|  |  |  |  |
| --- | --- | --- | --- |
| *Prepared by* | | | |
| 1. | **Omnia Ahmad Fathy** | **Id: 20173248** |  |
| 2. | **Salma Essam** | **Id: 20172617** |  |
| 3. | **Nada Magdy** | **Id: 20172687** |  |

****

**Future University in Egypt**

**Faculty of Computers and Information Technology**

**Department of Computer Science**

CS 251 ( Software Engineering \_1 )

**Supervised by**

**Prof. Ramadan Moawad**

**AL. Amr Mansour**

**Project Name**

**Dentistry Clinic Software**

# Content page:

## Chapter one:

1. Problem definition…………………………………………….3
2. Scope…………………………………………………………………3
3. introduced solution……………………………………………3
4. Example scenarios……………………………………………..4
5. Requirements…………………………………………………….6
6. Target environment………………………………………..…7
7. Deliverables…………………………………………………….…7
8. Deadline………………………………………………………….…7
9. Acceptance criteria………………………………………….…7

## Chapter two:

1. Actors………………………………………………………………..8
2. Functional and Non-Functional Requirements….8
3. Use Case Diagram ……………………………………………10
4. Use case tables…………………………………………………11
5. Scenario tables ………………………………………………..16
6. Class Diagram………………………………………………….18
7. Sequence Diagram…………………………………………..19

# **Chapter one:**

# 1. Problem definition:

* 1. In clinics patient’s information is often kept in one file that include everything about the patient (name, email, phone number, address, medical history). Files information isn’t organized enough when searching for specific data.
  2. When a patient goes to another branch their file has to be transferred in some way from the original clinic to the other clinic and back which may cause loss of file.
  3. Classification of patients is troublesome using files in addition to the patient’s file might get lost.
  4. Calculations of procedures might get disorganized.

# Scope:

this software will be available for clinics with multiple branches or only one branch.

Can be used by medical clinics or dental clinics.

# Introduced Solution:

Normal user:

* + Every user can access their own file only without being able to change their medical history (for data consistency purposes), patient can change only their contact info or address.
  + Normal user would also be able to schedule an appointment with his doctor online with conformation of administrator.
  + Normal user will receive notification on their email if any change occurs to their appointment.

Administrator:

* can manage appointments.
* Can create profile for patient.
* View patients’ profile.
* update patient information.
* Send patient an email about their situation if needed.
* Manage prices of procedures.
* Confirm and cancel reservation.
* Add current medication
* Add service and print receipt.

# Example Scenarios:

* 1. Patient scenarios:
     1. Reservation:

Patient can reserve their appointment by logging in the system and clicking new appointment button then they can choose the day and time. then they can click save to submit or cancel to cancel.

* + 1. Change data:

The patient can login to change his number or address by clicking update data then click save or cancel.

* + 1. See file:

Patient can see his own page by clicking home or see file to see their data.

* + 1. Prices:

Patient can see the newest prices by clicking prices.

* 1. Administrator scenarios:
     1. Create patient medical history:

Nurse can create profile for patient medical history by choosing from the drop-down list his medical conditions and there is a text that she can fill if there is one of his medical

conditions that not in the drop-down list.

* + 1. View Patient history:

Doctor or administrator can see patient history by searching the patient name, then clicking on the view patient history button.

* + 1. Upgrade prices:

Doctors can upgrade prices by clicking upgrade prices button then save button when done or cancel.

* + 1. Create profile:

Nurse can create profile for patient by clicking create new profile then fill the name text, age text, ID text, address text and telephone number text.

* + 1. Create patient dental history:

Nurse can create profile for patient dental history by choosing from the drop-down list his dental conditions, then choose date and time. There is a text that she can fill if there is one of his dental

conditions that not in the drop-down list.

* + 1. Reserve appointment:

Nurse can reserve an appointment by clicking the button reserve appointment then choose date and time.

* + 1. Reservation:

When the nurse clicks this button, all reservations will show to her.

* + 1. Confirm reservation:

When the nurse chooses the button reservation all the reservation will appear, when she finds that the date and time is suitable, she will click the button confirm reservation.

* + 1. Add service:

When the nurse chooses patients service from the drop-down list, price will be calculated.

4.2.10. Print recite:

Recite will be printed when nurse clicks print recite button.

4.2.11. View profile:

Nurse can view patient’s profile.

