Hospital Appointment System

Requirements Specification and Analysis

1.0

10.11.2019

Alp İnce

Atanur Demir

Şükrü Taha Demirer

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc496873294)

[1.1. Purpose of the System 1](#_Toc496873295)

[1.2. Scope of the System 1](#_Toc496873296)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc496873297)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc496873298)

[1.5. Overview 1](#_Toc496873299)

[2. Current System 1](#_Toc496873300)

[3. Proposed System 1](#_Toc496873301)

[3.1. Overview 1](#_Toc496873302)

[3.2. Functional Requirements 2](#_Toc496873303)

[3.3. Nonfunctional Requirements 2](#_Toc496873304)

[Usability 2](#_Toc496873305)

[Reliability 2](#_Toc496873306)

[Performance 2](#_Toc496873307)

[Supportability 2](#_Toc496873308)

[Implementation 2](#_Toc496873309)

[Interface 2](#_Toc496873310)

[Packaging 2](#_Toc496873311)

[Legal 2](#_Toc496873312)

[3.4. System Models 2](#_Toc496873313)

[Scenarios 2](#_Toc496873314)

[Use case model 2](#_Toc496873315)

[Object model 2](#_Toc496873316)

[Dynamic model 2](#_Toc496873317)

[User interface—navigational paths and screen mock-ups 3](#_Toc496873318)

[3.5. Project Schedule 3](#_Toc496873319)

[4. Glossary 3](#_Toc496873320)

[5. References 3](#_Toc496873321)

1. **Introduction**
   1. **Purpose of the System**

The purpose of the “Hospital Appointment System” is to prevent patients from waiting for a doctor’s turn in hospitals so that, patients can spend less time in the hospital. In addition, another purpose of the system is to prevent the queuing disorder in the hospital. For doctors, the purpose is the make their job easier, and also make their work days more tidy.

**1.2 Scope of the System**

Hospital Appointment System is available for three provinces in Turkey. These provinces are Istanbul, Ankara and Izmir. People who lives in these three provinces, they can make appointment for any hospital in these provinces. Also they can see their upcoming appointments and their appointment history. In addition, doctors who works in these hospitals, they can check their schedule and they can give prescription to the patients.

**1.3 Objectives and Success Criteria of the Project**

For patients, the objectives are;

* Patients can make an appointment from the doctor that he/she wants and the hospital he/she wants.
* Patients can see their upcoming appointments and their appointment history.
* System will remind the patients their appointment day by sending an e-mail.

For doctors, the objectives are;

* To make doctors work day more tidy and efficient. They can see their schedule from the system.
* Doctors can give prescription to the patients and print these prescriptions.
* Doctors can check patient’s appointment history.

The success criteria is prevent patients from wasting their time in hospital and making doctor’s work day more organized.

**1.4 Definitions, Acronyms, and Abbreviations**

* Patient: a person who is receiving medical care, or who is cared for by a particular doctor or dentist when necessary.
* Appointment: a formal arrangement to meet or visit someone at a particular time and place
* Admin: the activities involved in managing or organizing a business or other organization.
* Doctor: a person with a medical degree whose job is to treat people who are ill or hurt
* Hospital: a place where people who are ill or injured are treated and taken care of by doctors and nurses
* Login: a name that you enter in order to be able to use a computer system.
* Register: to put information, especially your name, into an official list or record.
* Homepage: the first page of a website, which usually gives an introduction to the business or organization it belongs to and links (= connections) to more detailed information on other pages.
* Password: a secret word or combination of letters or numbers, used for communicating with another person or with a computer to prove who you are.
* Prescription: a piece of paper on which a doctor writes the details of the medicine or drugs that someone needs.
* Update: to make something more modern or suitable for use now by adding new information or changing its design.
* Diagnosis: a judgment about what a particular illness or problem is, made after examining it.
* E-mail: the system for using computers to send messages over the internet.

**1.5 Overview**

The rest of the RAD contains, functional (high-level) requirements, and nonfunctional requirements (user-level). Also, scenarios, use cases, use case diagram and object model. The last parts of the RAD are the Glossary and References parts.

**2. Current System**

The hospital system we are designing is not based on renewing the existing system. Instead, we've built an existing system with similar functions. While in the existing hospital system there are only accounts for admin, patient and visitor; we also have an extra doctor account. The system has separate and common functions for each account.

Functions in the system:

1. Visitors can view the mainpage of site, but cannot log in.
2. Visitors can become ‘Patient’ through register.
3. Visitors can become ‘Doctor’ through registerViaAdmin
4. Patients can make an appointment.
5. Patients can cancel appointments.
6. Patients can see future appointments.
7. Patients can see their own past appointments.
8. Doctors can see who made an appointment.
9. Doctors can prescribe the patient.
10. Doctors can view the medical documents of the patient.
11. Administrators can see / manage all the background tasks, such as adding doctors, system administration, checking appointments.
12. All members (Patients, Doctors, Administrators) can change their passwords, addresses, email addresses and GSM numbers.
13. All members can create a forgot password query.

**3.Proposed System**

**3.1 Overview**

**3.2 Functional Requirements**

1. The site has four different types of users: Visitors, Doctors, Patients and Administrators. Each user type has its own functions.
2. Visitors can read the news and frequently asked questions on the home page.
3. Visitors cannot login before registering. Therefore, visitors cannot access the functions used by patients or doctors.
4. Visitors can register with their ID number, password, name, age, gender, address, email and telephone information.
5. Doctors, Patients and Administrators have access to the homepage that visitors can access. Since all three classes are registered, they cannot be re-registered.
6. Doctors, Patients and Administrators must login to use their own functions. In order to login, they have to enter their identification number and password.
7. Patients can make an appointment by selecting the date, time, hospital and doctor.
8. Patients can see their appointment history. They can sort and filter the appointment history by date, hospital, department or doctor.
9. Patients can see open appointment slots within the next three months. They can sort and filter the future appointments by date, hospital, department or doctor.
10. Patients and doctors can see their personal information. They can be update their passwords, addresses, emails and telephone information.
11. Doctors, Patients and Administrators can safely logout the system. The next time they log in, they have to type their ID numbers and passwords again.
12. Patients and doctors must be able to change password if s/he forgot the password. In order to do this, they must first enter the email address in which they are registered. After that, users receive their new password by email.
13. Patients can cancel appointments.
14. Doctors can see patients who have an appointment with them. They can sort and filter appointment list by date and diagnosis.
15. Doctors can notify patients who have not come to the appointment.
16. Doctors can check the appointment history of specific patients.
17. Administrators can forward appointments that have been modified by patients to doctors. Doctors receive this notification via email.
18. Administrators can forward appointments that have been modified by doctors to patients. Patients receive this notification via email.
19. Administrators can ban patients who do not attend their appointments often.
20. Administrators can view the information of each type of registered user and change each registered information.
21. Administrators can close specific appointment slots depending on the requests of doctors.
22. Administrators can create extra appointment slots for patients.

**3.3 Nonfunctional Requirements**

**Reliability**

Multiple appointments cannot be made on the same day and time by User.

User cannot make any appointments at the times which are the doctor is not available.

The system shows a loading spinner during database processes.

**Performance**

Login process should be completed in maximum 5 seconds.

**Supportability**

The system will give a feedback message when the user successfully makes an appointment.

The system sends an e-mail to the patient in order to remind the appointment

**3.4 System Models**

**Scenarios**

|  |  |
| --- | --- |
| Scenario Name | cancelAppointmentsViaAdmin |
| Participating Actor  Instances | Taha: Admin  Atanur : Doctor  Alp : Patient |
| Flow of Events | 1. Atanur has appointments for the future. But in the last minute, a job occurred which he must do and he has no time to cancel them all. He entered the system by entering his ID number and password. After login, he clicked the red “System Admin Contact” button. 2. After clicking that button, a form occurred which contains Doctor’s name, and ID as default, and a message bar to fill. He wrote the message about his busyness. And he clicked to Green “Send” button. 3. Taha has received an email from the system, which contains date, time, doctor’s name, doctor’s ID and doctor’s message. 4. He entered the system by using his ID and password. Admin page has opened to him. He saw a new message occurred in his message box. He clicked the button “Search For Doctors”. 5. A form opened, which consists text bar “search by ID”. He filled the form with the Doctor’s ID and saw his information. 6. Taha clicked to bar which consists Doctor’s name, doctor’s ID and the Hospital name of the doctor works. A new page opened which consists a table of future appointments of the doctor. The table has a search bar for patient names, a Date set bar. 7. He set the date to the day which the doctor is busy and see the appointments of the day. He clicked red “CANCEL” button at the right of every single appointment. 8. After this, Taha clicked to the “System Message” button and filled “To” bar with doctor’s ID, wrote to message box about process is completed. 9. Alp has received an SMS and an E-Mail from the system about his appointment is canceled. |

2.

|  |  |
| --- | --- |
| Scenerio Name | registerNewDoctor |
| Participating Actor  Instances | Taha: Admin  Atanur: Doctor |
| Flow of Events | * 1. Taha has received a message about a registration of a new doctor to the system.   2. Taha has entered the system from the login page by using his ID and password. Admin page has opened.   3. He clicked the blue “New Doctor” button. Doctor Registration page has opened.   4. He filled bars with Doctor’s ID, name, GSM, E-Mail, Address, he choose doctor’s title from the comboBox. And he choose Doctor’s department and hospital from the ComboBoxes, via ID numbers of hospital and department. He filled the table of working days and hours.   5. He clicked the “Save” button. Dr. Atanur has saved to the system.   6. Dr. Atanur has received an information e-mail which consists the login information, ID and password. |
|  |  |

3.

|  |  |
| --- | --- |
| Scenerio Name | blockCriminalPatient |
| Participating Actor  Instances | Taha: Admin |
| Flow of Events | * 1. Taha has received a message from Police Department about a patient. The patient has subjected a doctor to violence. Police department has sent the ID and Name information of the criminal patient, and wants to block him 12 months from the system as a punishment.   2. Taha has entered the system from the login page by using his ID and password. Admin page opened.   3. He clicked to the red “BLOCK A USER” button. A new page opened which consists text bar of “ Search by ID”, columns of name, ID, e-mail and GSM.   4. Taha typed the ID of criminal patient and found him in the system. He clicked red “BLOCK” button at the right of row of the patient. A pop-up has occurred which has a comboBox of “Months of Duration”. He choose 12, and clicked “BLOCK” button on pop-up.   5. After clicking the button, a new pop-up has occurred. There’s “Are you sure to block this user?” text in pop-up. He clicked “YES” button. And patient blocked from making appointments for 12 months. |

4.

