select your doctor--'. Below this are fields for 'Date' (mm/dd/yyyy) and 'Hour' (dropdown menu). A text area labeled 'Please Enter Your Medical Record' contains placeholder text: 'eg: What medications do you take at home? What is each medicine for? What is the dose?'. At the bottom of the page are two blue buttons: 'PayOnline' and 'PayOffline'. A 'Back' link is located at the very bottom left.

Figure 157 User Guide: Add appointment with doctor Page Part2

A. Online Pay view

The screenshot shows a web browser window titled 'Patient' with the URL 'localhost:44359/Patient/Payment'. The page is titled 'Pay Invoice' and features logos for VISA, MasterCard, American Express, and Discover. It includes fields for 'Payment amount' (\$120), 'Card Number', 'Name on card', 'Expiry Date' (mm/dd/yyyy), 'Security Code', 'ZIP/Postal code', and a large green 'Pay \$120' button. A 'Back' link is at the bottom left. A watermark at the bottom right reads 'Activate Windows Go to Settings to activate'.

Figure 158 User Guide: Online Payment Page

B. Offline Pay view

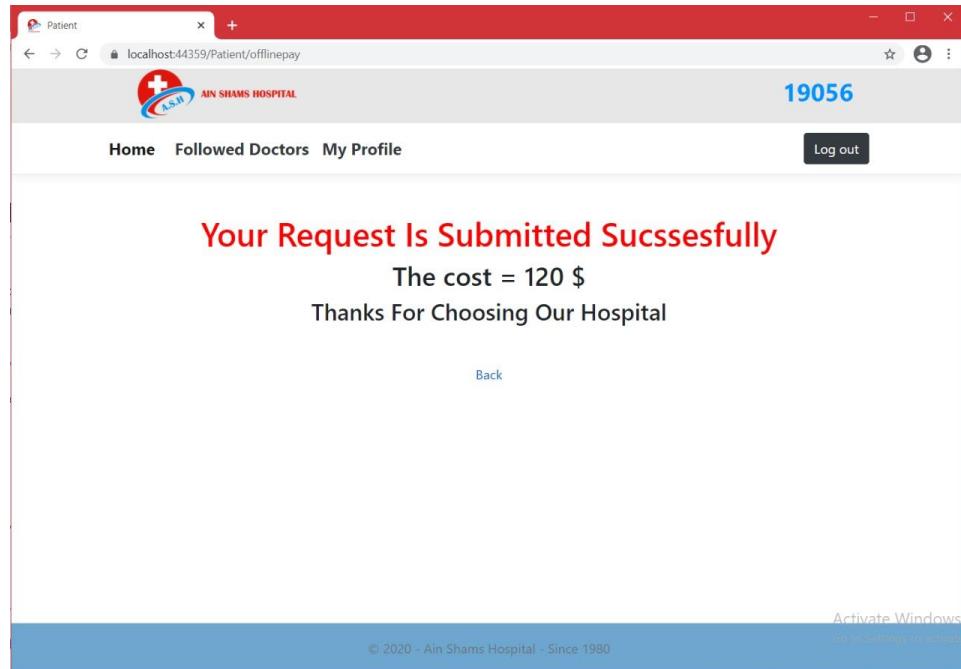


Figure 159 User Guide: Offline Payment Page

5. Add appointment in the lab:

1. Click on "Laboratory" to view different tests you can do.
2. Select your lab then click "Next"
3. From the menu select your test, date and time.
4. Type your previous medicals record if there.
5. Finally choose your pay method.

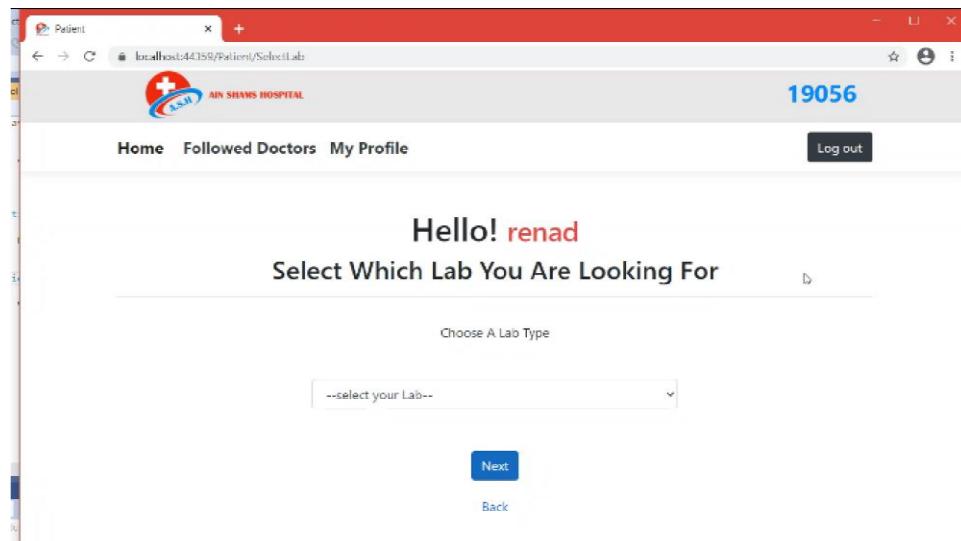


Figure 160User Guide: Add appointment with lab Page Part1

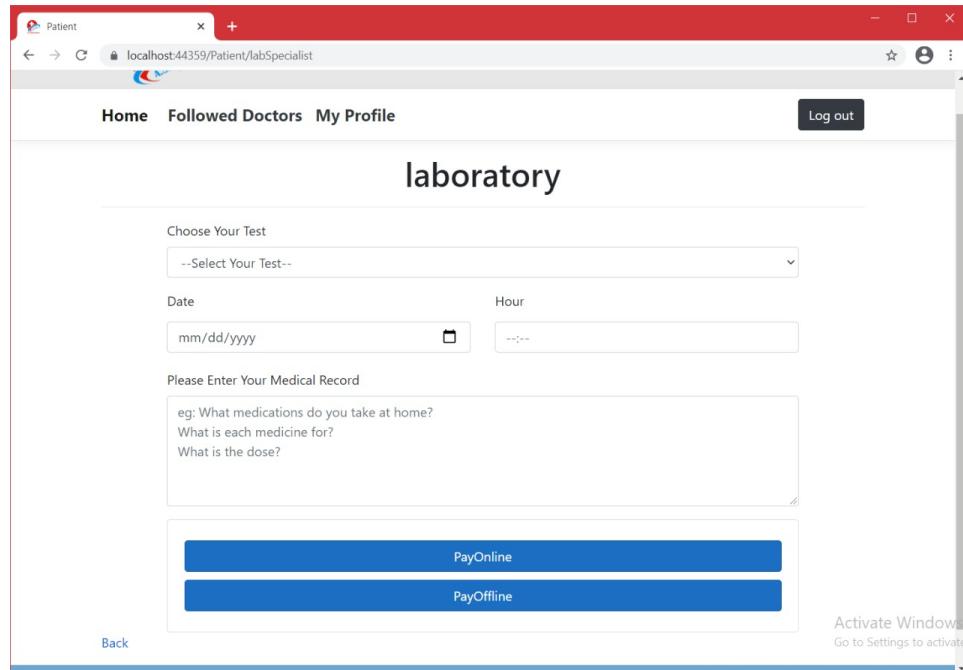


Figure 161User Guide: Add appointment with lab Page Part2

6. Add surgery with a doctor:
 1. Click on "Surgery".
 2. From the menu select your doctor, date and time.
 3. Type your previous medicals record if there.
 4. Finally choose your pay method.

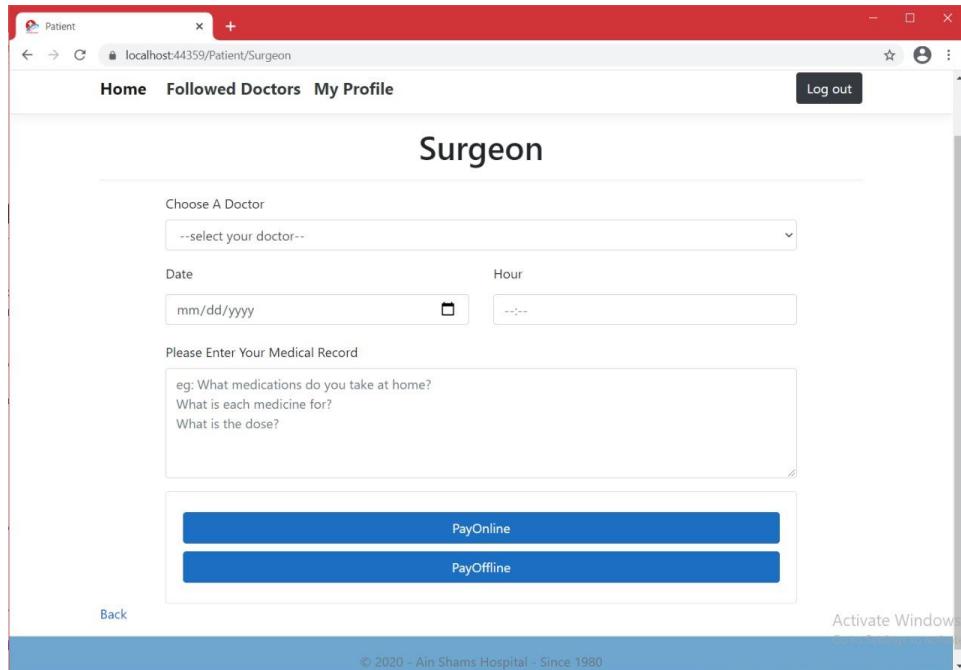


Figure 162 User Guide: Add Surgery with doctor Page

7. Follow doctors:

Click on “Followed Doctors” to view your followed doctors list and their schedule's.