4.2.12. Cancel reservation:

Nurse can cancel reservation by clicking cancel reservation.

4.2.13. Current medication:

Nurse can add current medication by clicking add current medication then fill doctor name text, medication name text, date text, note text.

# Requirements:

* 1. Contain two different users
  2. Support backup
  3. Search by file name, file date
  4. Send and receive mails from users
  5. Make prices fixed
  6. Data abstraction
  7. Secure software

# Target environment:

* Since DMS will be J2EE application so.
* Work on different OS e.g. windows and Linux.
* JVM.
* Apache Tomcat Server.
* MYSQL as DBMS.

# Deliverables:

* Proposal
* Requirement Analysis Document (RAD)
* Software Design Document (SDD)
* Implementation
* Test document

# Deadlines:

All deliverables deliver on 6-7-2020 at 12:00 PM

# Acceptance Criteria:

**The project DMS is accepted if at least the following success criteria are satisfied:**

* The DMS executable is available on (Date) and satisfies the functional requirements specified in previous sections.

# **Chapter two:**

## Actors:

* 1. Patient
  2. Medical staff
  3. Receptionist

## Functional and nonfunctional requirements:

2.1. Functional Requirement: -

2.1.1. Patient: -

•View file: the user can shell view their information on clinic system without being able to change their medical history.

•Schedule appointment: the user can shell see the appointment of doctor’s clinic.

•View prices: the user can shell check the prices of clinic online.

•Booking: the user can book an appointment at doctor’s clinic online.

•Checking: user (patient) can see their medical history and check medicine.

2.1.2. Receptionist:

• Create profile: the administrators can create a profile for the patient with: name, id, address, telephone number, medical and dental history.

• Edit appointment: the administrators can cancel, edit, change an appointment.

* Confirmation: administrators can confirm the appointment of the user by email.
* Update patient information: administrators can update, add, remove patient information.
* Edit prices: administrators can add, edit, update prices.

2.1.3. Medical staff:

•Add medical history: administrator can add a medical condition.

•Add dental history: administrator can add dental history of patient.

•Add service: doctor adds service from a drop-down list for a procedure.

•Calculate price of procedure: price of procedure is calculated automatically when administrator add service.

2.2. Non-functional requirement: -

2.2.1Patient: -

• patient can’t change their information.

• patient can’t change the clinic prices.

• Patient an only change their info &address.

• Patient should wait for a confirmation message from the administrators.

• Patient is not allowed to change anything in their medical history.

2.2.2. Receptionist and medical staff (adminstraitors): -

• can’t add two appointments at same time.

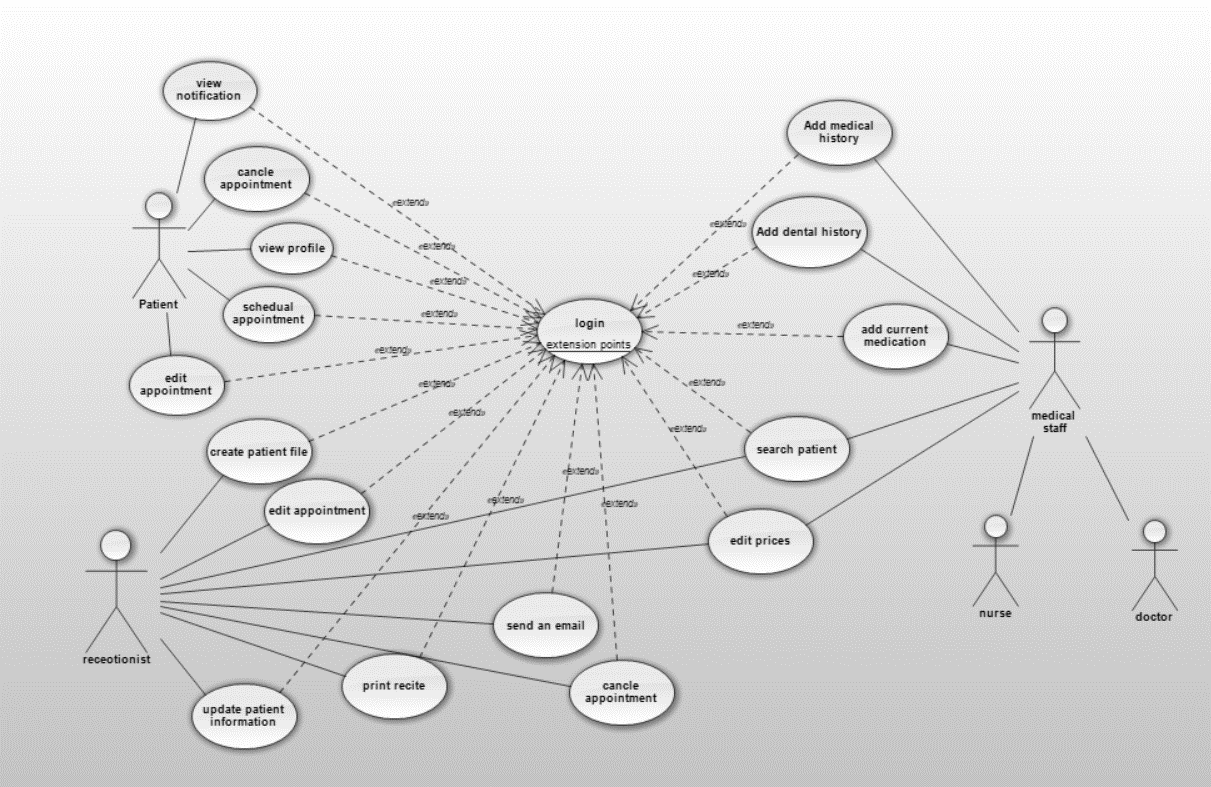
• administrators only can add, edit, update the patient information.