|  |  |
| --- | --- |
| Scenerio Name | deleteDoctor |
| Participating Actor  Instances | Taha: Admin |
| Flow of Events | 1. Taha has received a message from a Hospital, about a doctor’s quit. Hospital wants to delete the doctor from the Hospital records. 2. Taha opened the login page and entered the system by using ID and password. He clicked the “Hospitals” page button. A new page opened. 3. He clicked the search bar and typed the name of hospital. A table of persons registered to that hospital opened. 4. He found the doctor’s name who has quitted from job. He clicked his name and the information page of that doctor is opened. 5. He has no information about the doctor’s new hospital. Thus, he clicked the “Change Information” button, after that, “Hospital” comboBox of doctor, and turned it into “not known”. “Working Hours” table automatically turned to gray, not able to change. And he clicked to save button. |

5.

|  |  |
| --- | --- |
| Scenerio Name | transferDoctor |
| Participating Actor  Instances | Taha: Admin |
| Flow of Events | 1. Taha has received a message about a doctor is changing the hospital he works, from the hospital. That information have to be changed. He opened the login page and entered the system by using his ID and password. 2. He clicked “Doctors” button, a new page consisting table of doctors opened. He typed doctor’s name to search bar. And found the doctor’s name. 3. There’s two doctors with the same name, he compared the ID’s, and found the correct one. He clicked his name. Doctor’s informations page opened. 4. Taha clicked to “Change Information” button. After, he clicked to “Hospital” comboBox of doctor. He changed hospital information to the actual one. Working address statement automatically filled to the new hospital’s address. He clicked the save button. |

6.

|  |  |
| --- | --- |
| Scenerio Name | registerANewHospital |
| Participating Actor  Instances | Taha: Admin |
| Flow of Events | 1. Taha has received an email consists documents of permission and confirmation from Ministry of Health. They want to register this new hospital to the system. 2. He opened the login page and entered the system by using his ID and password. He clicked to “Register a Hospital” button. A new page opened. 3. He filled the blank fields with “Hospital’s Name”, Address, Phone, Departments, Number of Rooms, Number of Beds, Number of Doctors, Working Hours. 4. He clicked the save button and new hospital has saved to the system. |

7.

|  |  |
| --- | --- |
| Scenerio Name | deleteHospital |
| Participating Actor  Instances | Taha: Admin |
| Flow of Events | 1. Taha has received a message from Ministry of Health, about a closing hospital. Ministry wants to delete this hospital from the system. 2. Taha opened the login page and entered the system by using his ID and password. Admin page opened. He clicked to “Hospitals” button. 3. A table of hospitals opened. He typed the name of hospital to search bar, found the hospital, clicked to the name. Hospital’s Information page opened. 4. He clicked to little red “Delete Hospital” button. A pop-up occurred which consists “Are you sure want to delete this hospital? This may have consequences.” Sentence. He clicked to “YES” button. Hospital is deleted from the system. |

8.

|  |  |
| --- | --- |
| Scenerio Name | viewTakenAppointments |
| Participating Actor  Instances | Atanur: Doctor |
| Flow of Events | 1. Atanur. wants to see patients who have an appointment with him. Firstly, he login the system by entering his identification number and password. After that, he clicks on the “show taken appointments” button on the homepage. 2. After the page opens, he can see the list of appointments made on the opened page. In the menu that opens, the doctor can see the appointments received from him day by day. Whichever day he wants to view, he must click the day's button. 3. These appointments are sorted by date and contain information including patients' names, genders and ages. 4. After receiving the required documents from on “appointment list” page, If the doctor wants, he can turn off (logout button) the system with a single button or go back to the main menu. (back button) |

9.

|  |  |
| --- | --- |
| Scenerio Name | reportNotParticipate |
| Participating Actor  Instances | Atanur: Doctor |
| Flow of Events | 1. In order to use this operation, a doctor must be login to system with his/her id number and password. 2. Therefore, Atanur login to system to use ‘notify absent patient to Administrator’ function. Following to login, he go to the ‘Absent Patient’ page. 3. On this page, he can be see the future appointments made by him and if he wishes, can fill the little box on patient names. Filled boxes means that patients did not come to the appointment. 4. Atanur presses the "send" button after identifying patients who do not come. 5. In this way, all missing patients are reported to their admissions in one go. |

10.

|  |  |
| --- | --- |
| Scenerio Name | searchForPatient |
| Participating Actor  Instances | Atanur: Doctor |
| Flow of Events | 1. In order to use this operation, a doctor must be login to system with his/her id number and password. 2. Therefore, Atanur login to system to use ‘information of a specific patient’ function. Following to login, he go to the ‘Search Patient’ page. 3. To next page, Atanur must be write the name and surname of the patient that he wants to investigate. 4. After entering the name and surname data, patient names that meet these criteria are presented to the user in tabular form. 5. If there are multiple patients with the same name, Atanur should be check patients ‘identification number’ on this page to find spesific names that he wants. 6. Once Atanur find the patient that he wants, just clicks on his/her names. 7. Following that, Atanur accesses his patient’s previous appointment records. These registrations can be seen from which departments and how often appointments were made. |

11.

|  |  |
| --- | --- |
| Scenerio Name | givePescription |
| Participating Actor  Instances | Atanur: Doctor |
| Flow of Events | 1. Atanur wants to prescribe his patient. 2. In order to do this, he enters the system with his id and password. 3. On the following main screen, Atanur press the "prescribe" button to go to the page where the recipe is written. 4. The recipe page comes in front of the doctor in the form of a blank list. If the doctor wishes to do so, he will write the list of drugs he wishes to give. 5. After he have written all the medicines, press the "give" button to finish the prescription. 6. On the next page, the prescription appears in front of the doctor, both as a written with list, barcode and as a random generated code. 7. The prescription given by the doctor is saved in the system. 8. If the doctor wants, he can turn off (logout button) the system with a single button or go back to the main menu. (back button) |

12.

|  |  |
| --- | --- |
| Scenerio Name | printPrescription |
| Participating Actor  Instances | Atanur: Doctor |
| Flow of Events | 1. In order to perform this process, the prescription must be given first. 2. After the function of “givePescription”, Atanur can print this recipe with the help of a printer. 3. First the doctor gives his patient a prescription. Prescription page both as a written with list, barcode and as a random generated code. 4. The doctor can print all this data on the paper using the “print” button on this page. 5. If no usable printer is defined, the doctor receives an error message. 6. If the procedure was successful, the doctor will receive a message that the procedure was successful. 7. If the doctor wants, he can turn off (logout button) the system with a single button or go back to the main menu. (back button) |

13.

|  |  |
| --- | --- |
| Scenerio Name | registerViaAdmin |
| Participating Actor  Instances | Atanur: Doctor |
| Flow of Events | 1. Atanur, in order to become a doctor member, he must contact to admin. 2. To do this, Atanur should use the function of “contact with the admin.” on the main page. 3. On the next page, he should fill in his ID number and his mail number for contact the administrator of the website. 4. Administrators evaluate this message and check whether the person is a doctor or not. 5. If the person who sent this message is actually a doctor, admin will send the relevant user's ID and password to the registered e-mail address. |

14.

|  |  |
| --- | --- |
| Scenario name | registerForAppointment |
| Participating Actor  Instances | alp: Patient |
| Flow of events | 1. Alp, wants to make an appointment for his illness and that’s why he enters the hospital appointment website. Then he clicks the “register button”. 2. After register page opened, he fills the name, surname, id, age, gender, address, e-mail, phone number and password fields. Then he clicks the register button. 3. After the clicking “register” button, login page opened. After the fills the login informations, he can make an appointment for his treatment. |

15.

|  |  |
| --- | --- |
| Scenario name | loginForAppointment |
| Participating actor  instances | Alp:Patient |
| Flow of events | 1. 1. Alp, he is a registered user and he wants to make an appointment for his illness. Then, he enters the website. 2. In the login screen, he needs to fill the, his ID and password areas. After the Alp has entered his informations correctly, he clicks the “login” button and complete login operation. |

16.

|  |  |
| --- | --- |
| Scenario name | forgotPassword |
| Participating actor  instances | Alp:Patient |
| Flow of events | 1. Alp, he is a registered user and he wants to login to the system but he can’t remember his password. That’s why he clicks the “forgot password” button. 2. In the forgot password page, he fills the his id and e-mail and clicks the “send” button. Then he receives an e-mail that contains a link for reset password.      1. He clicks the link, then a new page opened. In this page he needs the fill new password and new password again areas. 2. After filling this areas, he clicks the “save” button then, login page opens. Thus, he successfully change his password. |

17.

|  |  |
| --- | --- |
| Scenario name | MakingAppointment |
| Participating actor  Instances | Alp:Patient |
| Flow of events | 1. Alp, wants to make an appointment for his illness and that’s why he enters the hospital appointment website. After login process, he enters the appointment page. 2. He enters the appointment informations like date, province, district, clinic, hospital, neighbourhood polyclinic, examination location and doctor. After this part completed, clicks the “search” button. 3. After the clicking “search appointment” button, he sees the avalible times for doctor. He choose the time and cliks the “save appointment” and he successfully made an appointment. |

18.

|  |  |
| --- | --- |
| Scenario name | changingPersonalInformations |
| Participating actor  Instances | Alp:Patient |
| Flow of events | 1. Alp, wants to check his personal informations and if there is a wrong information he wants to change it. Firstly, he login to the system. 2. After the login operation, he clicks the “Personal Information” button and pop-up screen is opened. Alp, can see his personal informations like name, surname, gender, age, address, e-mail, phone number. 3. After checking his personal informations, he realized that his phone number is wrong. Then he corrects the phone number and clicks the “save” button 4. After saving his contract details he clicks the “cross” button and close the personal informations pop-up screen. |

19.

|  |  |
| --- | --- |
| Scenario name | seeingFutureAppointments |
| Participating actor  Instances | Alp:Patient |
| Flow of events | 1. Alp had made multiple appointments and he can’t remember appointment dates. So, he login to the system and in the appointment page he clicks the “Future Appointments” button. 2. Then new pop-up screen opened, and he can check all the future appointments in this pop-up screen. But he realized that, his appointment list is too long and he can’t find one specific appointment. 3. So, he sorted the appointment list for date. Thus, he found the appointment that he wants. |

20.

|  |  |
| --- | --- |
| Scenario name | seeingAppointmentHistory |
| Participating actor  instances | Alp:Patient |
| Flow of events | 1. Alp wants to check his appointment history in order to see his last appointment date. So he login to the system and in the appointment page he clicks the “Appointment History” button. 2. Then new pop-up screen opened, and he can check all the past appointments from this pop-up screen. But he realized that appointment history list is too long. 3. So, to find a specific one appointment he filtered the appointment list by polyclinic. Thus he found the last appointment that he wants. |