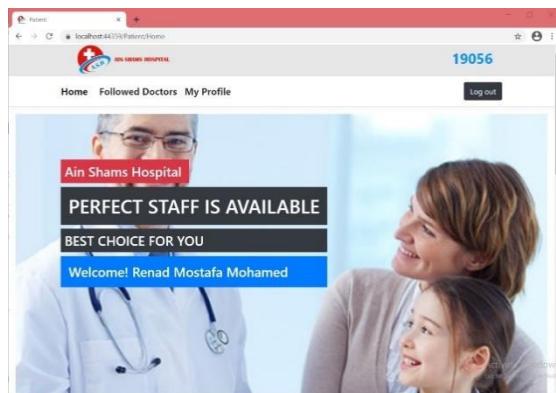


Figure 163 User Guide: Followed Doctors in Home Page

8. View your profile:

In the main page Click on “Profile” to view your profile.

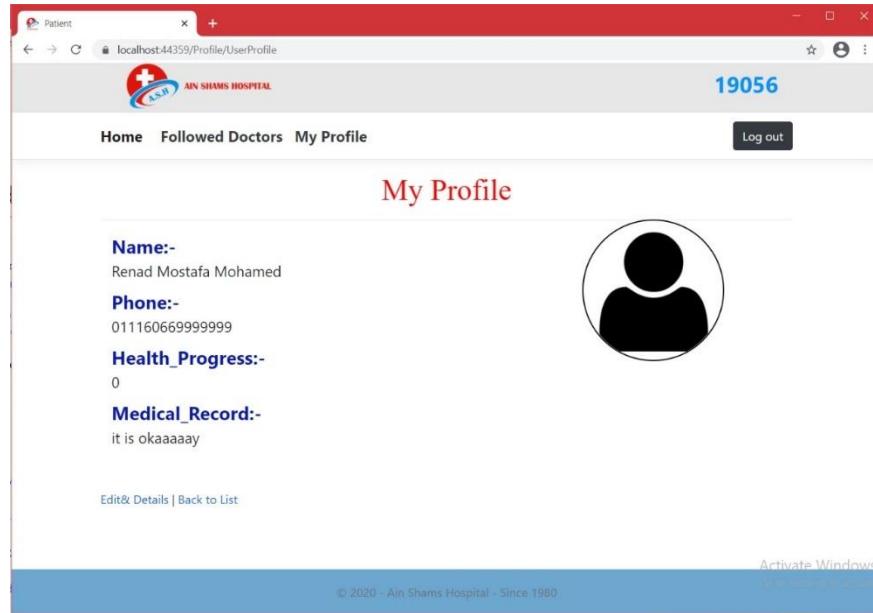


Figure 164 User Guide: Patient Profile Page

1. If you want to edit your information press "Edit Details"

The screenshot shows the "Edit Your Profile" page. The header includes the "AIN SHAMS HOSPITAL" logo and the number "19056". Below the header, there's a navigation bar with "Home", "About", "Contact", and a "Log out" button. The main content area is titled "Edit Your Profile". It contains three input fields:

- Name:** nur
- Phone:** 01111122344
- Medical_Record:** rrrrrrrrrrrcccccc

Below these fields is a section titled "Your Progress" with a progress bar. Underneath is another section titled "Your Current Hospital" showing "Hod El Marsoud". At the bottom, there is a "SaveChanges" button and a "Back" link.

Figure 165 User Guide: Patient Edit Profile Page

- After editing your information click on “Save changes” to save your new information.

3. Manager section:

1. View Manager's page :

After successfully logging in Manager's home page will appear.

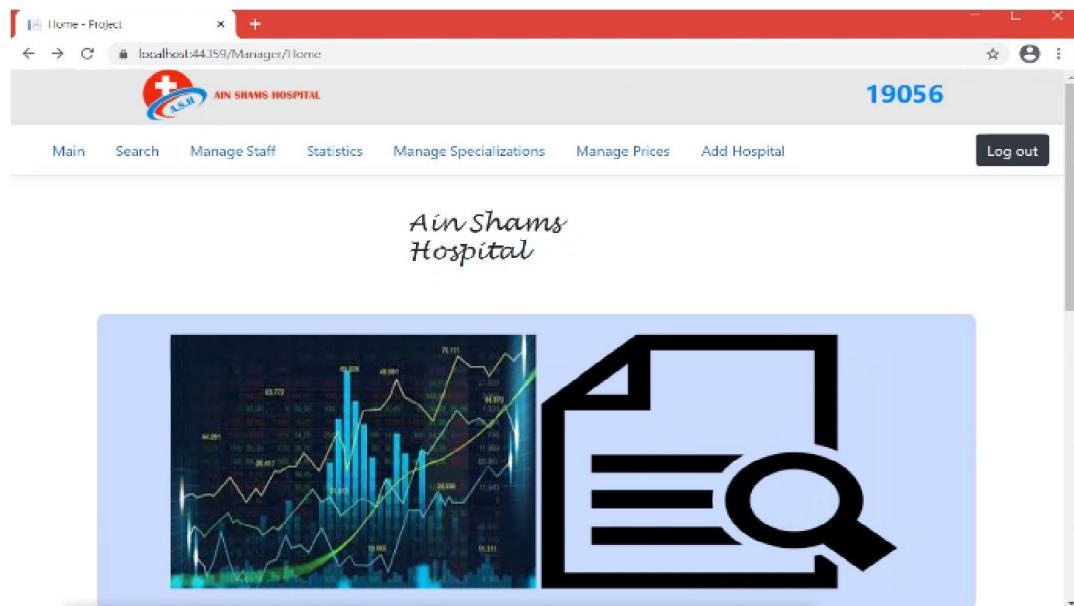


Figure 166 User Guide: Manager Home Page part1



Figure 167 User Guide: Manager Home Page part2

2 Manager search for certain staff:

1. Click on "Search" in the nave bar or in the home page
2. Manager writes a staff name and Search for it's information.
 3. click "Search".
 4. The Staff information will appear in the card

The screenshot shows a web application interface for managing staff. At the top, there is a red header bar with the title 'ManagerHome - Project'. Below the header is a navigation menu with several links: 'Main', 'Search', 'Manage Staff', 'Statistics', 'Manage Specializations', 'Manage Prices', 'Add Hospital', and 'Log out'. The main content area is titled 'Staff Personal Information' and contains five input fields: 'Name', 'Phone', 'Starting Day', 'Specialization', and 'Email'. Above these input fields is a blue 'Search' button. The browser window has a vertical scroll bar on the right side.

Figure 168 User Guide: Manager Search for Staff Page

3. Manage staff:

1. Manager writes a certain name and deletes it from the hospital staff.
2. Click on manage staff in the nave bar
3. Manager writes a staff name and deletes it from the hospital staff.
4. . click "Delete".
5. The Staff information will be deleted from the database

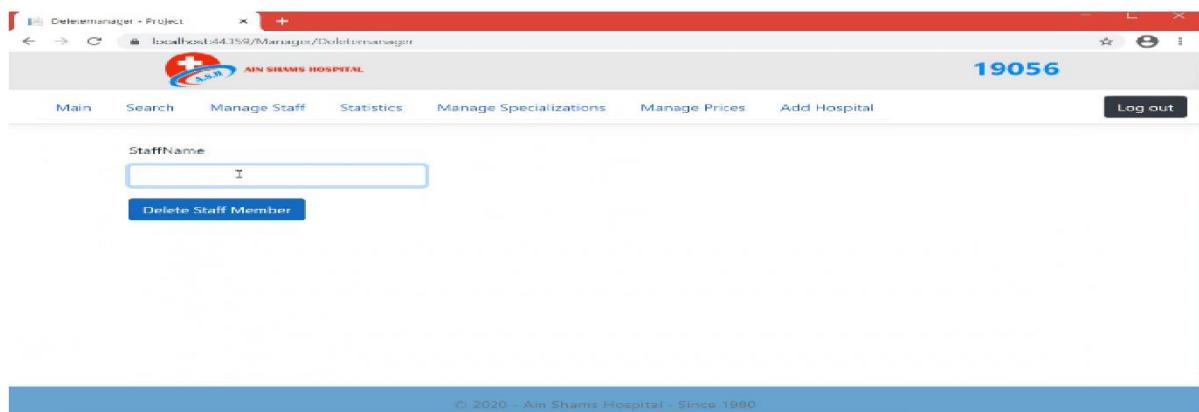


Figure 169 User Guide: Manager Manage Staff Page

3.Statistics:

1. Click on Statistics in the nave bar
2. Manger writes a certain period of time
3. click "Set Date"
4. Now you can open any card to see its income