• administrators should send confirmations email to approve the patient appointment.

• administrators the only one control the clinic prices.

• Administrators the only one can control clinic appointment.

## Use case diagrame:



## Use case tables:

4.1. use case reservation:

|  |  |  |
| --- | --- | --- |
| Use case id | 1 | |
| Use case name | Reservation | |
| Actors | Patient | system |
| Flow of events | Patient will click new appointment button |  |
|  | System will open the appointment page. |
| User will choose date and time |  |
|  | System highlight date and time |
| User will click save button |  |
|  | System will send date and time to administrator. |
| Pre conditions | Patient must login | |
| Post conditions | Appointment sent to admin | |
| Nonfunctional requirements | Patient should wait for a confirmation message from admin. | |
| Input | None. | |
| Output | None. | |
| Exceptions | User chose an appointment already chosen. | |

|  |  |  |
| --- | --- | --- |
| Use case id | 2 | |
| Use case name | View file | |
| Actors | Administrator (nurse, doctor) | system |
| Flow of events | User press click to see file |  |
|  | System will show patient file |
| Pre conditions | Admin must be logged in system | |
| Post conditions | Data of file will show to admin | |
| Nonfunctional requirements | none | |
| Input | None. | |
| Output | None. | |
| Exceptions | None. | |

4.2. use case view file:

4.3. use case show price:

|  |  |  |
| --- | --- | --- |
| Use case id | 3 | |
| Use case name | Show price | |
| Actors | user | system |
| Flow of events | Patient will choose prices button and click it |  |
|  | System will show prices page |
| Pre conditions | Patient must login | |
| Post conditions | Prices page appear | |
| Nonfunctional requirements | Patient can’t change anything in this page | |
| Input | None. | |
| Output | None. | |
| Exceptions | There’re no exceptions in this case | |

4.4. use case change data:

|  |  |  |
| --- | --- | --- |
| Use case id | 4 | |
| Use case name | Change data | |
| Actors | Patient | system |
| Flow of events | The user will login and can change their own data (address, number) |  |
|  | System will show them options to save or cancel the updated data. |
| The user clicks save if they want to update their data. |  |
|  | System will save option |
| Pre conditions | Patient must login | |
| Post conditions | The new data will be updated | |
| Nonfunctional requirements | Patient can’t change anything in this page | |
| Input | None. | |
| Output | None. | |
| Exceptions | The system will pass exception if user don’t write their username or password is not correct | |

|  |  |  |
| --- | --- | --- |
| Use case id | 5 | |
| Use case name | View patient history | |
| Actors | Administrator | system |
| Flow of events | Search for patient by name by typing their name is search bar and clicking search. |  |
|  | Provide a list with patients with searched name. |
| Click on required patient to view history. |  |
|  | Provide patient’s profile page. |
| View patient history |  |
| Pre conditions | Administrator has to be logged in | |
| Post conditions | None. | |
| Nonfunctional requirements | Administrator must be logged in as an administrator to have authority to look at patient’s profiles | |
| Input | Patient name | |
| Output | Patient profile | |
| Exceptions | -patient not found  -patient has no history | |