21.

|  |  |
| --- | --- |
| Scenario name | changingPassword |
| Participating actor  instances | Alp:RegisteredUser |
| Flow of events | 1. Alp, he is a registered user and he wants to change his password. He wants to make his password safer. 2. That’s why he login to the system and clicks the “Personal Informations” button. Then, he clicks the “change password” button in the personal informations pop-up screen. 3. Then a new pop-up screen opened and he needs to enter his old password and new password twice. After fills the blanks he clicks the “save” button and successfully completed changing password. |

22.

|  |  |
| --- | --- |
| Scenario name | cancelAppointment |
| Participating actor  instances | Alp:Patient |
| Flow of events | 1. Alp, he is a registered user and he had made an appointment. But he realized that he can’t go to this appointment. Thus, he wants to cancel his appointment 2. After login process, he enters the appointment page and clicks “future appointments” button. And he sees the all coming appointments. 3. Then in order to cancel his appointment, he clicks to “delete” button next to the appointment. So he successfully canceled his appointment |

**Usecase Model**

1.

|  |  |
| --- | --- |
| Usecase Name | TakenAppointments |
| Participating Actor | Initiated by Doctor |
| Flow of Events | 1. Doctor login the site successfully with her/his id number and password. 2. Doctor uses the “view future appointments” function on the main menu.    * 1. Appointment system responds by opening a new page.      2. The newly opened page contains columns with days. 3. Doctor clicks on the day what s/he want to see.    * 1. Appointment system respond by opening a new sub-column.      2. In this column, a table with the patient's name, age and attend hour information appears.      3. This list of patients is sorted by appointment time. 4. Doctor can change between days until s/he press the "back" key.    * 1. Clicking each day of the doctor closes the previous page and opens the information for the new day. |
| Entry Condition | Doctor must login into system successfully.  Doctor must go the page where the viewing Appointment page will be procedure. |
| Exit Condition | 1. Doctor press one of the "back" or "logout" buttons on the corresponding page. 2. If the doctor presses “back”, it will return to the main menu. If the doctor presses “logout”, s/he will logout the system. Either way, it terminates the process. |
| Quality Requirements | 1. The doctor should be able to see the appointments correctly. 2. The doctor should be able to see the list of patients who have made appointments day by day and should not wait during the transition between days. 3. Listing the appointments processes should not take more than five seconds. 4. When the doctor presses back or exit, he should receive a message that the procedure is complete. |

2.

|  |  |
| --- | --- |
| Usecase Name | ReportPatient |
| Participating Actor | Initiated by Doctor |
| Flow of Events | 1. Doctor login the site successfully with her/his id number and password. 2. Doctor uses the “report patient” function on the main menu.    * 1. Reporting system responds by opening a new page.      2. The newly opened page contains a table with a list of patients for that day.      3. There are small boxes next to these patients. 3. Doctor checks the boxes next to the patients who did not come to the hospital that day. 4. After marking all patients, the doctor presses the send button.    * 1. The system sends the marked patients to the management of the site.      2. The system notifies the doctor that the operation has been performed successfully. |
| Entry Condition | Doctor must login into system successfully.  Doctor must go to the Reporting page via the main menu. |
| Exit Condition | 1. After the reporting process is complete, Doctor can press either on of the “back” or “logout” buttons on the corresponding page. 2. If the doctor presses “back”, it will return to the main menu. If the doctor presses “logout”, s/he will logout the system. Either way, it terminates the process. |
| Quality Requirements | 1. The doctor should be able to successfully report the patients s/he wants. 2. After reporting, doctor should see a message that the transaction was successful. 3. Reporting processes should not take more than five seconds. 4. When the doctor presses back or exit, he should receive a message that the procedure is complete. |

3.

|  |  |
| --- | --- |
| Usecase Name | SearchPatient |
| Participating Actor | Initiated by Doctor |
| Flow of Events | 1. Doctor login the site successfully with her/his id number and password. 2. Doctor uses the “search patient” function on main menu.    * 1. Searching system responds by opening a new page.      2. The newly opened page contains an empty box for typing wanted patient’s ID. 3. The doctor writes the id number of the desired patient in the empty box and clicks the “search” button.    * 1. The system presents patients to the doctor as a list, depending on the criteria sought. 4. If the doctor wishes, he can search the box again by typing the information of the patient he is calling. 5. Once the doctor has finished the search, s/he can turn the search function off by pressing the back button. |
| Entry Condition | Doctor must login into system successfully.  Doctor must go to the Search page via the main menu. |
| Exit Condition | 1. Doctor press one of the "back" or "logout" buttons on the corresponding page. 2. If the doctor presses “back”, it will return to the main menu. If the doctor presses “logout”, s/he will logout the system. Either way, it terminates the process. |
| Quality Requirements | 1. The doctor should be able to search his/her patients and patients should be displayed with the correct information. 2. The doctor should see the previous appointments of the patient correctly. 3. Searching processes should not take more than five seconds. 4. When the doctor presses back or exit, he should receive a message that the procedure is complete. |

4.

|  |  |
| --- | --- |
| Usecase Name | GivePrescription |
| Participating Actor | Initiated by Doctor |
| Flow of Events | 1. Doctor login the site successfully with her/his id number and password. 2. Doctor uses the “Prescription” function on main menu.    * 1. Prescription system responds by opening a new page.      2. On the new page, there is a space to write the id information of the patient whose medication is intended to be written.      3. At the bottom of this page contains an empty list to fill in with drug names. 3. On this page, doctor enters the id information of the patient s/he wants to write medication. 4. After that, doctor enters the empty list of the drugs s/he wants. 5. Once the prescription is complete, doctor clicks the send button.    * 1. Prescription system notifies the doctor that the operation has been performed successfully.      2. On the notification page, there are “back” and “print” buttons. 6. If the doctor wishes to return to the main menu with the back key.    * 1. The system returns to the main doctor menu. 7. If the doctor wishes to print the prescription, s/he need to press “print” button on the page.    * 1. The system prints the entered recipe with the help of a printer. |
| Entry Condition | Doctor must login into system successfully.  Doctor must go to the Prescription page via the main menu. |
| Exit Condition | 1. Doctor press one of the "back" or "logout" buttons on the corresponding page. 2. If the doctor presses “back”, it will return to the main menu. If the doctor presses “logout”, s/he will logout the system. Either way, it terminates the process. 3. If the print button is used even though there is no printer defined on the computer, the system will fail and terminates the processes. |
| Quality Requirements | 1. The doctor's prescription should be sent to the system correctly. 2. The doctor should be able to see his/her prescription both as a written with list, barcode and as a random generated code. 3. Prescription (sending) processes should not take more than five seconds. 4. When the doctor presses back or exit, he should receive a message that the procedure is complete. |

5.

|  |  |
| --- | --- |
| Usecase Name | RegisterViaAdmin |
| Participating Actor | Initiated by Visitor, Admin;  Informed Admin, Visitor |
| Flow of Events | 1. Visitor comes to the home page of the site and clicks on the contact section.    * 1. Main site responds by opening the “contact” page.      2. There is a blank box on the contact page where visitors can send their messages. In this box, they can write any message they want. 2. Visitor writes the ID number and email address in this box. 3. After visitor enters the message s/he wants to write, s/he clicks the send button.    * 1. The system transmits the message written by the user to admins.      2. Admin displays this message on its home page. If the person who sent the message is a doctor, Admin can create a new doctor account for her/him and send account information to his/her mail account. 4. Visitor receives a message from admin to her/his email address. This message contains the ID and password of the new doctor account. 5. Visitor can now log in with a Doctor account. |
| Entry Condition | Doctor or Visitor must go to the Contact to Administrator page via the main menu. |
| Exit Condition | 1. Doctor press one of the "back" or "logout" buttons on the corresponding page. 2. If the doctor presses “back”, it will return to the main menu. If the doctor presses “logout”, s/he will logout the system. Either way, it terminates the process. |
| Quality Requirements | 1. Visitor must receive notification that s/he has successfully sent the message. 2. If visitor is not a doctor, this should be stated in the incoming mail. 3. Visitor must be able to send a 500-words message. 4. Admin should be able to display the message received from the user correctly. |

6.

|  |  |
| --- | --- |
| Usecase Name | RegisterDoctor |
| Participating Actor | Initiated by Admin, Informed Doctor |
| Flow of Events | 1. Admin enters the website and activates the “Register a Doctor” function.    * 1. Appointment System responds by opening a form. 2. Admin fills out form by typing Doctor’s ID, Name, GSM, E-Mail, selecting title, department, hospital, working days, working hours. Once the form is completed, Admin submits the form.    * 1. Doctor receives an e-mail contains login information. |
| Entry Condition | Admin has logged into Appointment System. |
| Exit Condition | Registration completed successfully. |
| Quality Requirements | 1. Registration should not take more than 5 seconds. 2. Doctor should receive a unique password. |

7.

|  |  |
| --- | --- |
| Usecase Name | BlockUser |
| Participating Actor | Initiated by Admin, Informed Patient |
| Flow of Events | 1. Admin activates the “Block a User” function from his terminal.    * 1. Appointment System responds by representing a list of users. 2. Admin finds the user, clicks to “Ban” button. Also set the duration of ban. Once the adjustment completed, Admin submits the ban form.    * 1. Appointment System displays the Ban Report, sends an e-mail to banned user. |
| Entry Condition | Admin logged into system.  Patient registered to system. |
| Exit Condition | Banning process completed successfully. |
| Quality Requirements | 1. Patient should receive e-mail. 2. Ban report representing should not take more than 5 seconds. |

8.

|  |  |
| --- | --- |
| Usecase Name | DeleteDoctor |
| Participating Actor | Initiated by Admin |
| Flow of Events | 1. Admin activates the “Search For Doctor” function of his terminal.    * 1. Appointment System responds by presenting a list of doctors to the Admin. 2. Admin types the ID of the doctor to the search bar.    * 1. Appointment System responds by presenting the row of doctor. 3. Admin clicks to the name of doctor.    * 1. Appointment System responds by presenting the information page of doctor. 4. Admin clicks to “Delete” button.    * 1. Appointment System responds by presenting a pop-up of check. 5. Admin submits the delete process.    * 1. Appointment System displays a pop-up of successfullness. |
| Entry Condition | Admin is logged into the system.  Doctor is registered into system. |
| Exit Condition | Deleting process completed successfully. |
| Quality Requirements | 1. Delete process should be completed less than 5 seconds. |

9.