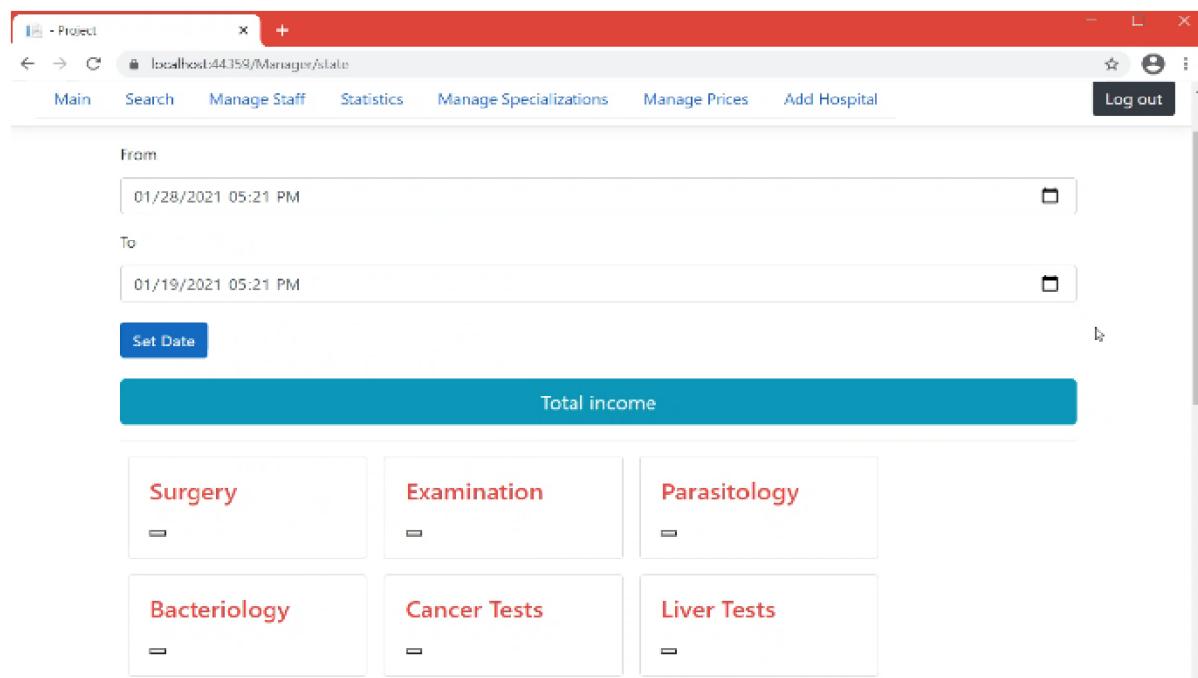
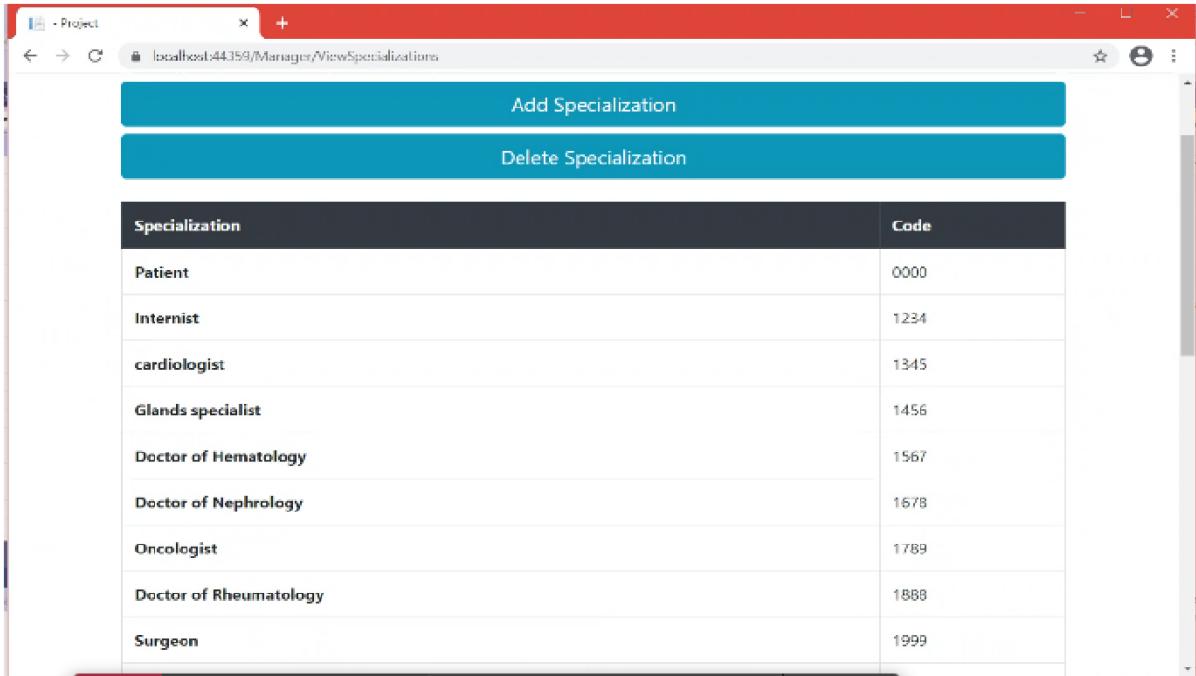


Figure 170 User Guide: Manager Statistics Page

4. Manage specializations:

1. Click on manage Specialization in the nave bar
2. A table contain all the Specialization codes will appear
3. Click on “Add specialization” or Click on “Delete specialization”

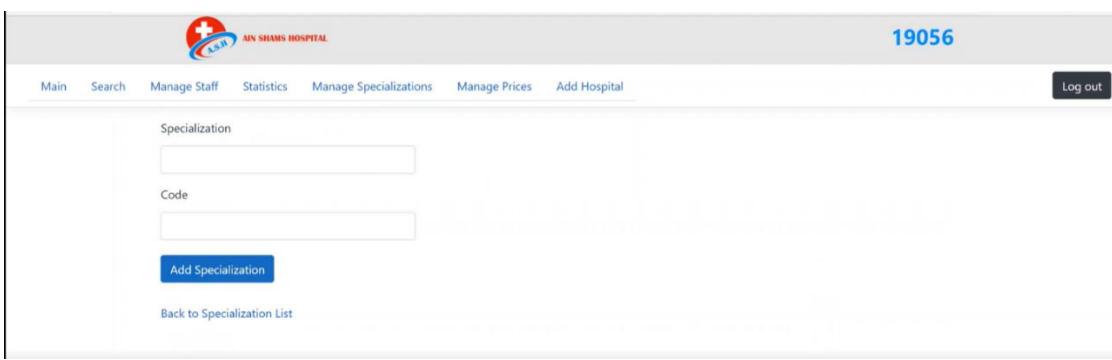


Specialization	Code
Patient	0000
Internist	1234
cardiologist	1345
Glands specialist	1456
Doctor of Hematology	1567
Doctor of Nephrology	1678
Oncologist	1789
Doctor of Rheumatology	1888
Surgeon	1999

Figure 171 User Guide: Manager Manage Specializations Page

4.1 Add specializations:

4. Insert the name of specialization wanted to be added
5. Insert the code for the new specialization wanted to be added
6. Click on “Add specialization”



Main Search Manage Staff Statistics Manage Specializations Manage Prices Add Hospital Log out

19056

Specialization

Code

Add Specialization

[Back to Specialization List](#)

Figure 172 User Guide: Manager Add Specialization Page

4.2Delete specializations:

7. Insert the name of specialization wanted to be deleted
8. Click on “Delete specialization”

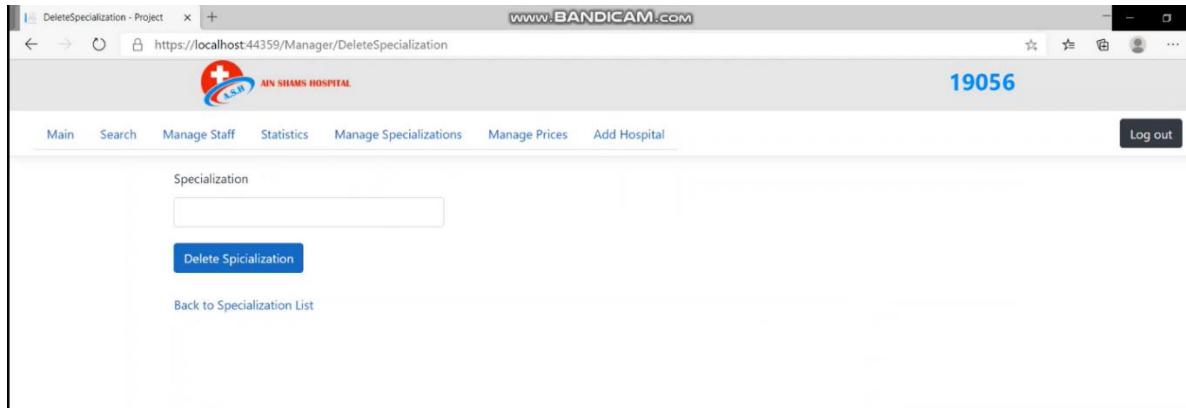


Figure 173 User Guide: Manager Delete Specialization Page

5.Manage prices:

1. Click on manage Prices in the nave bar
2. A table contain all Services prices will appear
3. Click on “Add Service” or Click on “Delete Service” or Click on “Edit Price”

The screenshot shows a web browser window titled "Project". The URL is "localhost:44359/Manager/ViewPrices". The page header includes the "AIN SHAMS HOSPITAL" logo and the number "19056". The navigation menu at the top has links for Main, Search, Manage Staff, Statistics, Manage Specializations, Manage Prices, and Add Hospital. On the right, there is a "Log out" button. The main content area features three large buttons: "Add Service" (blue), "Edit Prices" (grey), and "Delete Service" (blue). Below these buttons is a table showing service names and their prices.

Name	Price
Surgery	2600
Examination	500
Parasitology	600
Bacteriology	160
Cancer Tests	120

Figure 174 User Guide: Manager Manage Prices Page

5.1Add Service:

1. Click on “Add Service”
2. Insert the name for the service wanted to be added
3. Insert the price for the new service wanted to be added
4. Click on “Add Service”

The screenshot shows a web-based application interface for managing hospital services. At the top right, the user ID '19056' is displayed. The top navigation bar includes links for Main, Search, Manage Staff, Statistics, Manage Specializations, Manage Prices, Add Hospital, and Log out. Below the navigation, there are two input fields: 'ServiceName' and 'ServicePrice'. A blue 'Add Service' button is positioned between them. At the bottom left, a link 'Back to Service List' is visible.

Figure 175 User Guide: Manager Add Service Page

5.2Delete Service:

5. Click on “Delete Service”
6. Insert the name for the service wanted to be deleted
7. Click on “Delete Service”

This screenshot shows the same web-based application interface as Figure 175, but with a different focus. It features the same header, navigation bar, and input fields for 'ServiceName' and 'ServicePrice'. However, the central element is a blue 'Delete Service' button, which is highlighted or larger than the 'Add Service' button in Figure 175. The 'Back to Service List' link at the bottom left is also present.