4.5. use case view patient history:

|  |  |  |
| --- | --- | --- |
| Use case id | 6 | |
| Use case name | Update patient history | |
| Actors | Administrator | system |
| Flow of events | Search for patient by name by typing their name is search bar and clicking search. |  |
|  | Provide a list with patients with searched name. |
| Click on required patient to view history. |  |
|  | Provide patient’s profile page. |
| Click update medical or dental history |  |
|  | Show drop down list and a text field for text. |
| Choose patient conditions from drop down list and/or write condition in text field of unlisted. |  |
| Click save. |  |
|  | Drop down list and text field disappear and changes are saved. |
| Pre conditions | Administrator must be logged in as an administrator to have authority to look at patient’s profiles | |
| Post conditions | Patient history has been updated. | |
| Nonfunctional requirements | Administrator must be logged in as an administrator to have authority to look at patient’s profiles | |
| Input | Patient name, history changes. | |
| Output | Updated patient history. | |
| Exceptions | -patient not found. | |

4.6. use case update patient history

### Scenario tables:

##### View patient history:

|  |  |
| --- | --- |
| Scenario id | 1.1 |
| Scenario name | Patient history viewed |
| Scenario description | Doctor Ahmad wants to view the medical and dental history of a new patient named Hamed so doctor Ahmad searches the patient name in the data base and finds the Hamed’s file and views his patient history. |

##### Add medical condition:

|  |  |
| --- | --- |
| Scenario id | 1.2 |
| Scenario name | Add medical condition |
| Scenario description | Doctor Ahmad wants to add diabetes to patient Hamed’s file so doctor searches for Hamed’s file in the database then opens Hamed’s file and licks on update medical history. A drop-down list appears with blood pressure, diabetes,... etc. doctor choses diabetes and clicks save. |

##### Show price:

|  |  |
| --- | --- |
| Scenario id | 2.1 |
| Scenario name | Show price |
| Scenario description | When Ahmed wants to check the prices he will choose show price button then system will show him the price page. |

##### Reservation successfully:

|  |  |
| --- | --- |
| Scenario id | 2.2 |
| Scenario name | Reservation successfully |
| Scenario description | Ahmed wants to reserve an appointment so he will click the new appointment button so system will open the appointment page after that Ahmed will choose the time and the date he wants to reserve in .System will sent the date and time to the admin .if Ahmed can come in this time admin will send an email. Ahmed will reserve a notification tilling him that he can come. |

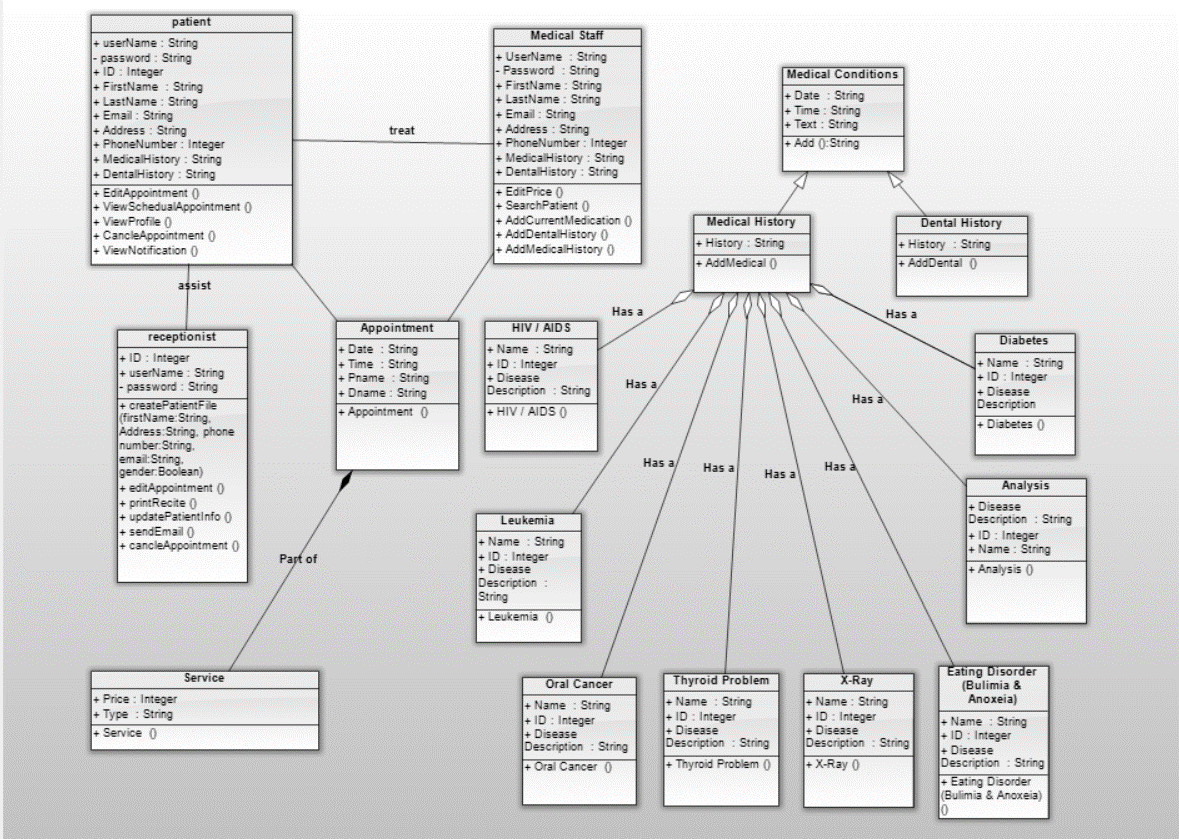
##### Change data:

|  |  |
| --- | --- |
| Scenario id | 3.1 |
| Scenario name | Change data |
| Scenario description | Hamed wants to change some information in his profile as number or address, so he logs in to system then starts changing his data and have two choices save or cancel |

##### See file:

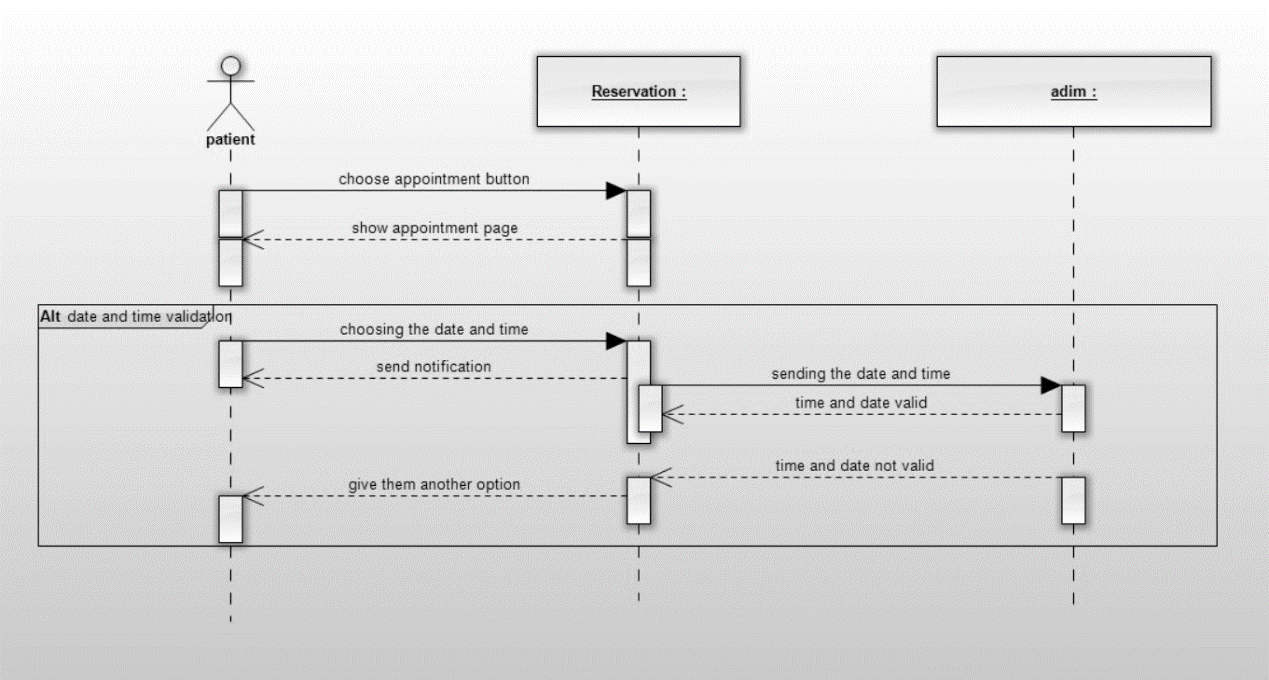
|  |  |
| --- | --- |
| Scenario id | 3.2 |
| Scenario name | See file |
| Scenario description | Hamed wants to see his profile, so he logs n to system then click home and can see his profile |