|  |  |
| --- | --- |
| Usecase Name | Doctor Transfer |
| Participating Actor | Initiated by Admin, Informed Doctor |
| Flow of Events | 1. Admin activates the “Search For Doctor” function of his terminal.    * 1. Appointment System responds by presenting a list of doctors to the Admin. 2. Admin types the ID of the Doctor to the search bar.    * 1. Appointment System responds by presenting the row of doctor. 3. Admin clicks to the name of Doctor.    * 1. Appointment System responds by presenting the information page of Doctor. 4. Admin changes the hospital of Doctor, submits the form to system.    * 1. Appointment System responds by presenting a confirmation pop-up, sends an e-mail to the Doctor. |
| Entry Condition | Admin is logged into Appointment System.  Doctor has registered into the system. |
| Exit Condition | Changes successfully confirmed by the system. |
| Quality Requirements | 1. Doctor should receive e-mail. 2. System should response less than 5 seconds. |

10.

|  |  |
| --- | --- |
| Usecase Name | RegisterHospital |
| Participating Actor | Initiated by Admin |
| Flow of Events | 1. Admin activates the “Register Hospital” function of his terminal.    * 1. Appointment System responds by presenting a form to the Admin. 2. Admin fills out the form by typing Hospital’s Name, Address, Phone, selecting City, Street, Postal Number, Number of doctors, rooms and beds, working hours. Once the form is completed, the Admin submits the form.    * 1. Appointment System receives the form and notifies the Admin. |
| Entry Condition | Admin is logged into Appointment System. |
| Exit Condition | Registration completed successfully. |
| Quality Requirements | 1. System should save new hospital less than 5 seconds. |

11.

|  |  |
| --- | --- |
| Usecase Name | DeleteHospital |
| Participating Actor | Initiated by Admin |
| Flow of Events | 1. Admin activates the “Search For Hospital” function of his terminal.    * 1. Appointment System responds by presenting a list of hospitals to the Admin. 2. Admin types the ID of the hospital to the search bar.    * 1. Appointment System responds by presenting the row of hospital. 3. Admin clicks to the name of hospital.    * 1. Appointment System responds by presenting the information page of hospital. 4. Admin clicks to “Delete” button.    * 1. Appointment System responds by presenting a pop-up of check. 5. Admin submits the delete process.    * 1. Appointment System displays a pop-up of successfullness. |
| Entry Condition | Admin is logged into Appointment System.  Hospital has registered into the system. |
| Exit Condition | Deleting successfully confirmed by the system. |
| Quality Requirements | 1. System should response less than 5 seconds. |

12.

|  |  |
| --- | --- |
| Usecase Name | AdminAppointmentCancel |
| Participating Actor | Initiated by Admin, Communicated with Doctor, Informed Patient |
| Flow of Events | 1. Doctor activates the “System Message” function of his terminal. 2. Doctor fills out the form by entering date, hour and message. Once the form is completed, Doctor submits the form.    * 1. Appointment System receives the form, and notifies Admin. 3. Admin reviews the submitted message, and activates “Search For Doctors” function of his terminal. 4. Admin finds the doctor, clicks to cancel necessary appointments. Once the cancelation is completed, Admin submits the appointment calendar.    * 1. Appointment System receives the cancelations, and notifies Doctor and Patient. |
| Entry Condition | Admin and Doctor has logged into Appointment System. |
| Exit Condition | Cancelation completed successfully. |
| Quality Requirements | 1. Patient should receive SMS. 2. Admin should response less than 1 hour. |

13.

|  |  |
| --- | --- |
| Usecase Name | Login |
| Participating Actor | Initiated by Patient or Doctor |
| Flow of Events | 1. Patient or doctor enters the hospital appointment website and fills out the form by typing id and password. And clicks the “login” button. Wrong id and password can lead the system to prompt message regarding to the error.    * 1. If patient or doctor can’t remember his/hers password, clicks the “forgot password” button. Then “forgot password” use case will be activated.      2. If user is patient and login information is correct, system redirects to the appointment page. If the user is doctor and login information is correct, system redirects to the doctor control page. |
| Entry Condition | Patient or doctor enters the hospital appointment website. |
| Exit Condition | Login process completed successfully. |
| Quality Requirements | 1. Login authentication should not take more than 5 seconds. 2. If login informations are wrong, system should prompt error message in 3 seconds. |

14.

|  |  |
| --- | --- |
| Usecase Name | Register |
| Participating Actor | Initiated by Patient or Doctor |
| Flow of Events | 1. Patient or doctor enters the hospital appointment website and clicks the “Register” button.    * 1. System redirects patient/doctor to the register page. 2. Patient/doctor fills out the form by entering name, surname, id, age, gender, address, e-mail, phone number and password. Then clicks the “register” button.    * 1. If there is an unfilled area system should prompt an error. Otherwise registration is successful and system redirects patient or doctor to the login page. |
| Entry Condition | Patient/doctor clicks the register button in the main page. |
| Exit Condition | Patient/doctor successfully register to the appointment system. |
| Quality Requirements | 1. Register authentication should not take more than 5 seconds. 2. If there is an unfilled area, system should give error message in 3 seconds after clicking the register button. |

15.

|  |  |
| --- | --- |
| Usecase Name | ForgotPassword |
| Participating Actor | Initiated by Patient |
| Flow of Events | 1. Patient enters the hospital appointment website and clicks the “forgot password” button.    * 1. System redirects patient/doctor to the “forgot password” page. 2. Patient fills out the form by entering his/hers id and e-mail and clicks “send” button.    * 1. System sends an e-mail to the user that contains a link for reset password. 3. Patient clicks to the link and then enters his/hers password twice. Then clicks to the “save” button. |
| Entry Condition | Patient clicks to the “forgot password” button. |
| Exit Condition | Patient successfully chances password. |
| Quality Requirements | 1. If there is an unfilled area, system should prompt an error after clicking “send” and “save” button. 2. System should send an e-mail in a minute. |

16.

|  |  |
| --- | --- |
| Usecase Name | MakingAppointment |
| Participating Actor | Initiated by Patient |
| Flow of Events | 1. After login process, patients enters to the hospital appointment system. 2. The patient fill out the form by selecting date, province, district, hospital, neighbourhood polyclinic, examination location and doctor. After this part completed, patient clicks the “search” button.    * 1. System shows the avalible times for selected doctor. 3. The patient selects the time he/she wants and clicks the “save” button.    * 1. System sends an e-mail that contains appointment details to the patient. |
| Entry Condition | The patient is logged into hospital appointment system. |
| Exit Condition | The patient, successfully made an appointment |
| Quality Requirements | 1. System should send an e-mail in a minute. |

17.

|  |  |
| --- | --- |
| Usecase Name | ChangingPersonalInformations |
| Participating Actor | Initiated by Patient |
| Flow of Events | 1. After login process, the patient clicks the “personal information” button in order to see his/hers personal information or change some specific informations.    * 1. System shows personal informations in a new pop-up screen.      2. System allows to change only address, e-mail and phone number. 2. After changing personal informations, patient clicks the “save” button.    * 1. System successfully change informations and save them. |
| Entry Condition | The Patient is logged into Hospital Appointment System and clicks to the “personal informations” button. |
| Exit Condition | The patient successfully change his/hers personal informations. |
| Quality Requirements | 1. System does not allow to change some informations like id, name, surname, age and gender. |

18.

|  |  |
| --- | --- |
| Usecase Name | ViewFutureAppointments |
| Participating Actor | Initiated by Patient |
| Flow of Events | 1. After login process, patient clicks the “Future Appointments” button    * 1. System shows upcoming appointments in a new pop-up screen. And lists all the future appointments. 2. If patient wants he/she can filter the list by date, polyclinic or doctor. |
| Entry Condition | The Patient is logged into Hospital Appointment System and clicks to the “Future Appointments” button. |
| Exit Condition | The patient successfully see his/hers upcoming appointments. |
| Quality Requirements | 1. Listing the appointments should not take 5 seconds. |

19.

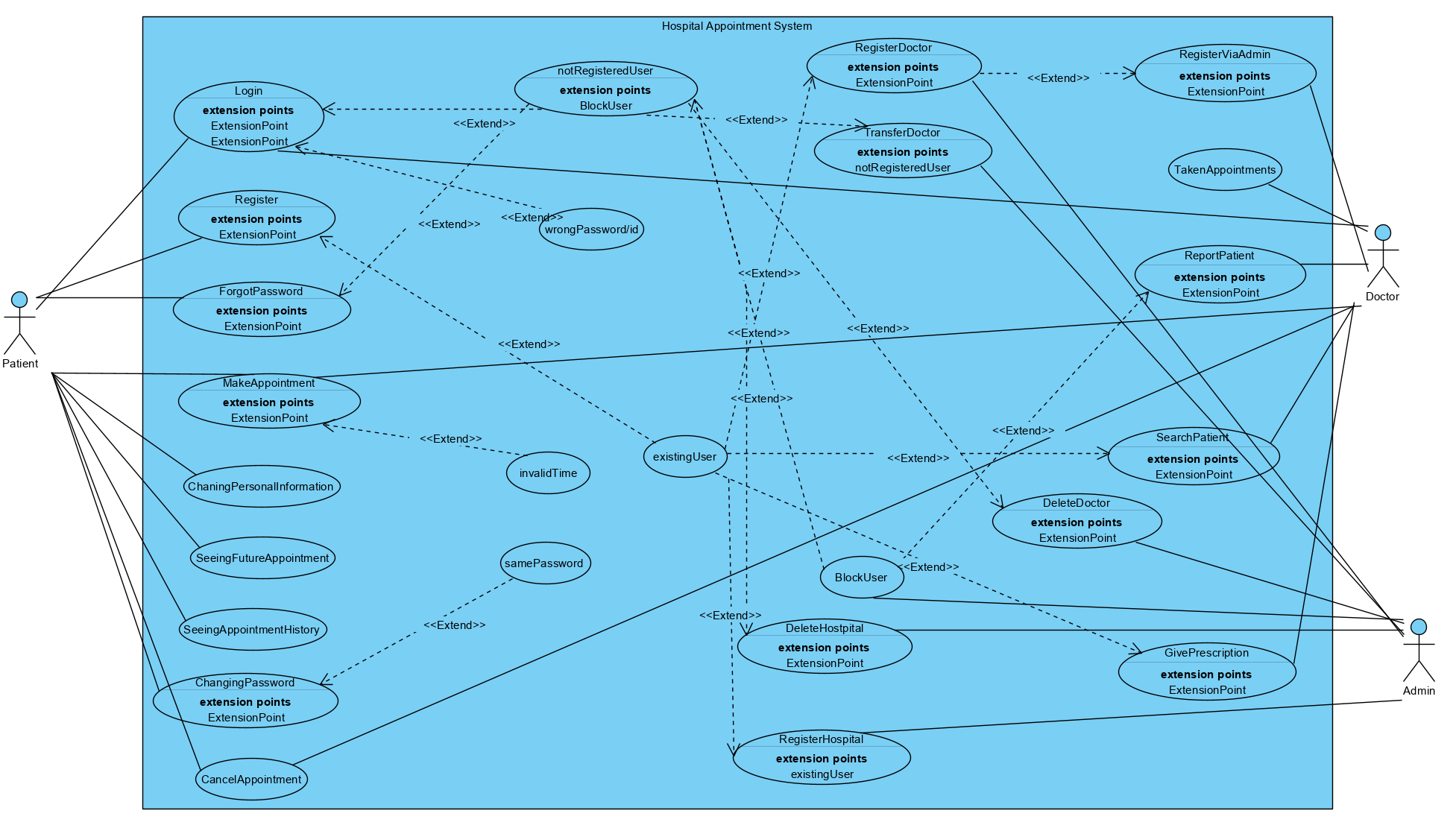
|  |  |
| --- | --- |
| Usecase Name | ViewAppointmentHistory |
| Participating Actor | Initiated by Patient |
| Flow of Events | 1. After login process, patient clicks the “Appointment History” button    * 1. System shows past appointments in a new pop-up screen. And lists all the past appointments. 2. If patient wants he/she can filter the list by date, polyclinic or doctor. |
| Entry Condition | The Patient is logged into Hospital Appointment System and clicks to the “Appointment History” button. |
| Exit Condition | The patient successfully see his/hers past appointments. |
| Quality Requirements | 1. Listing the appointments should not take 5 seconds. |

20.

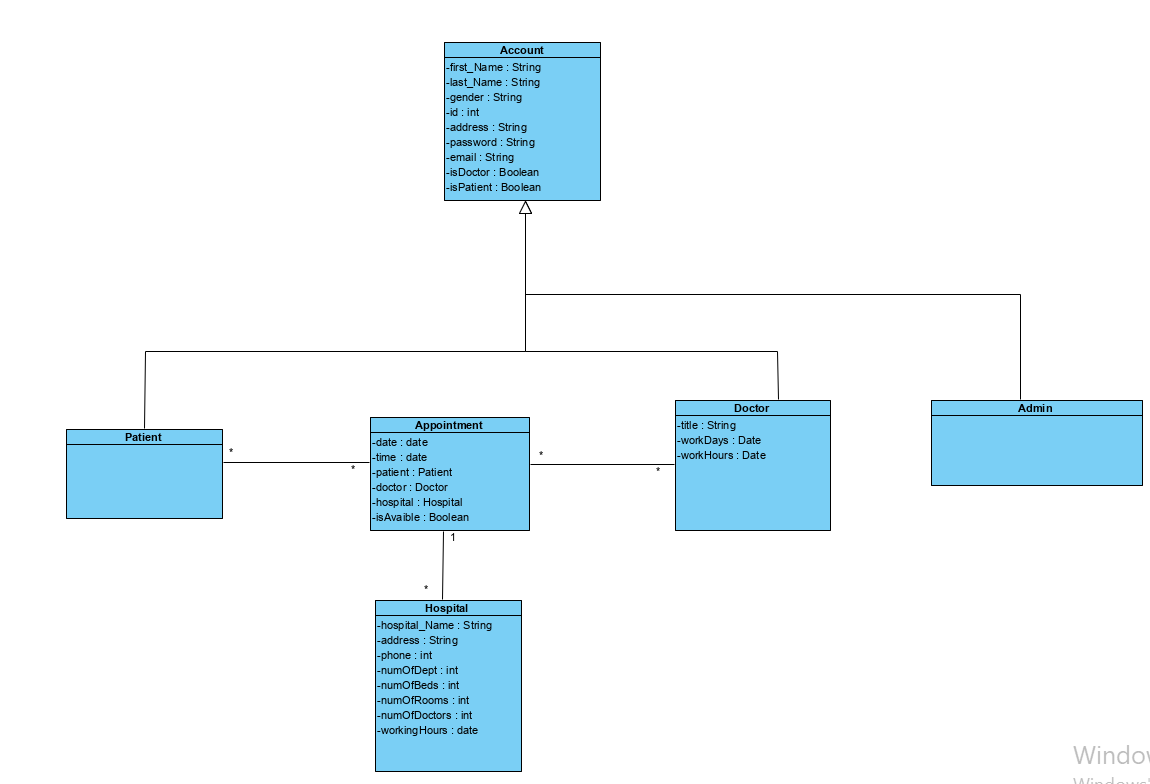
|  |  |
| --- | --- |
| Usecase Name | ChangingPassword |
| Participating Actor | Initiated by Patient or Doctor |
| Flow of Events | 1. After login process, patient/user clicks the “personal information” button.    * 1. System opens the personal informations pop-up screen. 2. Then patient/doctor clicks to the “change password” button.    * 1. System opens a new page in personal informations pop-up page. 3. Patient/doctor enters his/hers new password twice and clicks “save” button and successfully completed changing password. |
| Entry Condition | The Patient/doctor is logged into Hospital Appointment System and clicks to the “personal informations” and then “change password” buttons. |
| Exit Condition | Patient/doctor successfully completed the changing password. |
| Quality Requirements |  |

21.

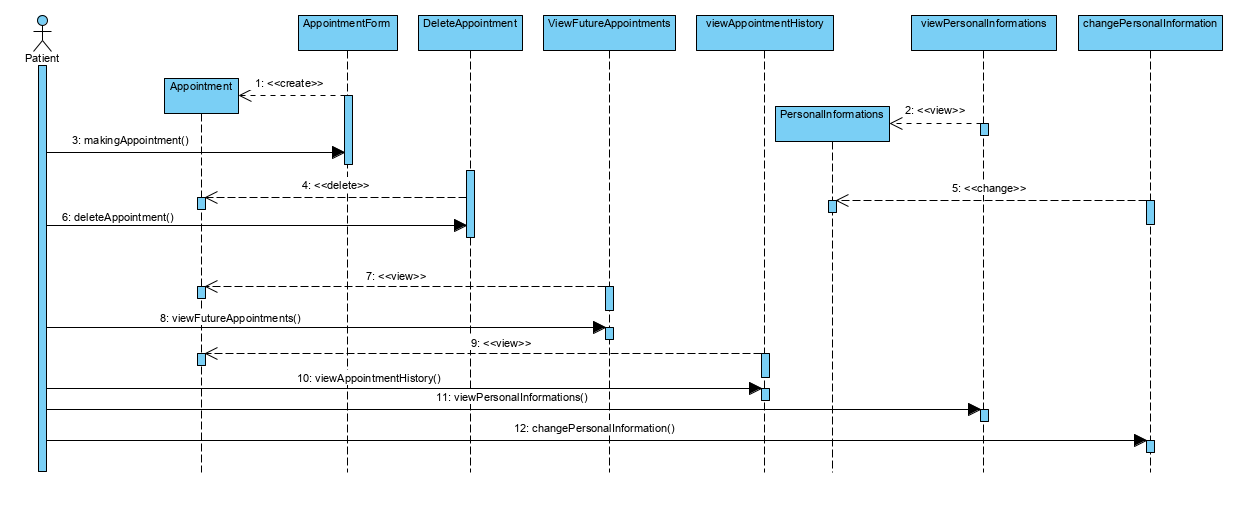
|  |  |
| --- | --- |
| Usecase Name | CancelAppointment |
| Participating Actor | Initiated by Patient |
| Flow of Events | 1. After login process, patient clicks the “Future Appointments” button    * 1. System shows upcoming appointments in a new pop-up screen. And lists all the future appointments. 2. If patient wants cancel his/hers upcoming appointment clicks to “delete” button next to appointment. |
| Entry Condition | The Patient is logged into Hospital Appointment System and clicks to the “Future Appointments” button. |
| Exit Condition | Patient successfully cancel his/hers appointment |
| Quality Requirements | 1. System should cancel appointment in 3 seconds . |

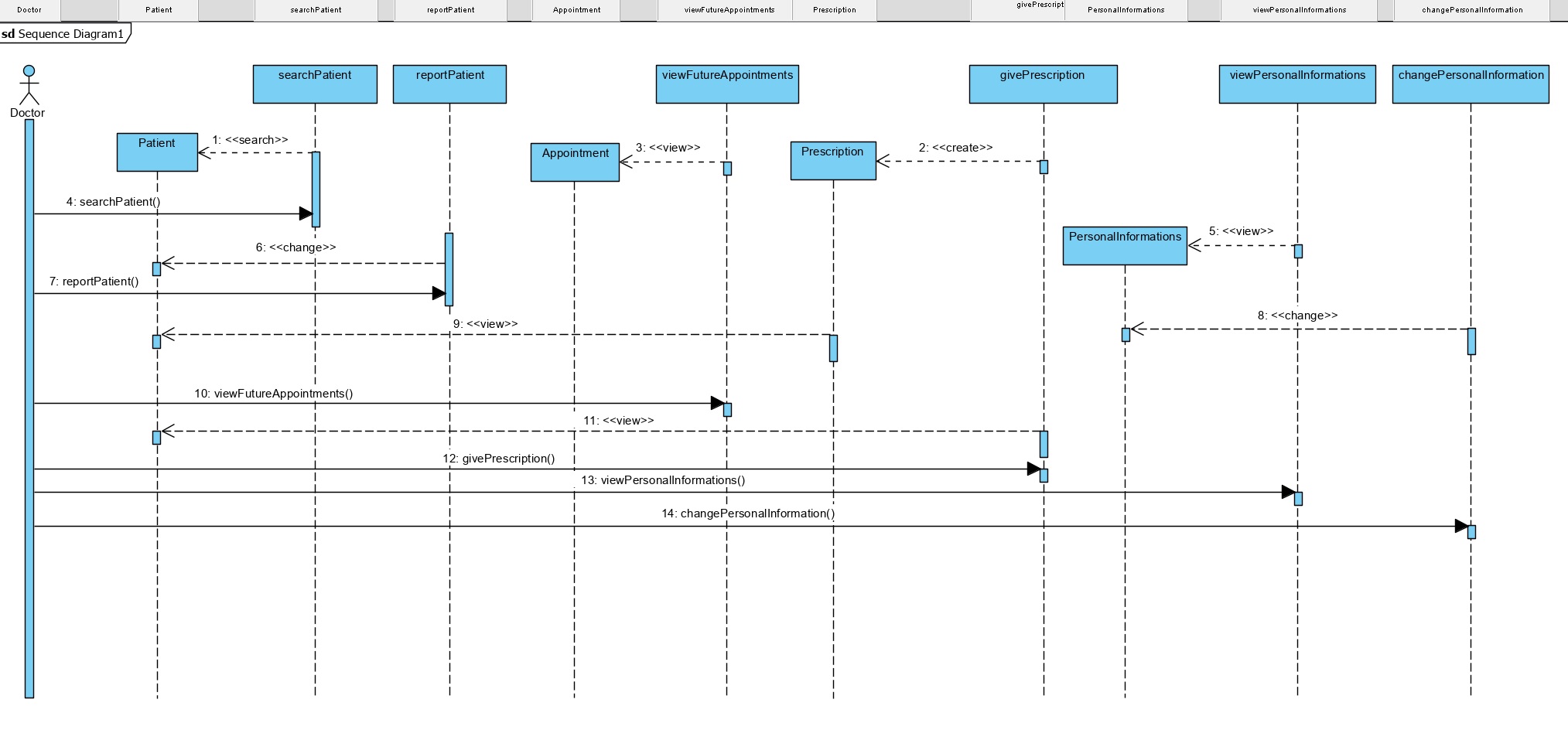
****

**Object Model**

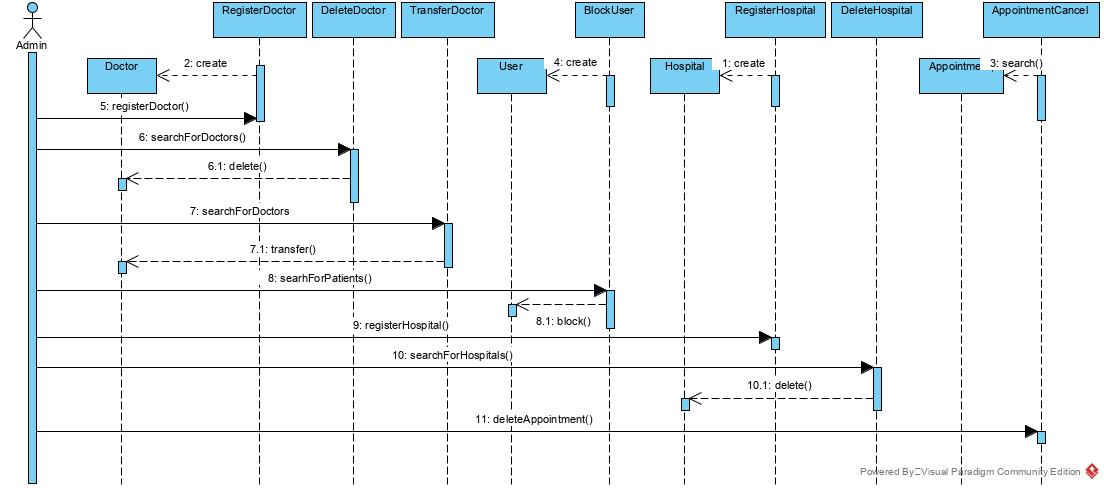


**Dynamic Model for Patient**

**Dynamic Model for Doctor**

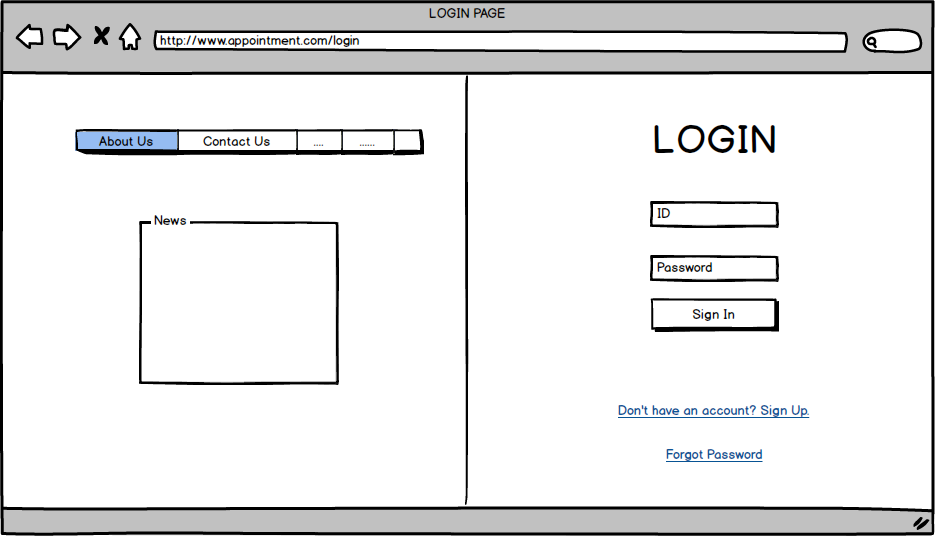


**Dynamic Model for Administrator**

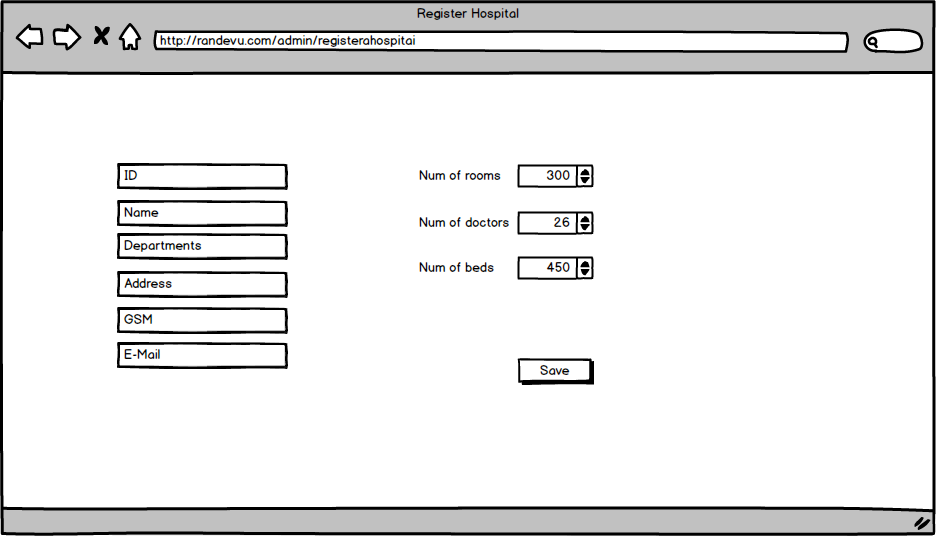
****

**Mockups**

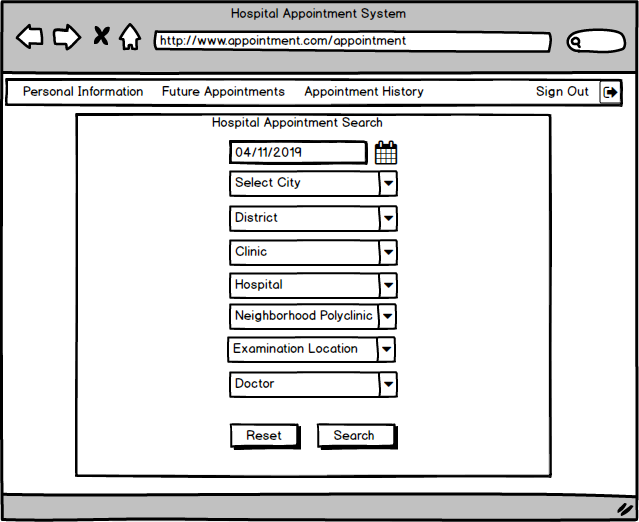
Login Mockup

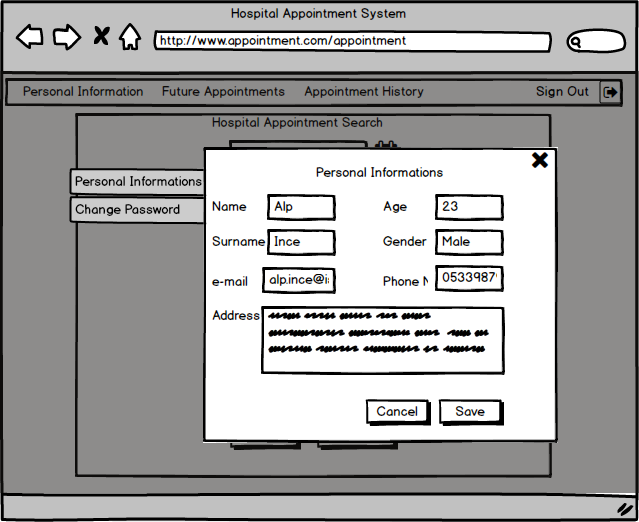


Register Mockup



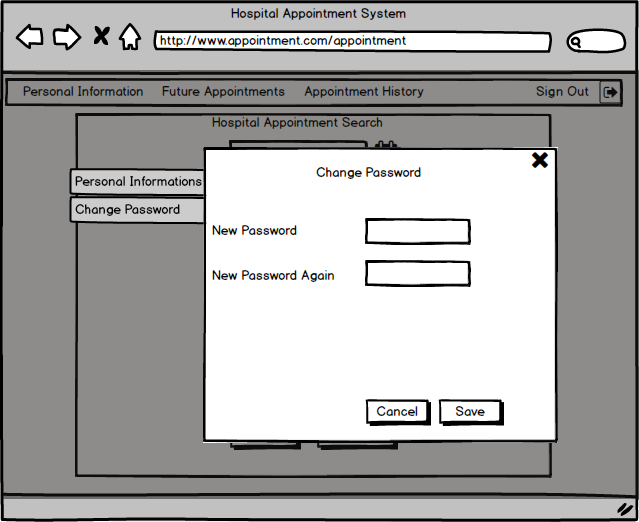
Make Appointment Mockup



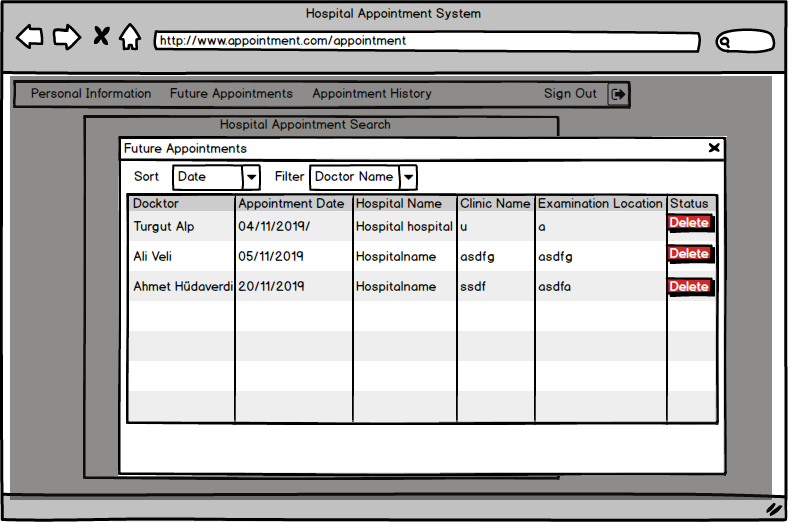


View Personal Informations Mockup

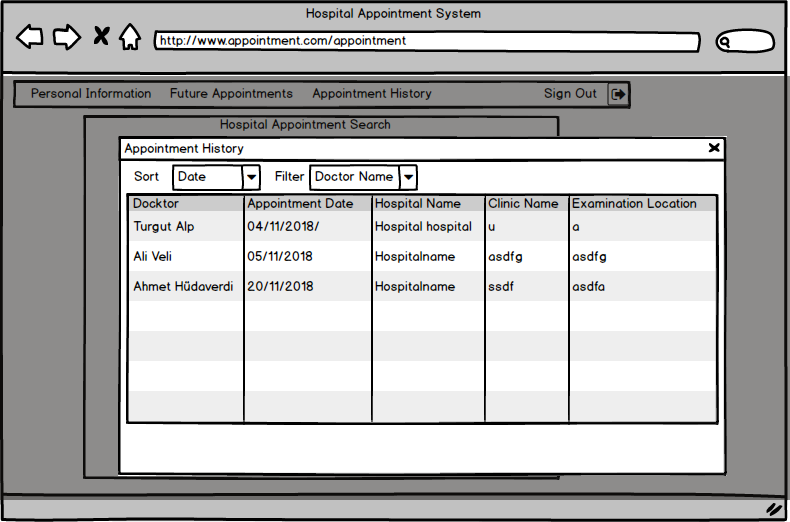
Change Password Mockup



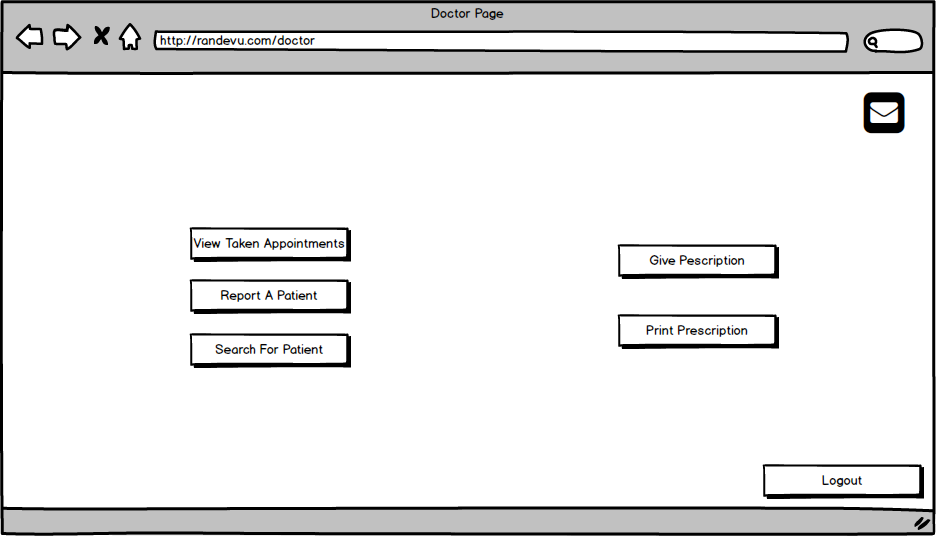
Future Appointments Mockup



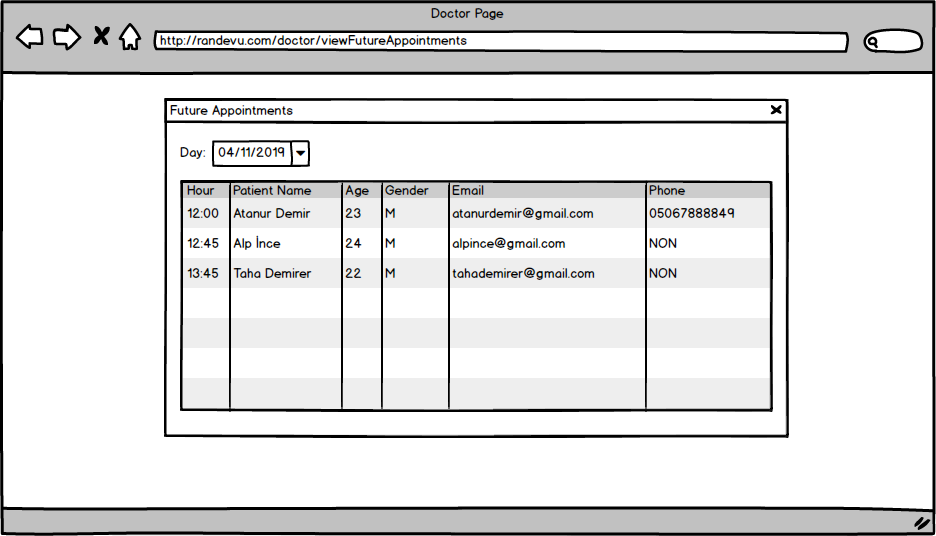
Appointment History Mockup



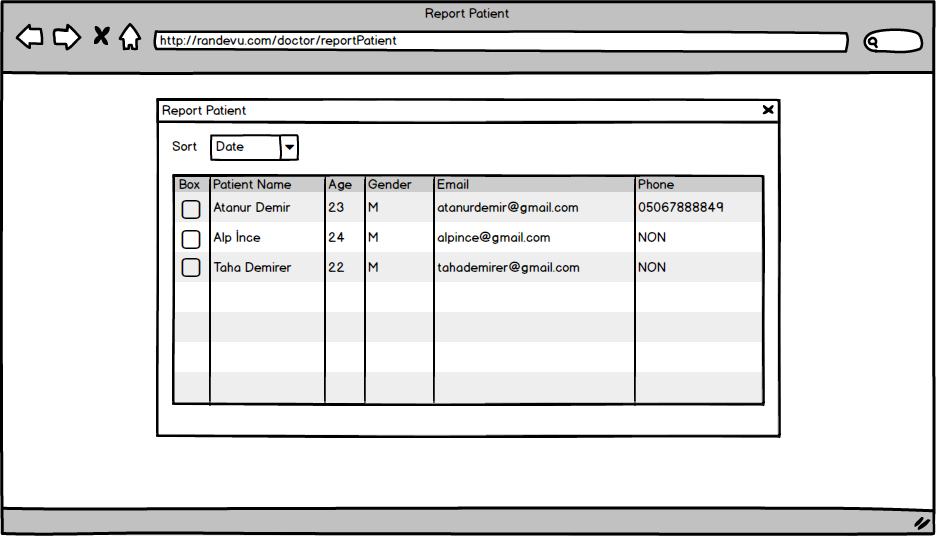
Doctor Page Mockup



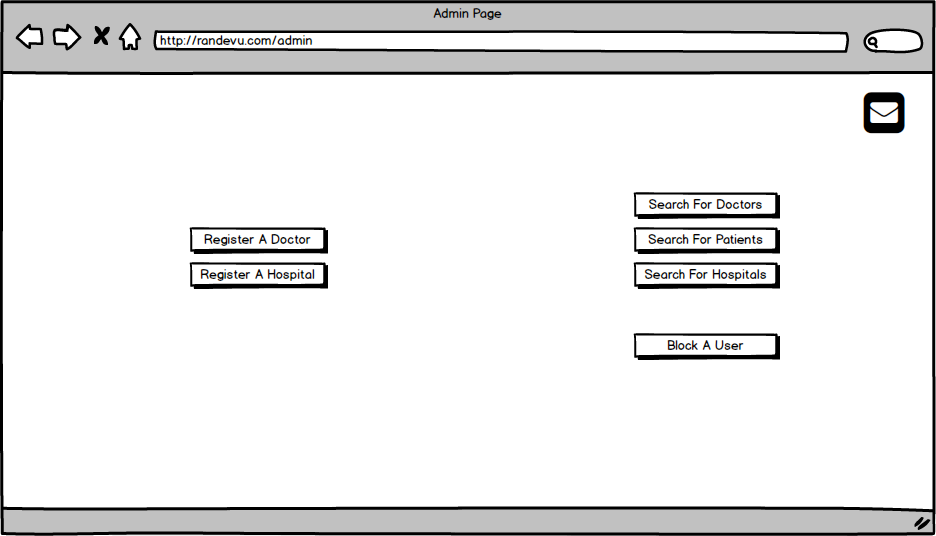
Future Appointments for Doctor Mockup



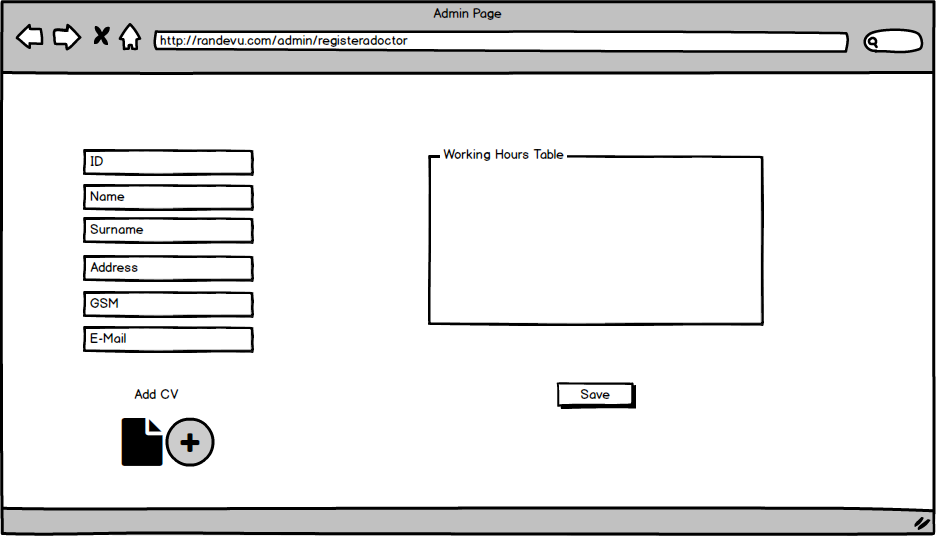
Report Patient Mockup



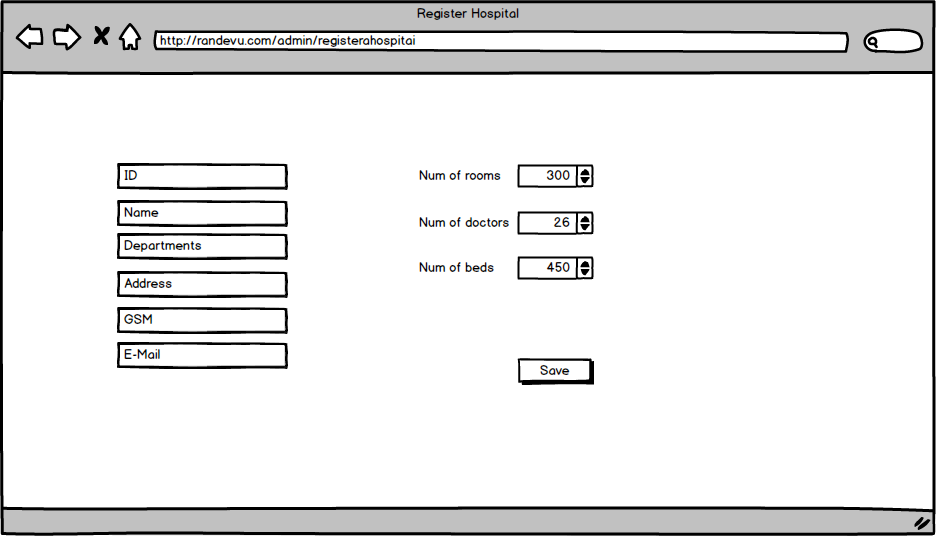
Admin Home Page Mockup



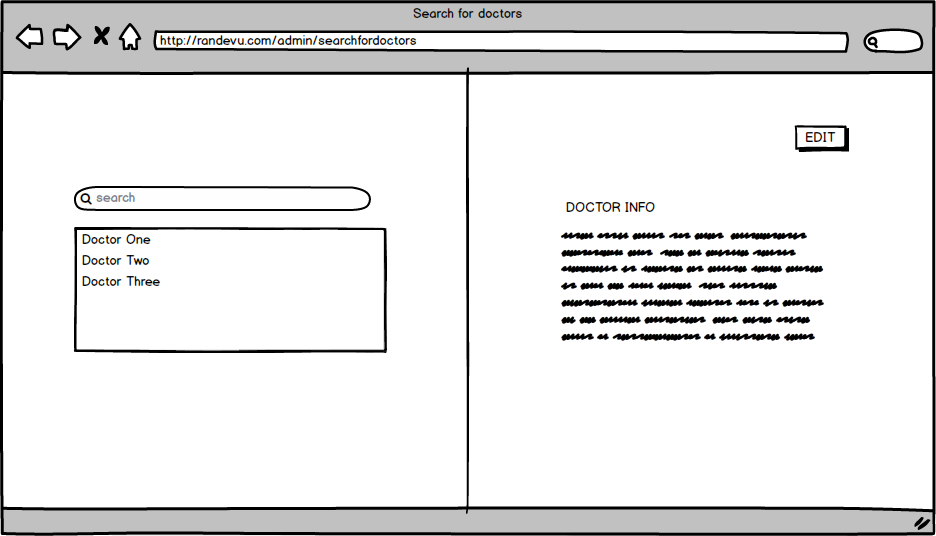
Register Docktor Mockup



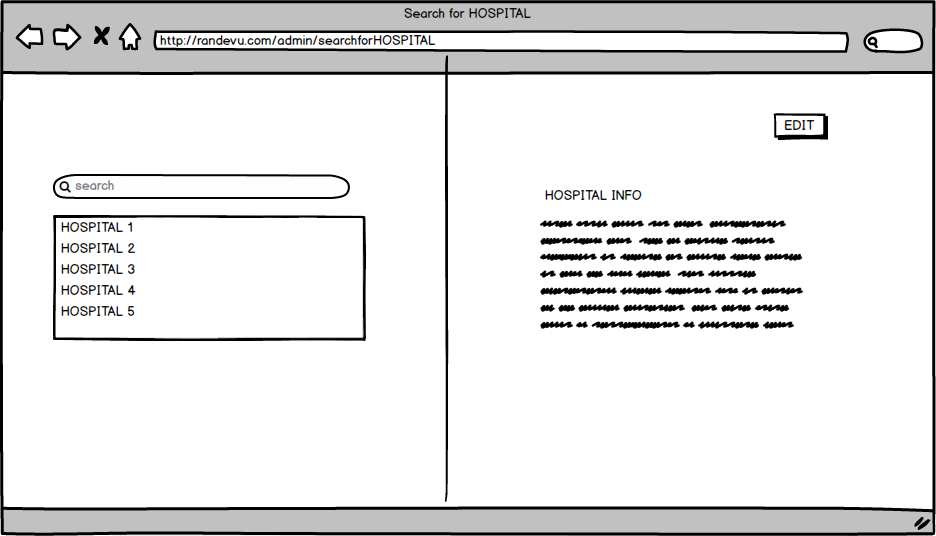
Register Hospital Mockup



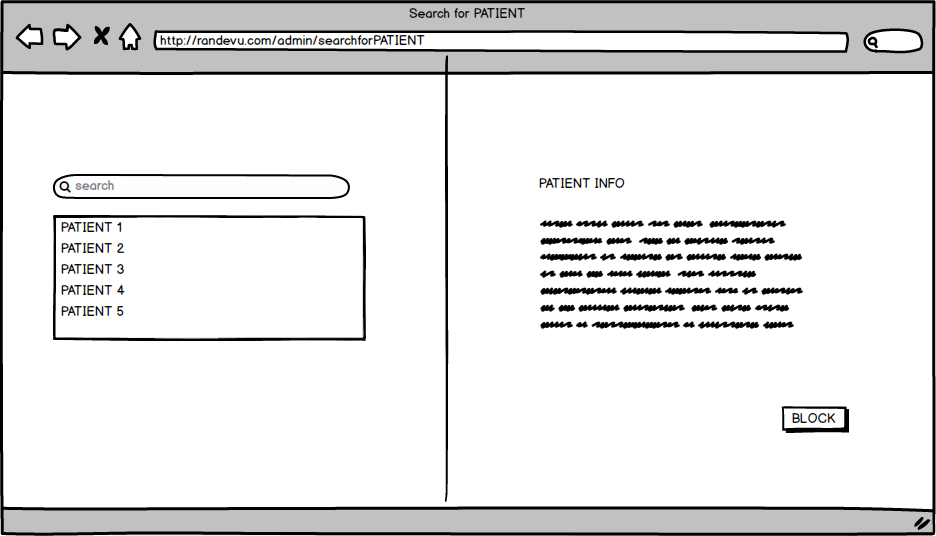
Search for Doctors Mockup



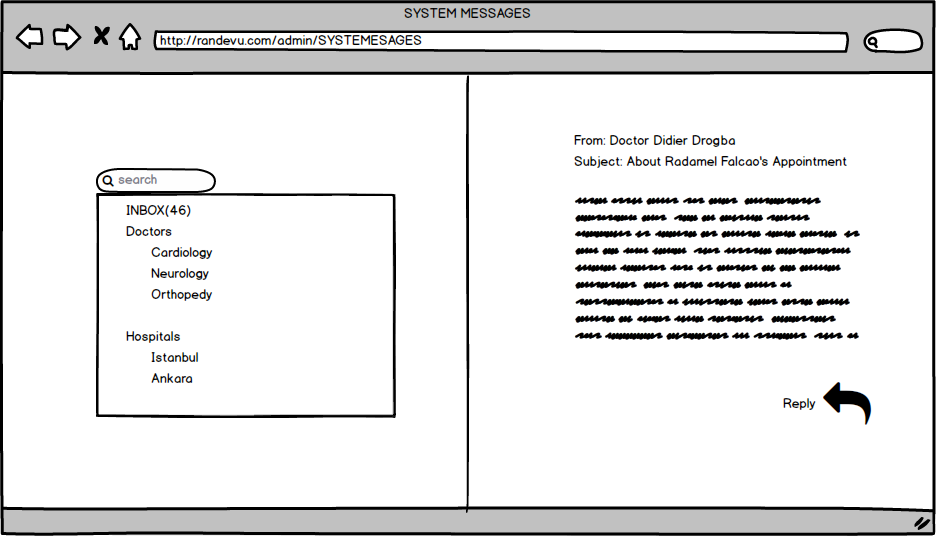
Search for Hospital Mockup

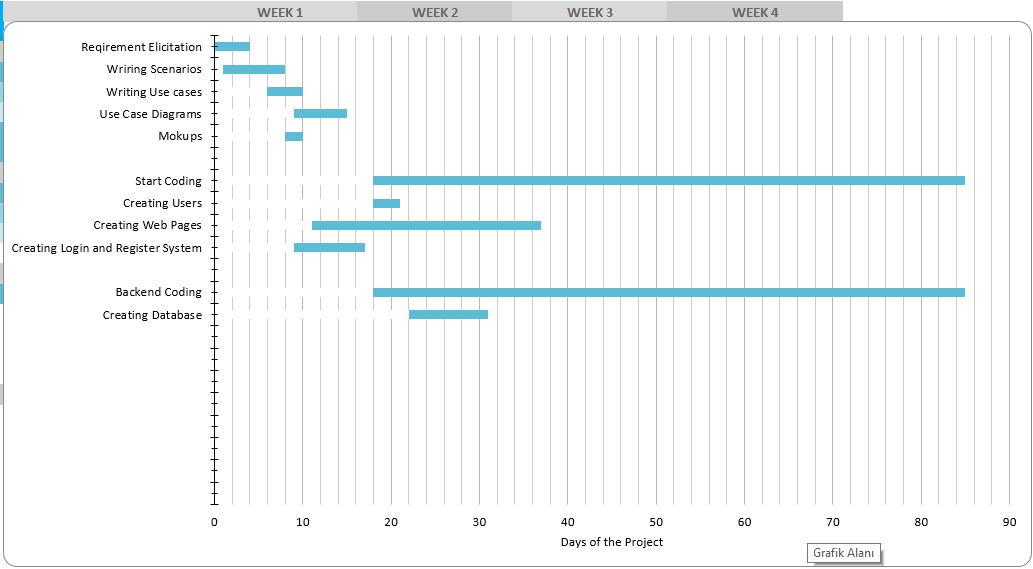


Search for Patients Mockup



System Messages Mockup



**3.5 Project Schedule**

**4. Glossary**

1. Administrator: the activities involved in managing or organizing a business or other organization
2. Doctor: a person with a medical degree whose job is to treat people who are ill or hurt
3. Patient: a person who is receiving medical care, or who is cared for by a particular doctor or dentist when necessary.

**5. References**