Figure 176 User Guide: Manager Delete Service Page

5.3Edit Price:

8. Click on “Edit Price”
9. Insert the name for the service wanted to be edited
10. Insert the price for it
11. Click on “Edit Price”

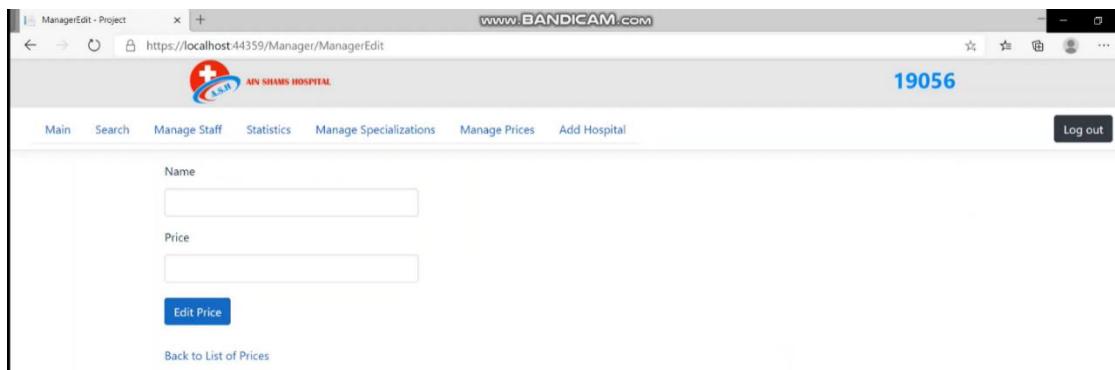


Figure 177 User Guide: Manager Edit Price Page

6.Add hospital:

1. Click on “Add Hospital” in the nave bar
2. A table contain all the Hospitals which the Patient can transfer to it will appear
3. Click on “Add Hospital”

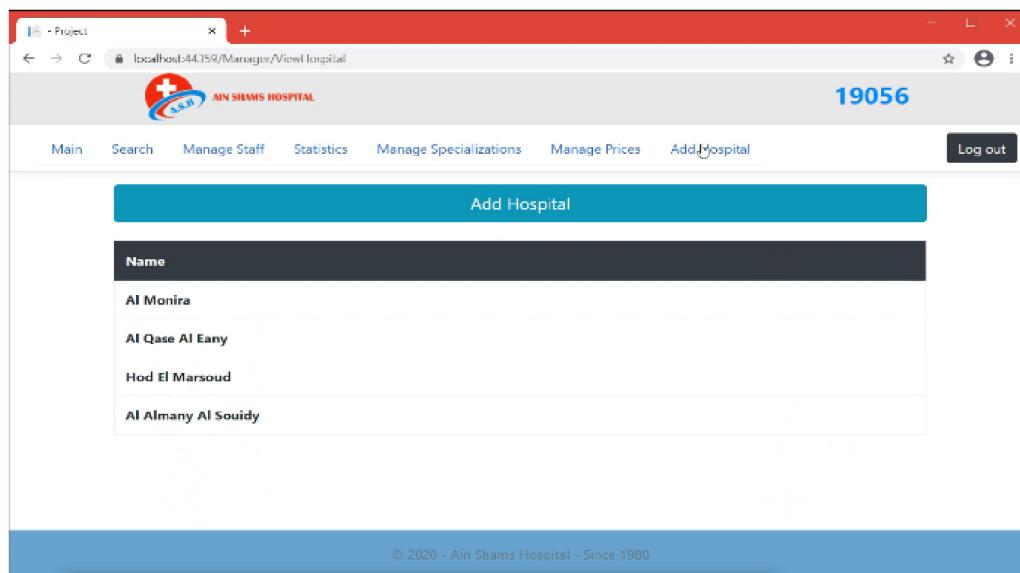


Figure 178 User Guide: Manager View Hospital Page

6.1Add Hospital:

1. Click on “Add Hospital”
2. Insert the name of Hospital wanted to be added
3. Click on “Add Hospital”

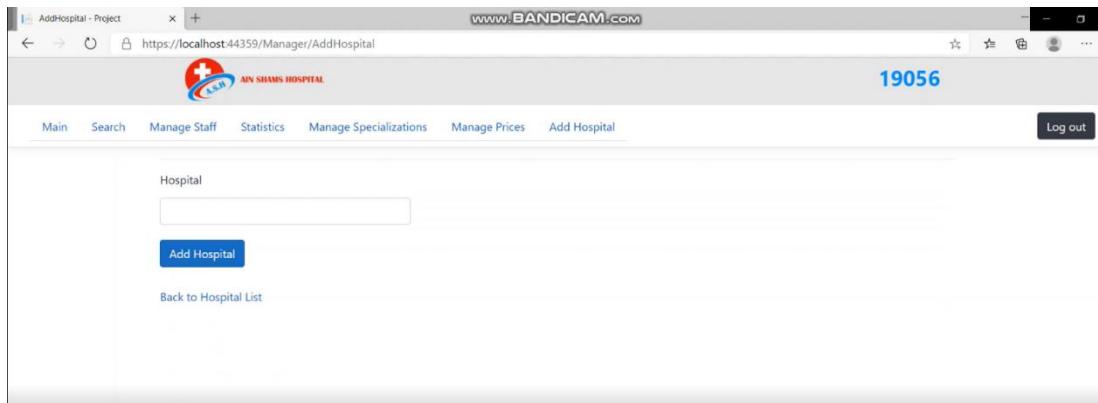


Figure 179 User Guide: Manager Add Hospital Page

4. Front desk section:

1. View Front desk 's page :

After successfully logging in Front desk 's home page will appear.

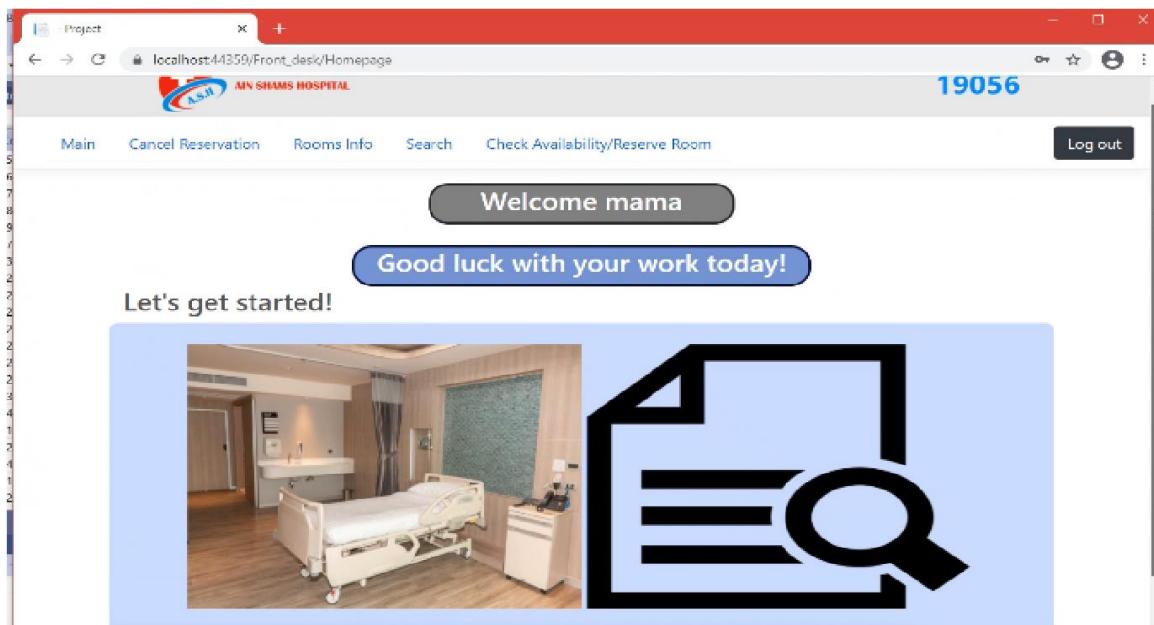


Figure 180 User Guide: Front Desk Home Page

2.Check Availability/Reserve Room:

1. Click on check Availability in the nave bar or on reserve room in the homepage
2. Two options will appears Rooms or surgery rooms
3. Click on “Rooms” or Click on “surgery rooms”

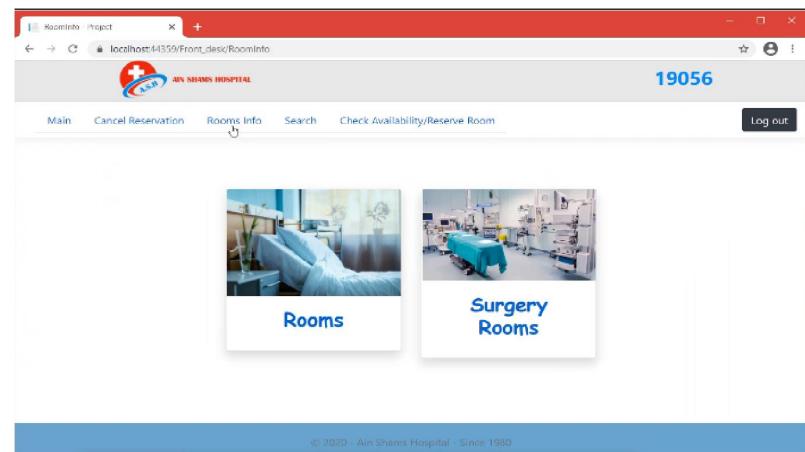


Figure 181 User Guide: Front Desk Check Room Availability Page

2.1 Check availability for surgery rooms:

1. Click on “Surgery Rooms”
2. Insert the START hour and the end hour expected for the surgery
3. Click on “Reserve Surgery room”

A screenshot of a web browser window titled 'Shoresservation - Project'. The URL is 'localhost:44359/Front_desk/SRoomreservation'. The page header includes the 'AIN SHAMS HOSPITAL' logo and the number '19056'. Below the header, there are two input fields: 'Start_Hour' and 'End_Hour', both with date and time pickers. Below these fields is a blue button labeled 'Reserve Surgery Room'. At the bottom of the page, there is a copyright notice: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 182 User Guide: Front Desk Check Surgery Room Availability Page part1

2.2 Check availability for surgery rooms:

1. Reserve room page will appear to check the room availability
2. Insert the dates you want to reserve the room From & To expected for the patient
3. Click on "Show available room"
4. Will appear the surgery room number and the room number which are available in those dates
5. Click on "Reserve Room or Click on "Transfer Patient"

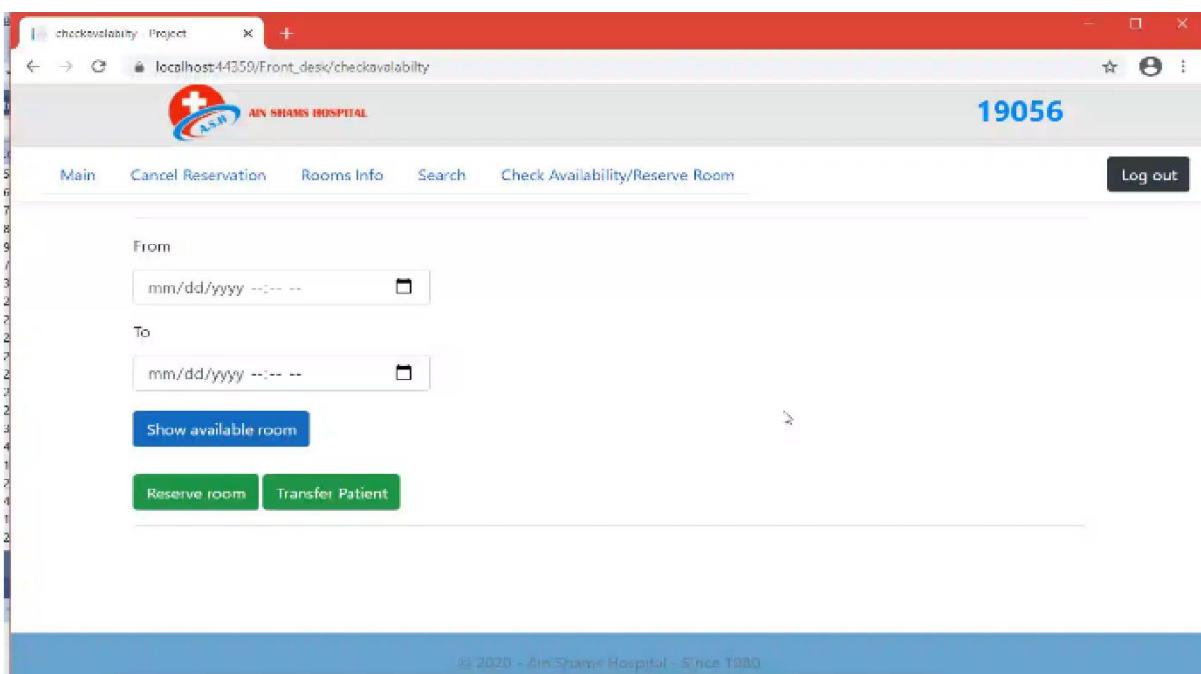
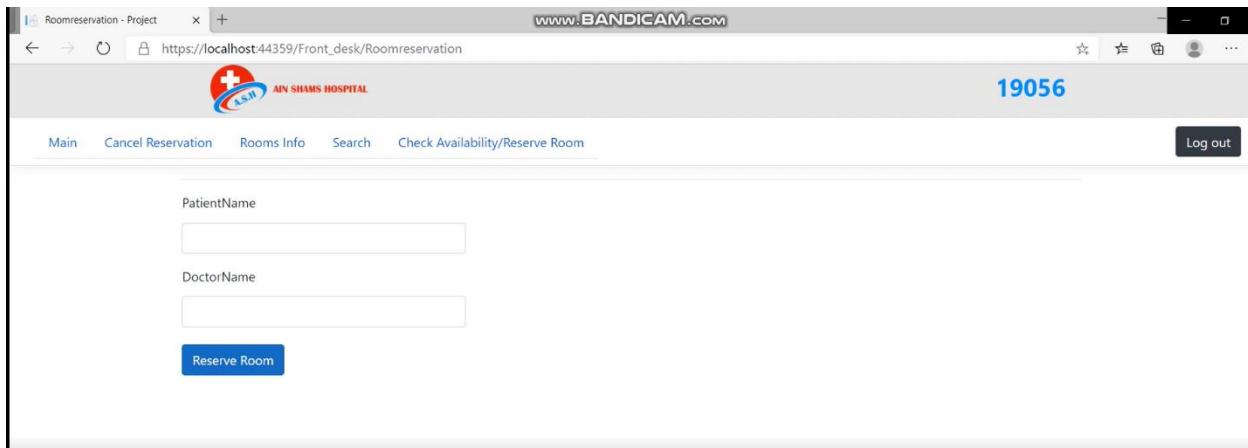


Figure 183 User Guide: Front Desk Check Surgery Room Availability Page part2

4. Reserve room:

1. Reserve room page will appear
2. Insert the patient Name & the doctor who is followed by the patient
3. Click on “Reserve room ”

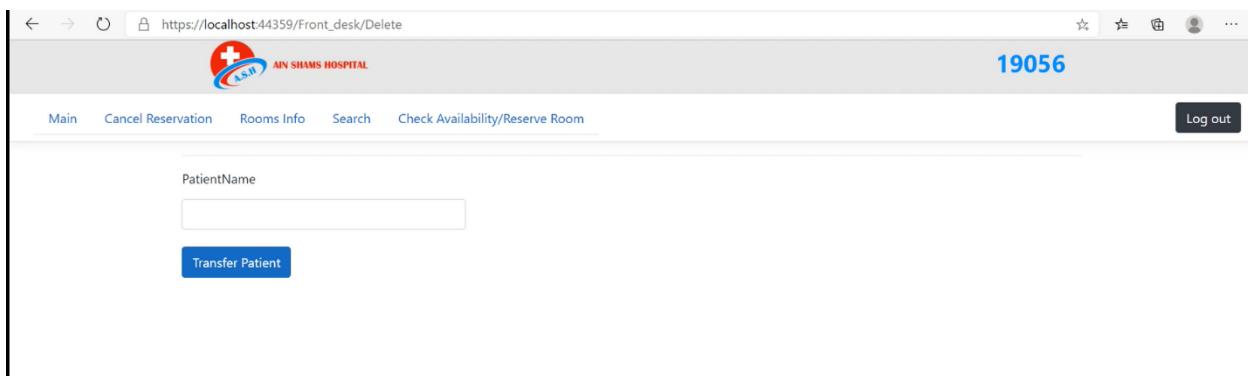


The screenshot shows a web browser window for 'Roomreservation - Project' at 'https://localhost:44359/Front_desk/Roomreservation'. The header includes the hospital logo 'AIN SHAMS HOSPITAL' and the number '19056'. The menu bar has links for 'Main', 'Cancel Reservation', 'Rooms Info', 'Search', 'Check Availability/Reserve Room', and 'Log out'. Below the menu, there are two input fields: 'PatientName' and 'DoctorName', each with a corresponding text input box. A blue 'Reserve Room' button is positioned below the DoctorName field.

Figure 184 User Guide: Front Desk Reserve Room Page

5. Transfer Patient:

1. Transfer page will appear
2. Insert the patient Name who going to be transferred
3. Click on “Transfer Patient ”



The screenshot shows a web browser window for 'https://localhost:44359/Front_desk/Delete'. The header includes the hospital logo 'AIN SHAMS HOSPITAL' and the number '19056'. The menu bar has links for 'Main', 'Cancel Reservation', 'Rooms Info', 'Search', 'Check Availability/Reserve Room', and 'Log out'. Below the menu, there is one input field labeled 'PatientName' with a corresponding text input box. A blue 'Transfer Patient' button is positioned below the input field.

Figure 185 User Guide: Front Desk Transfer Patient Page

6.Search

1. Click on Search in the nave bar or on Search in the homepage Transfer page will appear
2. Insert the patient Name
3. Click on “Search”

The screenshot shows a web browser window with the following details:

- Title Bar:** Search - Project
- Address Bar:** localhost:44359/Front_desk/Search
- Header:** AIN SHAMS HOSPITAL logo with a red cross and ASH letters, and the number 19056.
- Navigation:** Main, Cancel Reservation, Rooms Info, Search (with a magnifying glass icon), Check Availability/Reserve Room, Log out.
- Input Fields:** patientName (text input field), Today (date input field showing mm/dd/yyyy), and a Search button.
- Footer:** © 2020 - Ain Shams Hospital - Since 1980

Figure 186 User Guide: Front Desk Search Page

7.Rooms info:

1. Click on Room Info in the nave bar
2. Two options will appear Rooms or surgery rooms
3. Click on “Rooms” or Click on “surgery rooms”

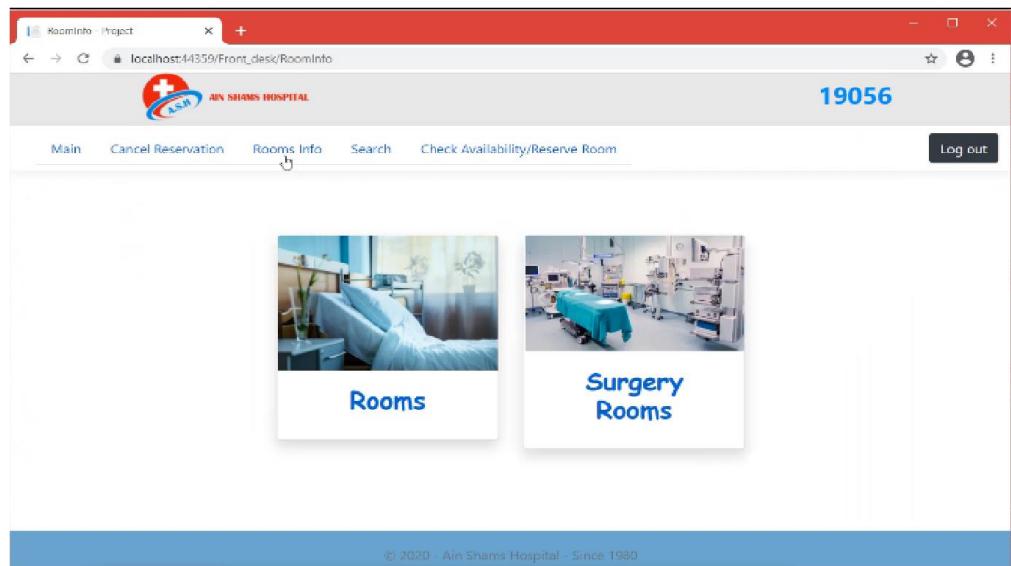


Figure 187 User Guide: Front Desk Room Info Page

7.2.Surgery Rooms info:

1. Click on “Surgery Rooms”
2. Cards with Surgery room numbers will appear

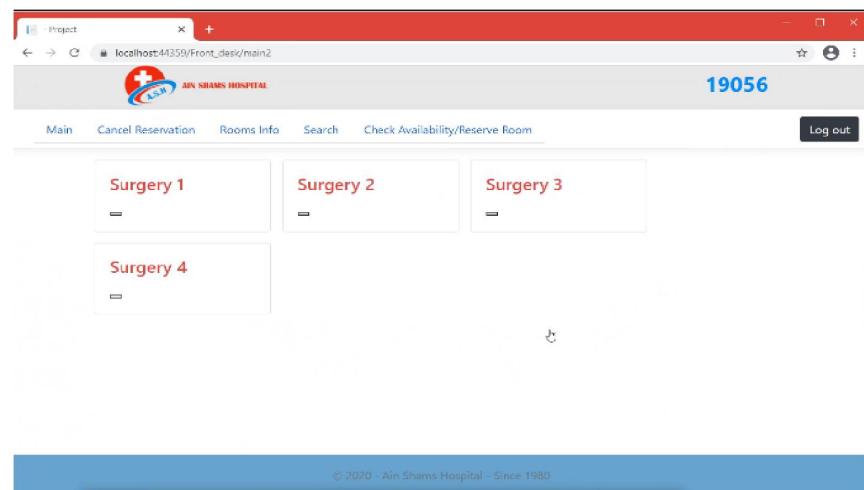


Figure 188 User Guide: Front Desk Surgery Room Info Page

7.3. Rooms info:

3. Click on “Rooms”
4. Cards with the room numbers will appear

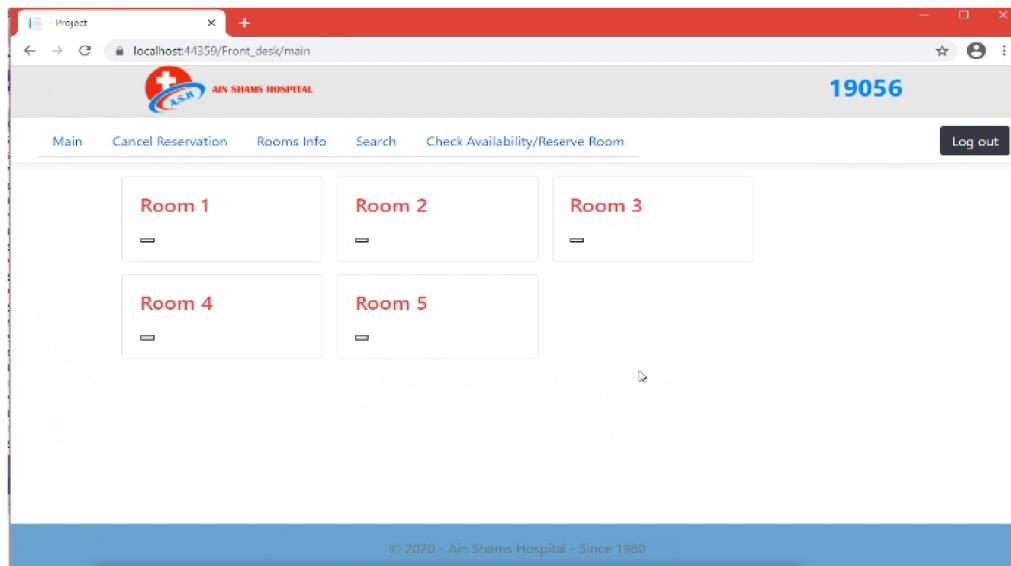


Figure 189 User Guide: Front Desk Rooms Info Page

Click on any card even the surgery or the room will appear a able contain all the details of reservation for this room

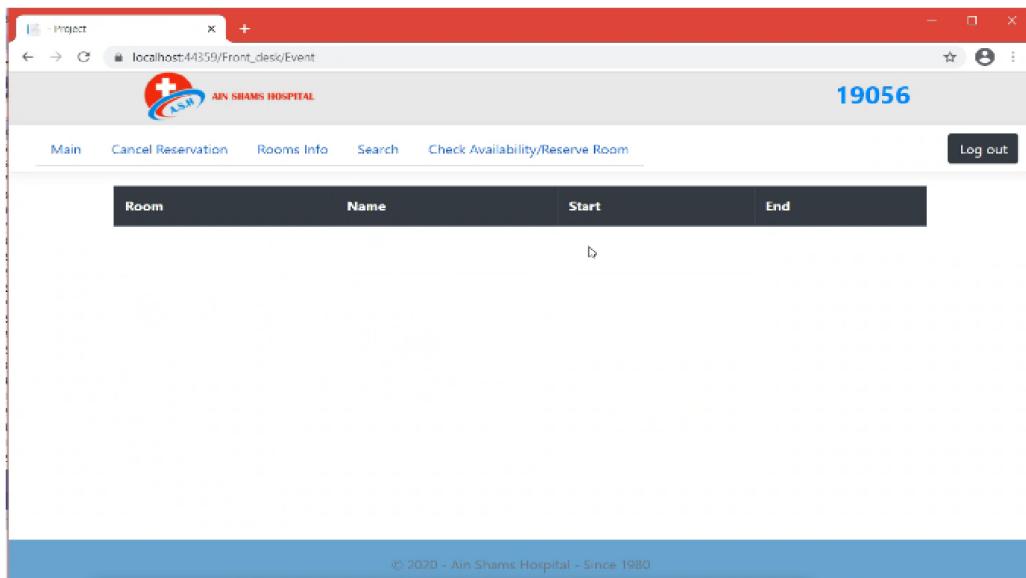


Figure 190 User Guide: Front Desk Room Details Page

8.Cancel Reservation:

1. Click on Cancel in the nave bar
2. Insert the patient Name
3. Click on “Cancel Reservation ”

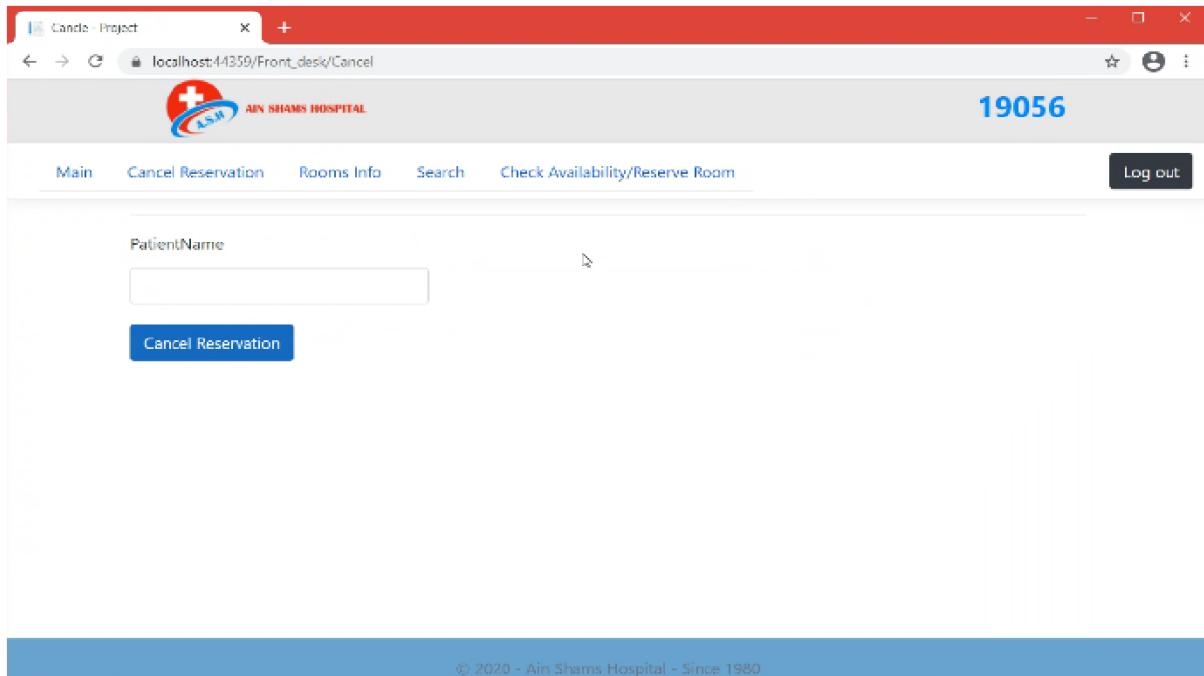


Figure 191 User Guide: Front Desk Cancel Reservation Page

4. Finance section:

1. View Finance 's page :

After successfully logging in Finance 's home page will appear.



Figure 192 User Guide: Finance Home Page

2.Statistics:

1. Click on Statistics in the home page
2. writes a certain period of time
3. click "Set Date"
4. Now you can open any card to see its income

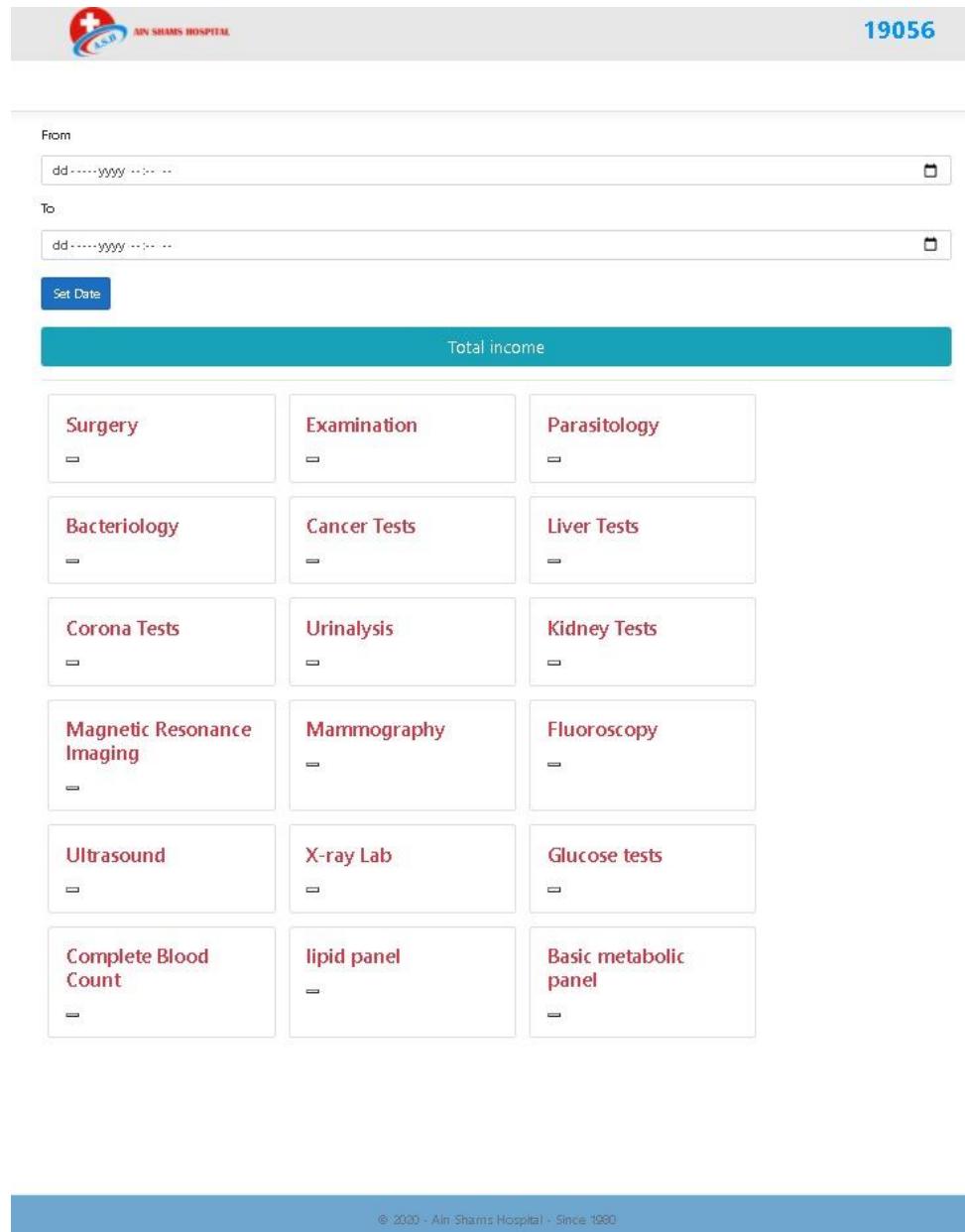


Figure 193 User Guide: Finance Statistics Page

The screenshot shows a user interface for managing financial services. At the top right, the number '19056' is displayed. On the left, there's a logo for 'AIN SHAMS HOSPITAL' with a red cross and the letters 'A.S.H'. On the right, a 'Log out' button is visible. Below the header, a 'Main' link is underlined. A table titled 'Service' lists three entries: 'Kidney Tests' (name: nour, money: 300, date: 2021-01-27T10:00), 'lipid panel' (name: nour, money: 450, date: 2021-01-22T09:00), and a total row 'Total = 750'. At the bottom of the page, a blue footer bar contains the text '© 2020 - Ain Shams Hospital - Since 1980'.

Service	Name	Money	Date
Kidney Tests	nour	300	2021-01-27T10:00
lipid panel	nour	450	2021-01-22T09:00
Total =		750	

Figure 194 User Guide: Finance Service Income Details Page

3.Patient Bill:

1. Click on “Patient Bill” in the home page
2. Insert the patient Name & Todays date
3. click "Show Bill"
4. A table will appear which contain the Patient bill

The screenshot shows a user interface for viewing a patient bill. At the top right, the number '19056' is displayed. On the left, there's a logo for 'AIN SHAMS HOSPITAL' with a red cross and the letters 'A.S.H'. On the right, a 'Log out' button is visible. Below the header, a 'Main' link is underlined. The word 'Bill' is centered above a form. The form includes fields for 'PatientName' (containing 'renad') and 'Todaydate' (containing '20-Jan-2021'). A 'Show Bill' button is present. Below the form is a table with columns 'Type' and 'Price'. The table has one row with the text 'Total=0'. At the bottom of the page, a blue footer bar contains the text '© 2020 - Ain Shams Hospital - Since 1980'.

Type	Price
Total=0	

Figure 195 User Guide: Finance Service Patient Bill Page

6. Doctor section:

1. View doctor's home page :

After successfully logging in doctor's home page will appear.

Doctor can easily view his schedule & profile.

Interaction style "menu selection"

Also doctor can search for patient from a search bar.

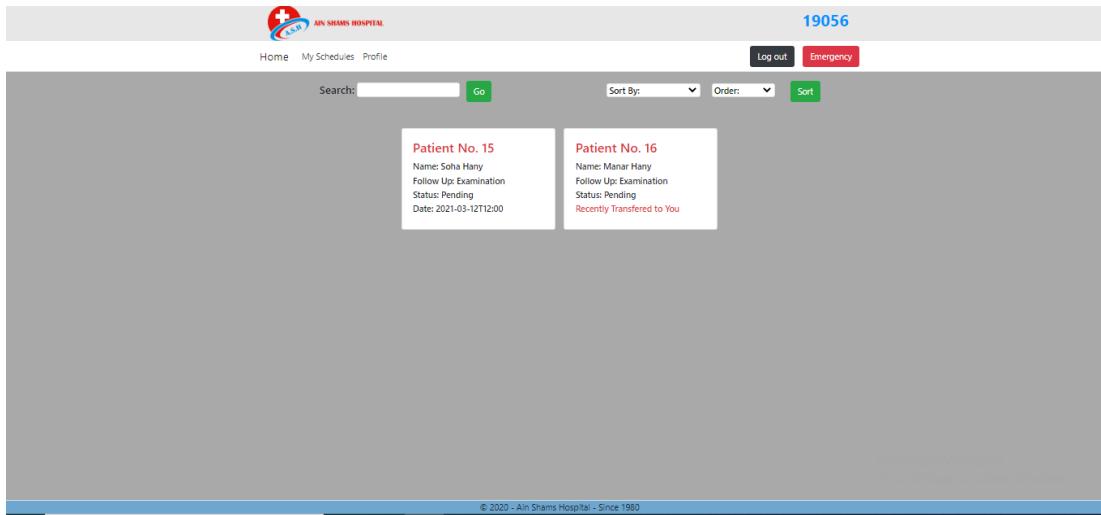


Figure 196 User Guide: Doctor Home Page

5. View doctor's profile page:

When click on profile button doctor can see and change his information and save changes and after change click on save change.

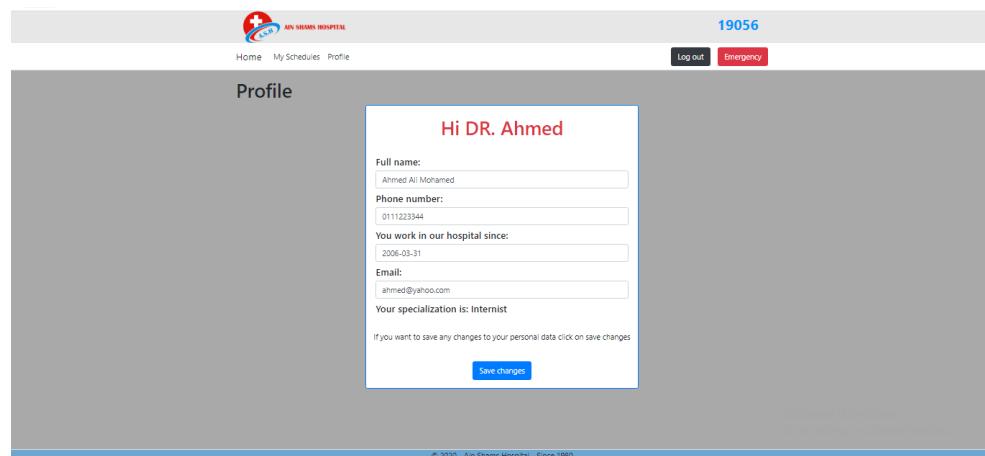


Figure 197 User Guide: Doctor Profile Page

6. View doctor's schedule page:

When click on schedule button Doctor can see his schedule.

The screenshot shows the 'My schedule' section of the doctor's dashboard. At the top right, the user ID '19056' is displayed. Below it are two buttons: 'Log out' (black) and 'Emergency' (red). The main area is titled 'My schedule' and contains a table with two columns: 'Date' and 'Follow Up'. The table has three rows: the first row is empty; the second row shows 'Examination' under 'Follow Up' for the date '2021-03-12T12:00'; the third row also shows 'Examination' under 'Follow Up' for the same date. At the bottom left of the page, there is a footer note: '© 2020 - Ain Shams Hospital - Since 1980'.

Figure 198 User Guide: Doctor Schedule Page

7. View emergency:

When click on emergency button Doctor can see emergency bed and blood unit amount.

The screenshot displays two tables side-by-side. On the left, under the heading 'Blood_Unit', is a table with columns 'Type:' and 'Amount:'. It lists nine blood types with their respective counts: A+ (28), A- (25), B+ (32), B- (32), O- (40), AB+ (18), AB- (43), O+ (10), and A+ (2). On the right, under the heading 'Emergency bed', is another table with columns 'Type:', 'Available:', 'Start_Working:', and 'End_Working:'. It lists four emergency beds, all of which are currently available ('True') and active ('All') from start to end. The table structure is identical to the one on the left.

Figure 199 User Guide: Doctor Emergency Page

8. Patients follow up:

When click on card doctor can see personal information of this patient and can edit his medical record

and click on edit medical record button to save it, can edit his health progress and

Click on edit health progress button to save it and can edit his status and

Click on edit status button to save it.

The screenshot shows a web interface for managing patient information. At the top right, the number '19056' is displayed. Below the header, there are links for 'Home', 'My Schedules', and 'Profile'. On the far right are 'Log out' and 'Emergency' buttons. The main content area has a title 'Patient personal informations' in green. It displays the patient's name 'Manar Hany', phone number '0126716', email 'manar@gmail.com', health progress '0', and status 'Pending'. Below this, there is a section for 'Patient Medical Record' with a text input field and an 'Edit Medical Record' button. A section for 'Add Medical Record:' is shown with a text input field and an 'Edit Medical Record' button. Another section for 'Add Health Progress:' includes a slider and a text input field with the value '0', accompanied by an 'Edit Health Progress' button. At the bottom, there is a note 'If you need to trasfer patient.' with a 'Transfer' button. The footer contains the copyright notice '© 2020 - Ain Shams Hospital - Since 1960'.

Figure 200 User Guide: Doctor Patient Follow Up Page Part1

This screenshot shows the continuation of the patient follow-up interface. It features the same header and navigation as Figure 200. The main content area includes sections for 'Health Progress: 0' and 'Status: Pending'. Below these are fields for 'Patient Medical Record' and 'Add Medical Record:' with an 'Edit Medical Record' button. The 'Add Health Progress:' section is identical to Figure 200. A new section for 'Add Status:' includes a dropdown menu for 'Status:' and an 'Edit Status' button. At the bottom, there is a note 'If you need to trasfer patient.' with a large 'Transfer' button. The footer is identical to Figure 200.

Figure 201 User Guide: Doctor Patient Follow Up Page Part2

9. Transfer patient to another hospital/doctor:

- When doctor click on transfer button, doctor can transfer patient to another hospital/doctor.
- If doctor choose another department, doctor go to another page to choose department, doctor and follow up type click submit to transfer.
- If doctor choose another hospital, doctor go to another page to choose hospital click submit to transfer.

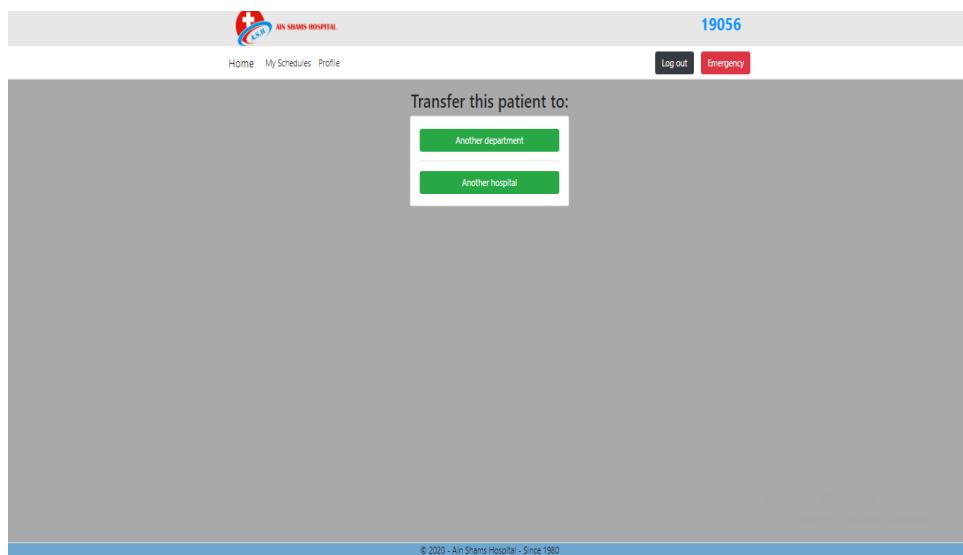


Figure 202 User Guide: Doctor Transfer Patient Page

A screenshot of a web application interface for a doctor. At the top, there is a header with the logo of 'AIN SHAMS HOSPITAL' and the number '19056'. Below the header, there are links for 'Home', 'My Schedules', and 'Profile', along with 'Log out' and 'Emergency' buttons. The main content area is titled 'Transfer this patient to:' and contains three dropdown menus: 'Department of' (with an option 'select a department'), 'Doctor' (with an option 'select a doctor'), and 'Follow Up Type' (with an option 'select a follow up type'). Below these dropdowns is a blue 'submit' button.

Figure 203 User Guide: Doctor Transfer Patient to Another Doctor Page

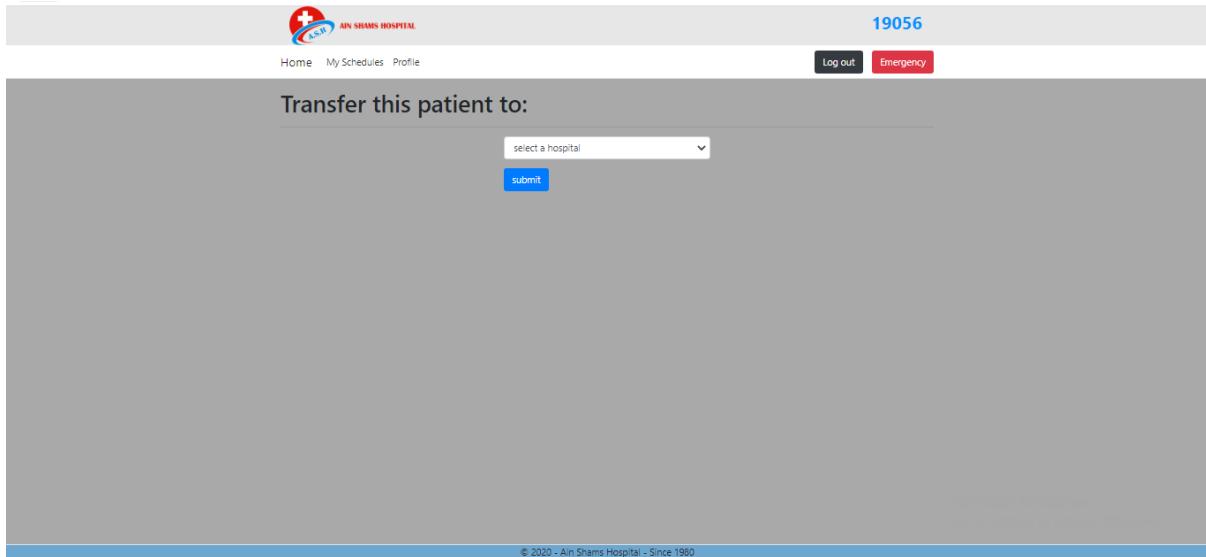


Figure 204 User Guide: Doctor Transfer Patient to Another Hospital Page

10. Lab and Radiology section:

1. View Lab specialist 's home page :

After successfully logging in lab's home page will appear.

Lab specialist can easily view his patient, Also Lab specialist can search for patient from a search bar.

A screenshot of the Lab Home Page. At the top right is the number "19056". Below it are navigation links: "Home", "About", "Contact", "Log out", and a search bar with a green "Go" button. To the right are dropdown menus for "Sort By:" and "Order:", and a green "Sort" button. Two patient cards are displayed: "Patient No. 2" (Name: Atef Ahmed, Follow Up: Corona Tests, Date: 2021-01-26T13:00, Status: Pending) and "Patient No. 1" (Name: Ahmed Nour, Follow Up: Parasitology, Date: 2021-01-19T11:00, Status: Pending). The bottom of the page features a blue footer bar with the text "© 2020 - Ain Shams Hospital - Since 1980".

Figure 205 User Guide: Lab Home Page

2. Patients follow up:

When click on card Lab specialist can see personal information of this patient and can edit his status and

Click on edit status button to save it.

The screenshot shows a web application interface for a patient's personal information. At the top, there is a header with the Ain Shams Hospital logo and the number 19056. Below the header, there is a navigation bar with links for Home, About, Contact, and Log out. The main content area is titled "Patient personal informations". It displays the following information:
Name: Atef Ahmed
Phone: 15598885641
Email: atef_ahmed@gmail.com
Status: In Progress

Below this, there is a section labeled "Add Status:" with a dropdown menu set to "In Progress" and a blue "edit Status" button. At the bottom of the page, there is a footer bar with the text "© 2020 - Ain Shams Hospital - Since 1980".

Figure 206 User Guide: Lab Patient Follow Up Page

9. blood bank section:

- After successfully logging in blood bank 's home page will appear.
- blood bank specialist can edit amount of blood unit, choose amount and click on plus.
- blood bank specialist can add blood unit when click on add choose amount and blood unit and click ok.
- blood bank specialist can delete blood unit when click on delete and click ok.

The screenshot shows a web application interface for managing blood units. At the top, there is a header with the Ain Shams Hospital logo and the number 19056. Below the header, there is a navigation bar with links for Home, About, Contact, and Log out. The main content area is titled "Blood_Unit" and contains a table with the following data:

Type:	Amount:	Edit:	Delete:
A+	28	<input type="text" value="0"/>	
A-	25	<input type="text" value="0"/>	
B+	32	<input type="text" value="0"/>	
B-	32	<input type="text" value="0"/>	
O-	40	<input type="text" value="0"/>	
AB+	18	<input type="text" value="0"/>	
AB-	43	<input type="text" value="0"/>	
O+	10	<input type="text" value="0"/>	

Below the table, there are input fields for adding new units: a dropdown menu showing "A+" and a text input field with the value "2". A green "Add" button is located to the right of these fields. At the bottom of the page, there is a footer with the text "© 2020 - Ain Shams Hospital - Since 1980".

Figure 207 User Guide: Blood Bank View and Edit Blood Units Page