### Class diagram:

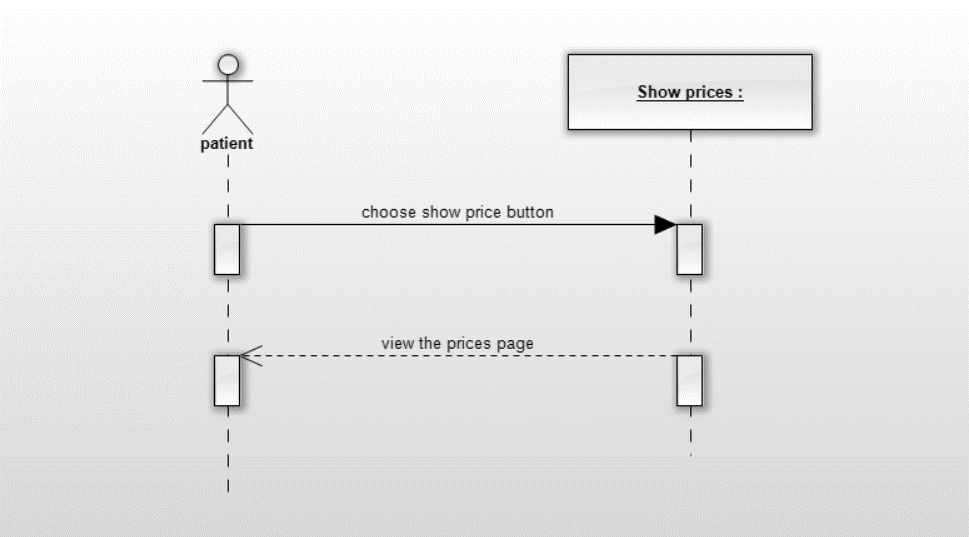


### Sequence diagram:

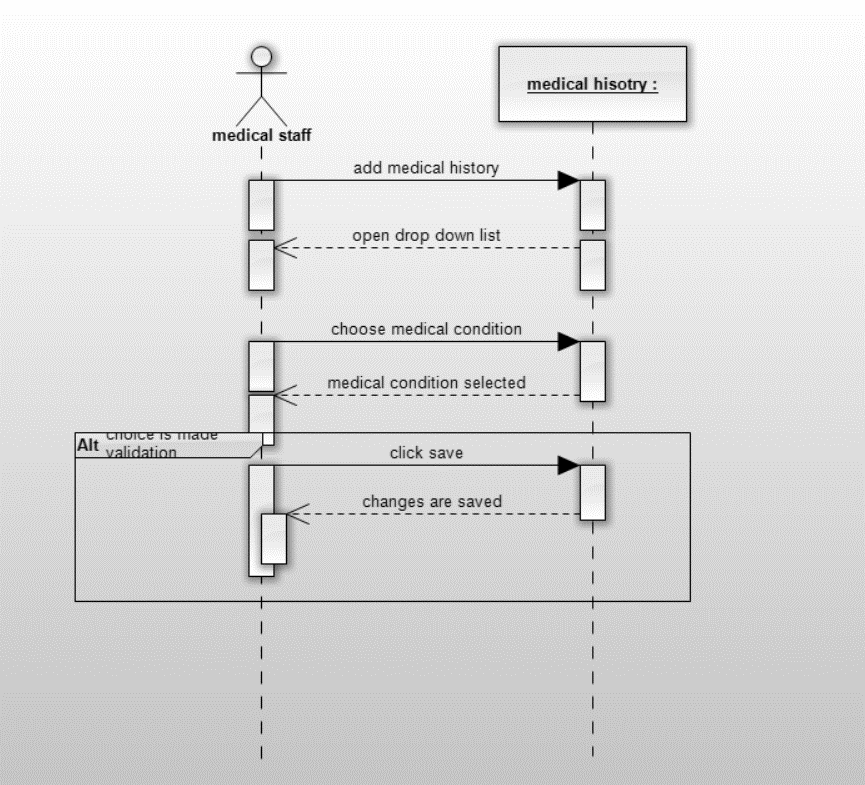
* 1. Sequence diagram reservation:



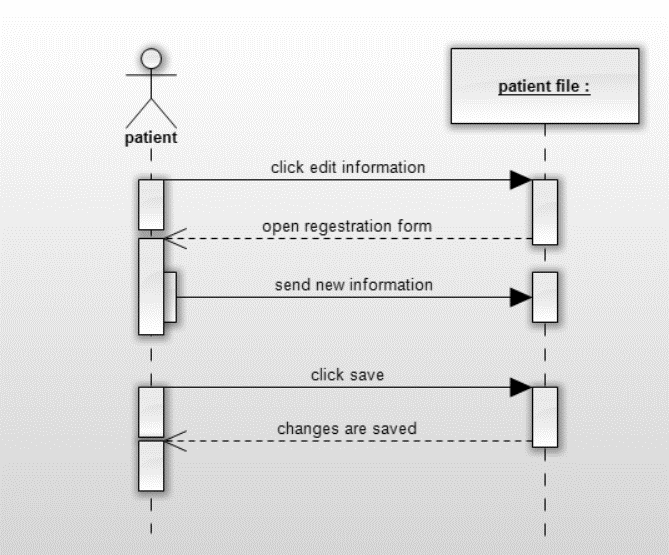
* 1. Sequence diagram show prices:



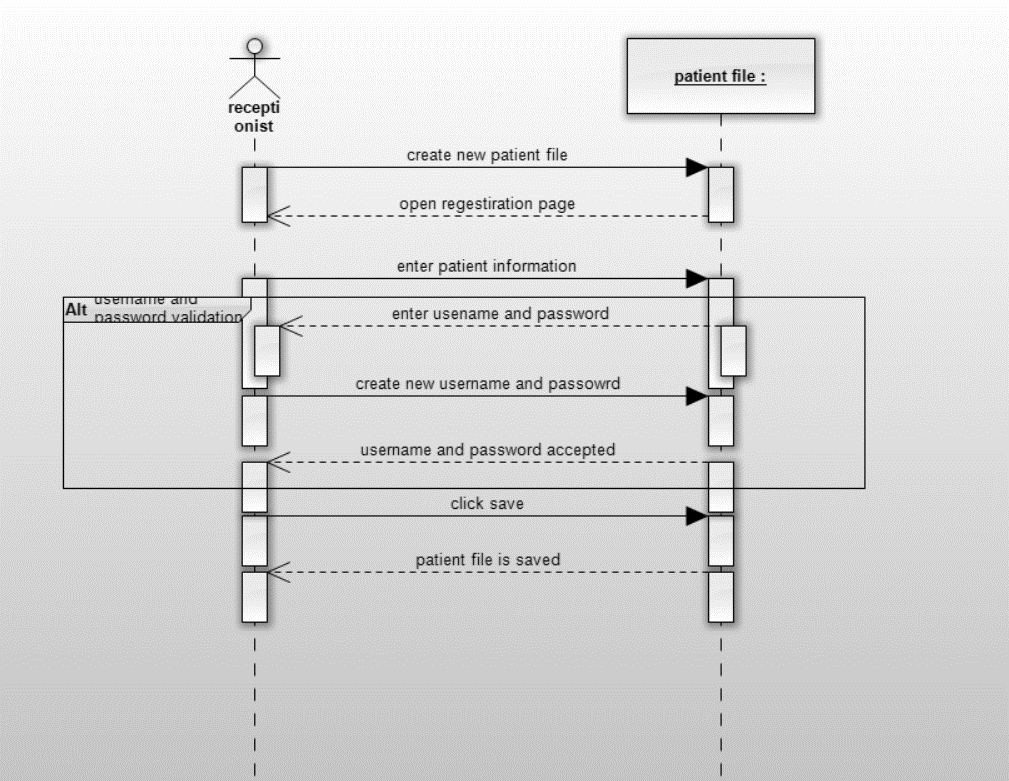
* 1. Sequence diagram add medical history:



* 1. Sequence diagram edit patient file:



* 1. Sequence diagram registration:



* 1. Sequence diagram view patient information:

