Software Requirements Specification

For

**Electronic Health Record**

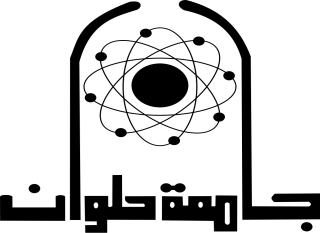
“EHR”

*Version 1.0 approved*

*Prepared by FCIH-Group 36*

*Helwan University*

*18/10/2018*



Faculty of computers system &

Information Systems Department

Supervised by:

DR. Manal Abdel-Kader

Team Members

Abdullah Mohamed Omar Gamal Abdalfattah Mohamed

Mohamed Ashraf Noran Mahmoud

Table of Contents

Table of Contents ii

Revision History ii

Chapter 1 : Introduction

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 2

Chapter 2 : project "Planning & Analysis "

2. 1 Overall Description 3

2.1.1 Product Perspective……………………………………………………………………………………………………………….….3

2.1.2 Product Functions 3

2.1.3 User Classes and Characteristics 4

2.1.4 Operating Environment 4

2.1.5 Design and Implementation Constraints 5

2.1.6 User Documentation 5

2.1.7 Assumptions and Dependencies 5

2.2 External Interface Requirements 6

2.2.1 User Interfaces 6

2.2.2 Communications Interfaces 12

2.3 Analysis System Requierment 13

2.3.1 User Requierments 13

2.3.1 .1 Mondatory Requierments 13

2.3.1.2 DesirableRequierments 13

2.3.1.3 Optional Requierments 13

2.3.4 System function Requierments 13

2.3.5 System non function Requierments 19

2.3.6 Use-case senario 29

Chapter 3 : project "Design "

3.1 Use-case Diagram 37

3.2 Activity Diagram 41

3.3 Sequence Diagram 51

3.4 Design of Database 61

3.5 Class Diagram 4

Chapter 4 : project "implemention "

4.1 Hardware Architecture 62

4.1 software Architecture 63

**Chapter 5: project "Testing "**

5.1 Unit Testing 62

5.2 Integrated Testing 62

5.3 Additional Testing 62

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## **Purpose**

The main purpose of our EHR is to get rid of all paper work that is being done in hospitals and laboratory by using an automated system that make it easier for the hospitals and patient to keep track of their Patient health history which help doctors to identify illness quick by seeing patient history and it will get rid of all paper work is done by a doctor and it will make it easier for laboratory to make the result into the patient account and add it to his history which will life easier for the patient where he will not be worried to lose the lab result and his radiation result.

## **Document Conventions**

This document will use IEEE format for clarity, acronyms and technical jargon, deemed uncommon by the author, will be annotated and included in the glossary. Standard IEEE template is the template used to organize the appearance of the document and its flow.

The document is prepared using Microsoft word 2010.The format for heading is as followed:

Major headings are in **bold18pt font**, concurrent headings in **bold 14pt font** but the fixed font size will be 12pt font with 1.5 line spacing. All pages expect the cover page are numbered, the numbers appear on the higher right hand corner of the page. Every image and data table are numbered and referred to the in main text. Sections are in the format of x .y, where x. y are real, positive integers.

## **Intended Audience and Reading Suggestions**

This document is intended for Developers, marketing staff, testers, Graduation project doctor and developers who might like to continue this project.

Recommendation to start reading from:

1. Developers: overall description, system features then nonfunctional requirements.

2. Marketing stuff: overall description then external interface requirements.

3. Testers: everything.

4. Graduation project doctors: introduction, overall description, system features then nonfunctional requirements.

5. Developers who might like to continue this project: everything.

## **product scope**

The electronic health record system is an online website allows patient to have a medical profile that contains his prescription, medicines, and lab analyses. Patient has an ID carries his information; doctor and pharmacy, lab doctors can see his medical history. The doctor can add data to the patient history. The data goes to pharmacy and lab at the time of upload it. Patient can go immediately to get his medicine without waiting in a queue and lab is the same. System notify patient who has a doctor appointment. Patient can review doctor, pharmacy, lab and hospital for their services; he can also see our partners from hospitals, pharmacies and labs.

## **References**

Big Data: (Hadoop- The Definitive Guide, 4th Edition Book)

System Language: Python 3

Framework: Django (python high-level framework)

EHR Information Resource: https://www.healthit.gov

Github: <https://github.com/abdullah12388/EHR-project>

# Overall Description

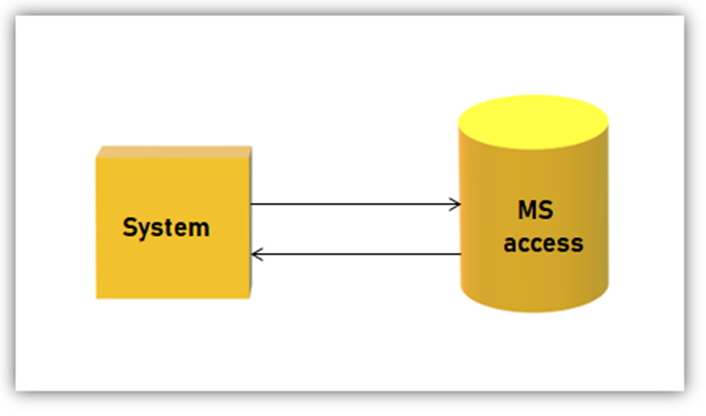
## **Product Perspective**

Our system will be web-based, self-contained system and independent software product.

It manages activities of the hospital, clinic, laboratory and radiation center as patient info.

It’ll not require any browser but it’ll need an external interface include keyboard, mouse, enabling navigation across screens.

Several of users will use our system such as patient, doctor, HR, Admin, pharmacy and laboratory but, at the same time there’re can’t do anything on our system without permission to access.



## **Product Functions**

One of the major functions is to store the patient data by a doctor or a staff member and also to have doctor appointment time in visible to the user and all his illness history which help in analysis to find if there is a new disease is spreading also the patient , doctor and staff can create account on the system the pharmacist can view only the medicine of the patient the doctor have written and the medicine for him and this and also their a statistics about the medicines and the lap result are add to the patient history.

## **User Classes and Characteristics**

One of the main users of system is:

The **doctor** the doctor can: View patient history, add to the patient history, Add the medicine name by using a voice recognition or by writing the name.

The **pharmacist** can: view only the patient medicine that he is taking and can view statistics about the medicine.

The **lap doctor** can add the patient lap result to his account and view previous lap results and view statistics about lap result of all patients.

The **Patient** can view his account and update his personal data and have a smart card that he uses to give to a doctor or any of the staff to access the specific data they need.

## **Operating Environment**

* System requirement (Hardware) :
* CPU: at least Dual core 3.6 MHZ.
* Processor: Pentium IV 2 GHZ and all above.
* Space: minimum of 500 MB.
* Monitor: VGA color monitor.
* Ram: minimum of 1 GB.
* System requirement (Software) :
* Web service: Django (framework) and spark.
* Operating system: Windows XP and all above.
* Browser: Mozilla fire fox, Microsoft internet explorer or Google chrome.
* Other service: python.

## **Design and Implementation Constraints**

* Hardware limitation:
* Camera Scan QR.
* Timing must be fast.
* Memory requirements are large.
* Software Limitation:
* Installation (python/framework)
* Hadoop-spark
* Security

## **User Documentation**

These things will be delivered while working on the project:

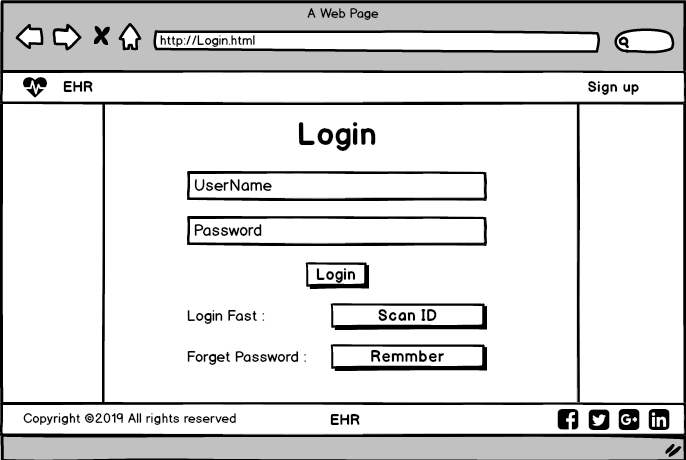
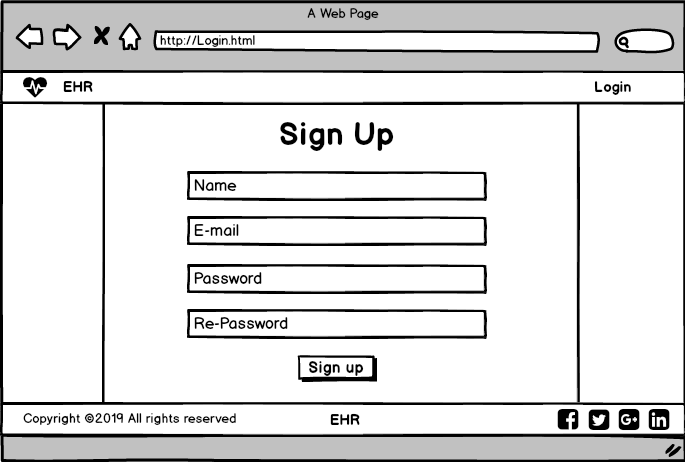
1. We will deliver an easy documentation that the user can back to it when he get stuck or something like that
2. We will make a simple video that shoes how to use our website that anyone can benefit from it
3. We will deliver a video that can be (.mp4) and web page inside the website

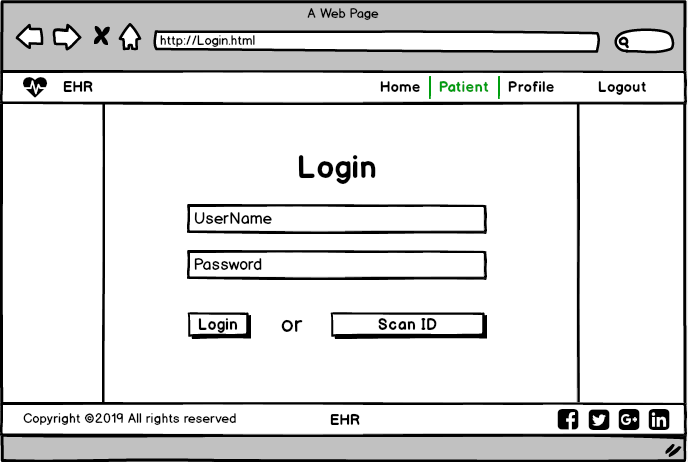
## **Assumptions and Dependencies**

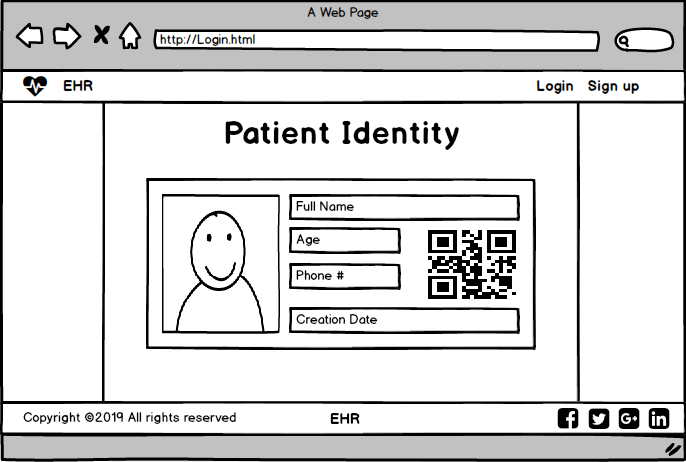
We assume that we will be the first one to build this project and make it in public. We will face some issues which will be: we do not have any data set to work on, so we built it from scratch. We assume that we will face a host problem because we want a secret one that can provide a good space and high security at the same time. We will not use any third party tools except ###tools for using a Big Data we will create a new data set to work on to do our transactions.

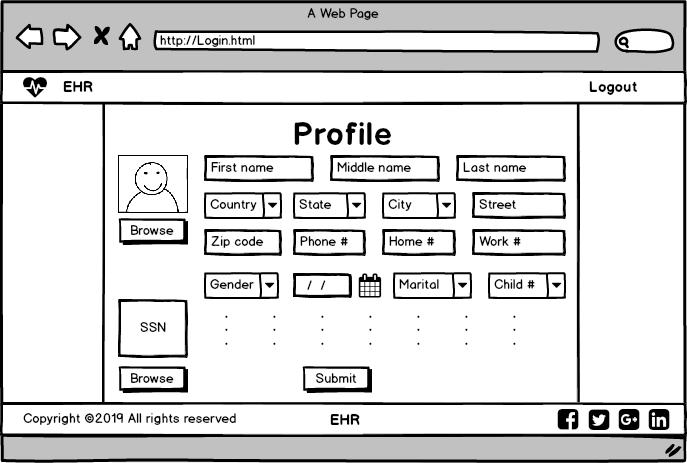
# External Interface Requirements

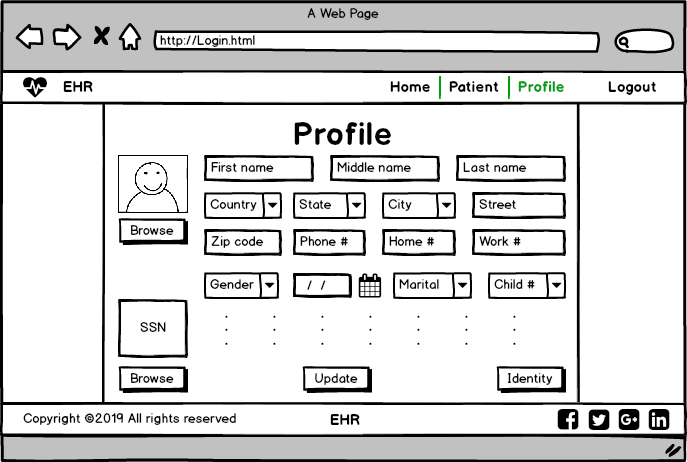
## **User Interfaces**

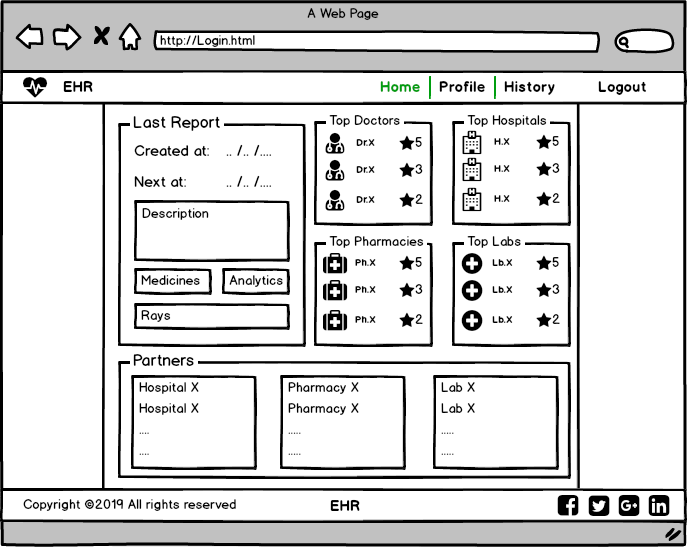
 

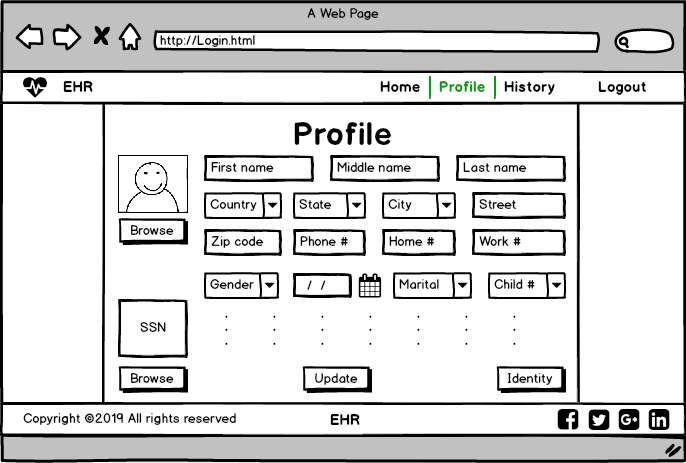


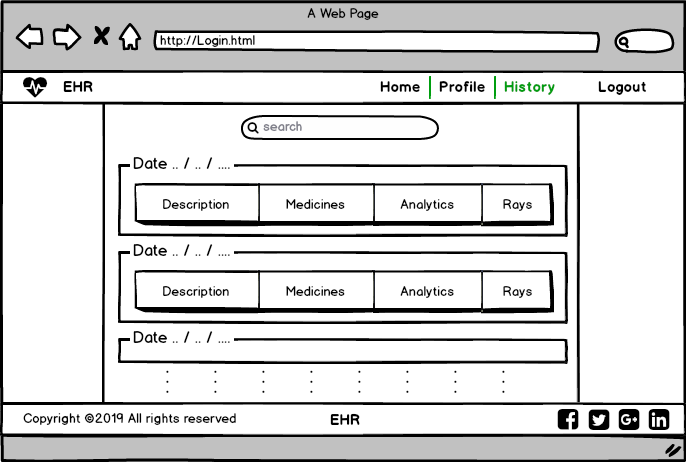


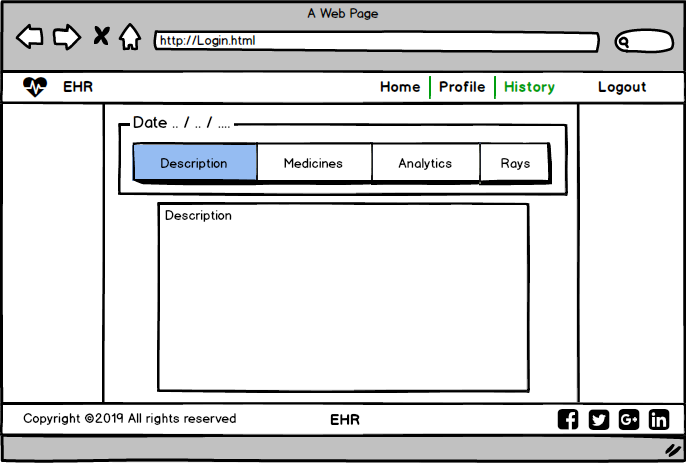


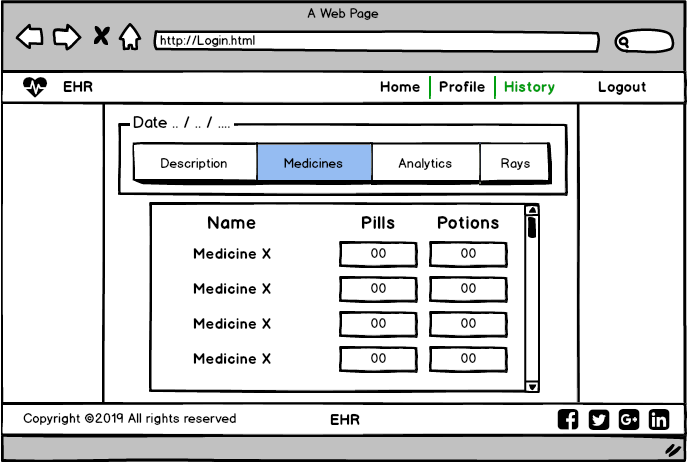


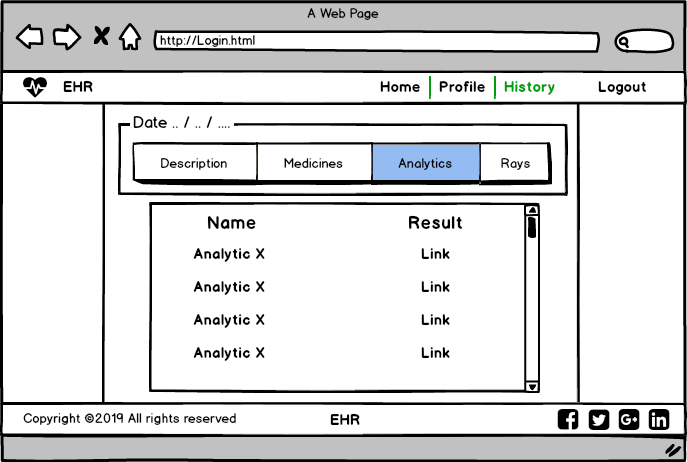


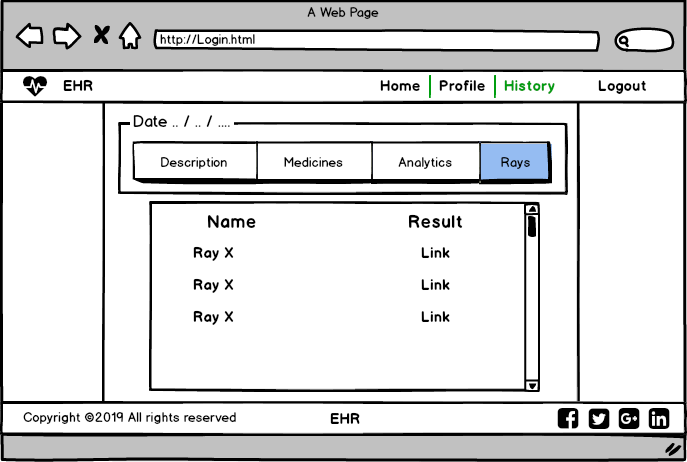


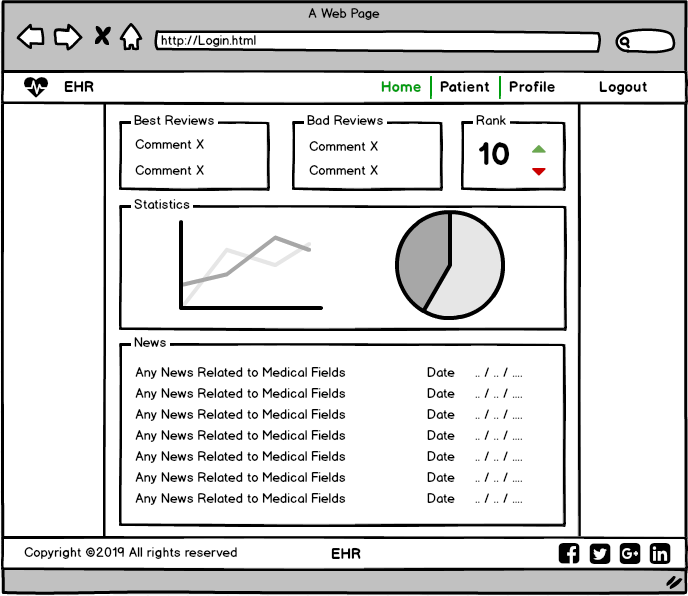




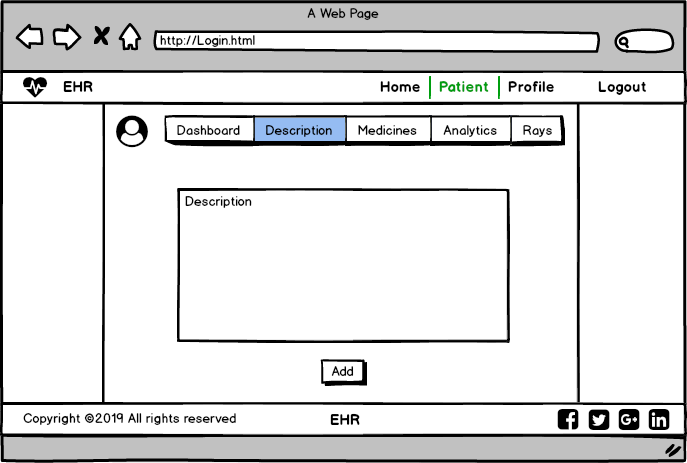
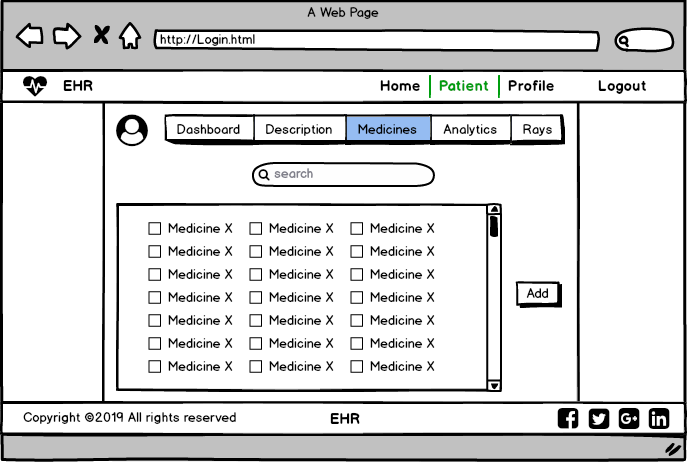


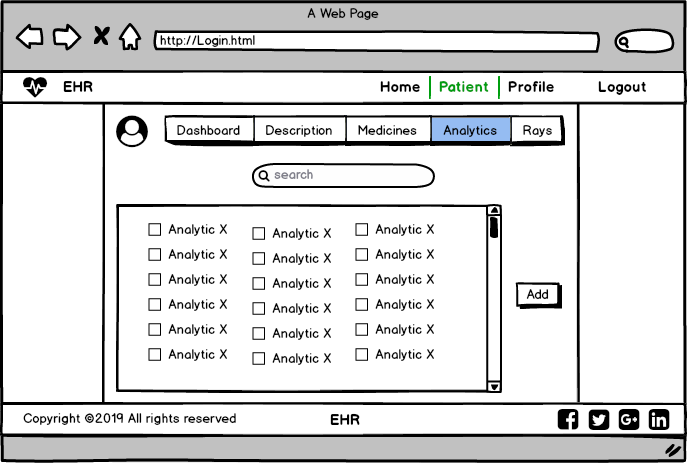


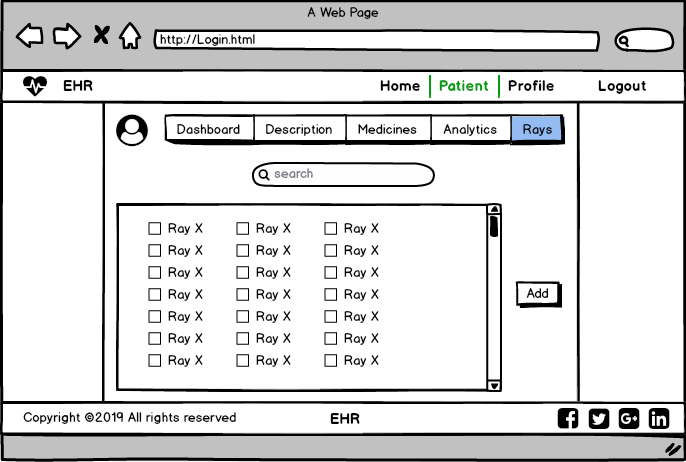










## **Communications Interfaces**

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Analysis System Requierment

## **user requirements:-**

## **Mandatory requirements:**

1- The user searches for any app or site to transfer all paper working in technical work at any place easily only by few clicks without need to broker so we will cover this point in medical field in our site.

2-The user not only needs that but also needs to notify by all doctor appointment and show more about good doctors , so we will cover all this categories in our system.

## **Desirable requirements:**

1- The user needs to use GPS to make the process of finding clinics and hospitals easily so, we will provide this feature in our app.

2- The user uses the app with him at any place and any time so our app should be available in any place and any time.

## **Optional requirement:**

1. The users prefer a social media part to communicate with each other so, we will provide this feature in our app.

## **function requierments**

|  |  |
| --- | --- |
| **Code** | F1 |
| **Name** | Log in |
| **Role** | Admin – client – HR – Lab – Pharmacy – Doctor |
| **Type** | Functional Requirement |
| **Description** | Give the actors a permission to do there functions and access the system features |
| **Pre-Condition** | Has a computer or mobile and internet connection |
| **Post-Condition** | Open log in page |
| **Input** | Enter log in required information |
| **Output** | Accept/reject log in action |

|  |  |
| --- | --- |
| **Code** | F2 |
| **Name** | Register |
| **Role** | Admin – client – HR |
| **Type** | Functional Requirement |
| **Description** | Take the required information from the actors, check if exists or not and make him an new account with some steps to log in our system successfully |
| **Pre-Condition** | Not registered before, has a computer or mobile and internet connection |
| **Post-Condition** | Open register page |
| **Input** | Enter register required information |
| **Output** | Accept/reject register action |

|  |  |
| --- | --- |
| **Code** | F3 |
| **Name** | Show data |
| **Role** | Lab – Pharmacy |
| **Type** | Functional Requirement |
| **Description** | General function allows actors to see kind of data related to his access for example lab can see rays and analytical data, pharmacy can see medicine data |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with the account |
| **Input** | Action from show data button |
| **Output** | Open data page |

|  |  |
| --- | --- |
| **Code** | F4 |
| **Name** | Show Reviews |
| **Role** | Client |
| **Type** | Functional Requirement |
| **Description** | It's a function allows the client to see the hospital, doctor, lab and pharmacy reviews and ranking |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with the account |
| **Input** | Press show reviews button |
| **Output** | Open reviews page |

|  |  |
| --- | --- |
| **Code** | F5 |
| **Name** | Show History |
| **Role** | Client – Doctor |
| **Type** | Functional Requirement |
| **Description** | This function allows actors to see the whole medical history from doctor description to the medicine and rays…etc |
| **Pre-Condition** | User must has an account |
| **Post-Condition** | Log in with his account |
| **Input** | Action from the history button |
| **Output** | Show the page of all data |

|  |  |
| --- | --- |
| **Code** | F6 |
| **Name** | Edit Personal Data |
| **Role** | Client – HR |
| **Type** | Functional Requirement |
| **Description** | This function allows the actors to edit their personal data like name and address and so on |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with his account |
| **Input** | New data |
| **Output** | Notification with successful editing and open profile |

|  |  |
| --- | --- |
| **Code** | F7 |
| **Name** | Get Doctor Appointment Notification |
| **Role** | Client |
| **Type** | Functional Requirement |
| **Description** | This function send a notification message to the patient remember him with the doctor appointment |
| **Pre-Condition** | Have an account, has a meeting with doctor |
| **Post-Condition** | Log in with his account |
| **Input** | Automatic input from doctor schedule |
| **Output** | Notification message with doctor appointment |

|  |  |
| --- | --- |
| **Code** | F8 |
| **Name** | Show Our Partners |
| **Role** | Client |
| **Type** | Functional Requirement |
| **Description** | This function allows the actor to know our partners like hospitals, pharmacies and labs to go to the nearest one |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with his account |
| **Input** | Press show our partners button |
| **Output** | Open partners page |

|  |  |
| --- | --- |
| **Code** | F9 |
| **Name** | Add Clinic |
| **Role** | Admin |
| **Type** | Functional Requirement |
| **Description** | The admin uses this function to add a clinic as an organization to our system and make the doctor responsible for it |
| **Pre-Condition** | Must has an account with admin permission |
| **Post-Condition** | Log in with his account |
| **Input** | Clinic and doctor data |
| **Output** | Clinic info and doctor profile |

|  |  |
| --- | --- |
| **Code** | F10 |
| **Name** | Add Hospital |
| **Role** | Admin |
| **Type** | Functional Requirement |
| **Description** | The admin uses this function to add a hospital as an organization to our system and make the HR responsible for it |
| **Pre-Condition** | Must has an account with admin permission |
| **Post-Condition** | Log in with his account |
| **Input** | Hospital data |
| **Output** | Hospital info added successfully |

|  |  |
| --- | --- |
| **Code** | F11 |
| **Name** | Remove Clinic |
| **Role** | Admin |
| **Type** | Functional Requirement |
| **Description** | This function allows the admin to delete and block the clinic from our system so we have not to deal with it anymore |
| **Pre-Condition** | Must has an account with admin permission |
| **Post-Condition** | Log in with his account |
| **Input** | Action from remove clinic button |
| **Output** | Notification of successful removing |

|  |  |
| --- | --- |
| **Code** | F12 |
| **Name** | Remove Hospital |
| **Role** | Admin |
| **Type** | Functional Requirement |
| **Description** | This function allows the admin to delete and block the hospital from our system so we have not to deal with it anymore |
| **Pre-Condition** | Must has an account with admin permission |
| **Post-Condition** | Log in with his account |
| **Input** | Action from remove hospital button |
| **Output** | Notification of successful removing |

|  |  |
| --- | --- |
| **Code** | F13 |
| **Name** | Add Hospital HR |
| **Role** | Admin |
| **Type** | Functional Requirement |
| **Description** | The admin uses this function after add a hospital department to add a person managing the hospital system  (human resource person) |
| **Pre-Condition** | Must has an account with admin permission |
| **Post-Condition** | Log in with his account |
| **Input** | Data of the hospital hr |
| **Output** | Notification of successful adding |

|  |  |
| --- | --- |
| **Code** | F14 |
| **Name** | Show Statistics |
| **Role** | Admin – lab – pharmacy – doctor |
| **Type** | Functional Requirement |
| **Description** | This is a general function allows actors to see statistics about the system but with different views for each actor |
| **Pre-Condition** | Must have an account |
| **Post-Condition** | Log in with the account |
| **Input** | Action from show statistics button |
| **Output** | Open statistics page |

|  |  |
| --- | --- |
| **Code** | F15 |
| **Name** | Upload X-Rays |
| **Role** | Lab |
| **Type** | Functional Requirement |
| **Description** | Get x-ray from a scanner and uploaded to the system and can be saved as pdf file where a doctor can access it and will much clearer |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with the account |
| **Input** | Action from press upload x-ray button |
| **Output** | A Pdf form of the x-rays |

|  |  |
| --- | --- |
| **Code** | F16 |
| **Name** | Add Analytics |
| **Role** | Lab |
| **Type** | Functional Requirement |
| **Description** | A feature helps the lap doctor add the analysis result of the patient to his/her account |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with the account |
| **Input** | Action from press add analytics button |
| **Output** | Add analytics done successfully |

|  |  |
| --- | --- |
| **Code** | F17 |
| **Name** | Check & Submit Result |
| **Role** | Lab – pharmacy |
| **Type** | Functional Requirement |
| **Description** | Check the data of the medicine and the analysis are correct and confirm it |
| **Pre-Condition** | Have an account and Patient data already entered |
| **Post-Condition** | Log in with the account |
| **Input** | Press confirm and submit button |
| **Output** | Accept/reject data entered |

|  |  |
| --- | --- |
| **Code** | F18 |
| **Name** | Add Patient Data |
| **Role** | Doctor |
| **Type** | Functional Requirement |
| **Description** | This general function allows the doctor to add the patient history or medical data and upload it in the system |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with the account |
| **Input** | Action from press add patient data button |
| **Output** | Add data done and uploaded successfully |

|  |  |
| --- | --- |
| **Code** | F19 |
| **Name** | Show Doctor Ranks |
| **Role** | Functional Requirement |
| **Type** | Doctor |
| **Description** | This function allows the doctor to see his rank in our system |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with the account |
| **Input** | Action from press show doctor rank button |
| **Output** | Ranking page will open |

|  |  |
| --- | --- |
| **Code** | F20 |
| **Name** | Block Doctors |
| **Role** | HR |
| **Type** | Functional Requirement |
| **Description** | This function helps the hr to delete the doctor from the hr's hospital that mean the doctor leave it and didn't work their anymore |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with the account |
| **Input** | Action from press block doctor button |
| **Output** | Notification with doctor removed successfully |

|  |  |
| --- | --- |
| **Code** | F21 |
| **Name** | Search |
| **Role** | HR |
| **Type** | Functional Requirement |
| **Description** | This function helps the hr to find the doctor in our system by writing his name |
| **Pre-Condition** | Must has an account |
| **Post-Condition** | Log in with the account |
| **Input** | Doctor name |
| **Output** | Matching results to the doctor name |

# system Nonfunctional Requirements

|  |  |
| --- | --- |
| **Name** | Dependability |
| **Type** | Product requirements |
| **Description** | Maintaining system & facing any kind of  Threats |

|  |  |
| --- | --- |
| **Name** | Testability |
| **Type** | Product requirements |
| **Description** | Testing system and code of functionality and  find bugs to ensure its performance |

|  |  |
| --- | --- |
| **Name** | Maintainability |
| **Type** | Product requirements |
| **Description** | The ability of Fixing and updating the system  to ensure its stability |

|  |  |
| --- | --- |
| **Name** | Portability |
| **Type** | Usability requirements |
| **Description** | The ability to use the software on different  Platform |

|  |  |
| --- | --- |
| **Name** | Robustness |
| **Type** | Usability requirements |
| **Description** | Handling errors, exceptions and invalid data |

|  |  |
| --- | --- |
| **Name** | Reliability |
| **Type** | Usability requirements |
| **Description** | System maintains its performance over long period of time |

## **3.1 usecase senario**

|  |  |
| --- | --- |
| Use case Name: | Register |
| Actors: | Lab Doctor |
| Basic Flow: | 1. Open Home page 2. Click on register button 3. Enter your information 4. Click register button |
| Pre-conditions: | Open Registration Page |
| Alternative flow: | 1. If Data entered are not vailed 2. The user re-enter information |
| Post-conditions: | The Data Saved and the user is Created successfully |

|  |  |
| --- | --- |
| Use case Name: | Login |
| Actors: | Lab Doctor |
| Basic Flow: | 1. Open Home page 2. Click on Login button 3. Enter your information 4. Click Login button |
| Pre-conditions: | Open Login Page and Enter email and password |
| Alternative flow: | 1. If Data entered are not vailed 2. The user re-enter information |
| Post-conditions: | The Data are accepted, and the user is logged in successfully and Send to the home page |

|  |  |
| --- | --- |
| Use case Name: | Show Lab Data |
| Actors: | Lab Doctor |
| Basic Flow: | 1. Open Home page 2. Click on Login button 3. Enter your information Click Login button 4. Go to Profile 5. Press Scan QR 6. Press View information Button |
| Pre-conditions: | Must be logged In and Scan QR |
| Alternative flow: | 1. QR not Scanned Correctly 2. Rescan QR Code |
| Post-conditions: | The Lab Doctor access lab info about specific patient |

|  |  |
| --- | --- |
| Use case Name: | Add analytics |
| Actors: | Lab Doctor |
| Basic Flow: | 1. Open Home page 2. Click on Login button 3. Enter your information Click Login button 4. Go to Profile 5. Scan QR Code 6. Press on Add patient Data 7. Enter Lab result Data 8. Press Add Data |
| Pre-conditions: | Must be Logged In /Scan QR Code |
| Alternative flow: | 1. The QR not Scanned correctly 2. Rescan the Scan The QR 3. Data are Not Valid 4. Re-enter lab result Data |
| Post-conditions: | The patient Data is updated, and the lap result Data are add to his account |

|  |  |
| --- | --- |
| Use case Name: | Upload X-rays |
| Actors: | Lab Doctor |
| Basic Flow: | 1. Open Home page 2. Click on Login button 3. Enter your information Click Login button 4. Go to Profile 5. Scan QR Code 6. Press on Add patient Data 7. Press on choose X-rays 8. Choose X-rays from File 9. Press Upload x-rays |
| Pre-conditions: | Must be Logged In /Scan QR Code |
| Alternative flow: | 1. The QR not Scanned correctly 2. Rescan the Scan The QR 3. The file not the supported format 4. Re upload the x-rays |
| Post-conditions: | The file uploaded successfully to the patient account |

|  |  |
| --- | --- |
| Use case Name: | Lab Statistics |
| Actors: | Lab Doctor |
| Basic Flow: | 1. Open Home page 2. Click on Login button 3. Enter your information Click Login button 4. Go to Profile 5. Press on Show Lap Statistics |
| Pre-conditions: | Must be Logged In |
| Alternative flow: | n/a |
| Post-conditions: | The Lab Doctor will view statistics about all lab results |

|  |  |
| --- | --- |
| Use case Name: | Check and Submit |
| Actors: | Lab Doctor |
| Basic Flow: |  |
| Pre-conditions: |  |
| Alternative flow: |  |

|  |  |
| --- | --- |
| Use case Name: | Log in |
| Actors: | HR |
| Basic Flow: | 1. Open Home page 2. Click on Login button 3. Enter required data 4. Click GO button |
| Pre-conditions: | Open login Page |
| Alternative flow: | 1. If Data entered are incorrect  2. HR re-enter data required |
| Post-conditions: | The Data entered, checked with real data and go to HR’s home page |

|  |  |
| --- | --- |
| Use case Name: | Register Doctor |
| Actors: | HR |
| Basic Flow: | 1. Open Home page   2.Click on Login button  3.Enter required data and Click Login button  4.Go to his home page   1. Press add/register Doctor   6.add all doctor data required |
| Pre-conditions: | Open HR home Page |
| Alternative flow: | 1.If Data entered are incorrect  2.re-enter doctor data required |
| Post-conditions: | A new Doctor added successfully on our system |

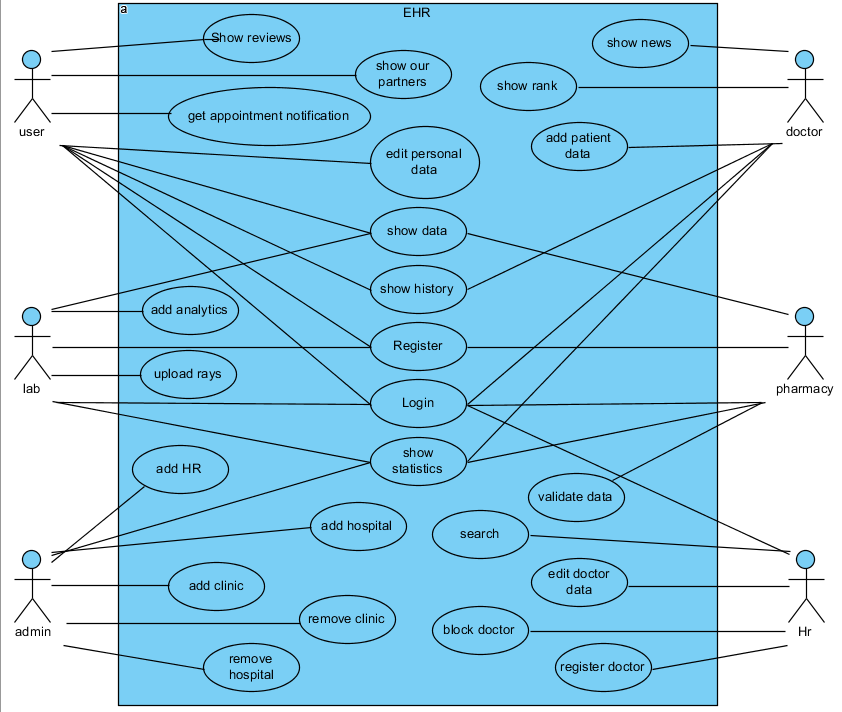
|  |  |
| --- | --- |
| Use case Name: | Block Doctor |
| Actors: | HR |
| Basic Flow: | 1.Open Home page  2.Click on Login button  3.Enter required data and Click Login button  4.Go to his home page  5.Press delete/Block Doctor  6.determine doctor  7.press ok |
| Pre-conditions: | Open HR home Page |
| Alternative flow: |  |
| Post-conditions: | Selected doctor deleted successfully |

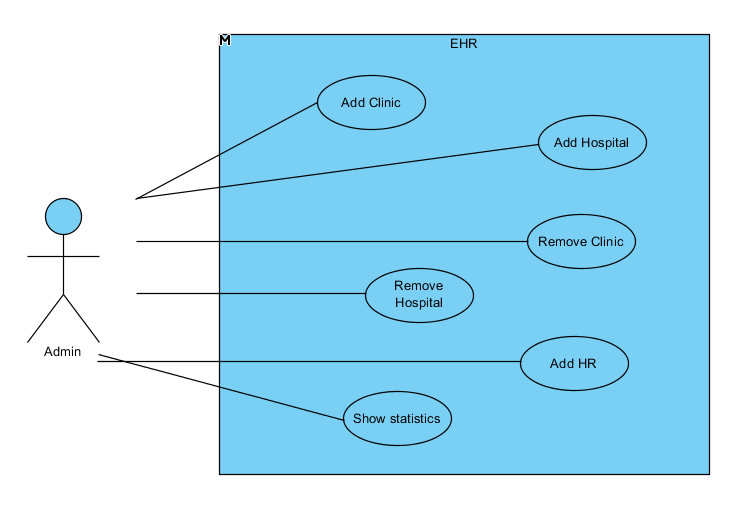
|  |  |
| --- | --- |
| Use case Name: | Search |
| Actors: | HR |
| Basic Flow: | 1.Open Home page  2.Click on Login button  3.Enter required data and Click Login button  4.Go to his home page  5.Press search Doctor button  6.write doctor name and select it  7.go to this doctor page |
| Pre-conditions: | Open HR home Page |
| Alternative flow: | 1.If Data entered are not valid  2.HR re-enter doctor name |
| Post-conditions: | All Doctors names which follow this name are appeared |

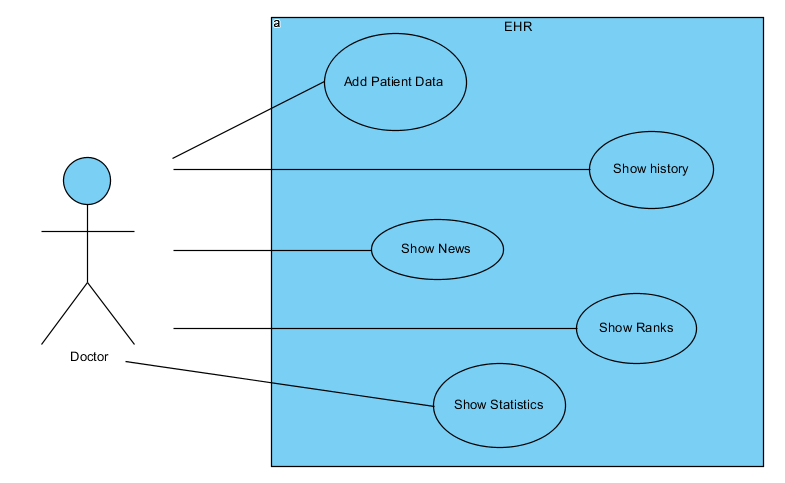
|  |  |
| --- | --- |
| Use case Name: | Edit personal data |
| Actors: | HR |
| Basic Flow: | 1.Open Home page  2.Click on Login button  3.Enter required data and Click Login button  4.Go to his home page  5.Press Edit personal data  6.enter new edit data  7.press save/submit/ok |
| Pre-conditions: | Open HR home Page |
| Alternative flow: | 1.If Data entered are not valid  2.HR re-enter information and repeat it again |
| Post-conditions: | The new Data Saved successfully and appear for him |

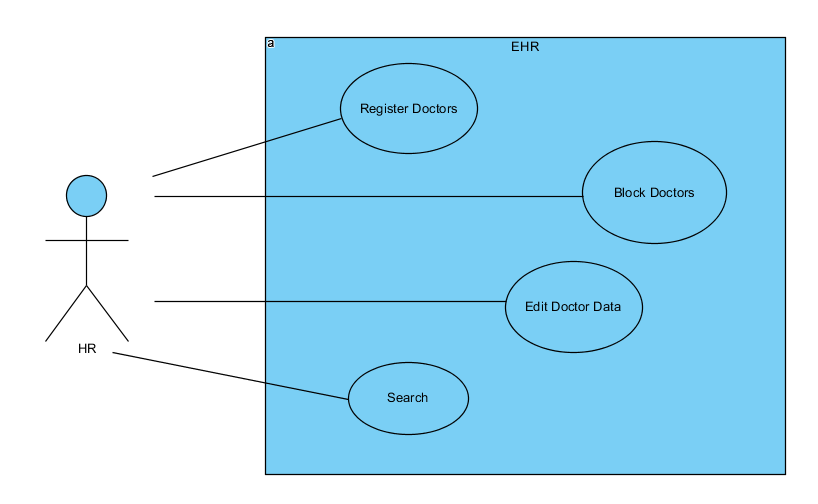
# Chapter 3 : project "Design "

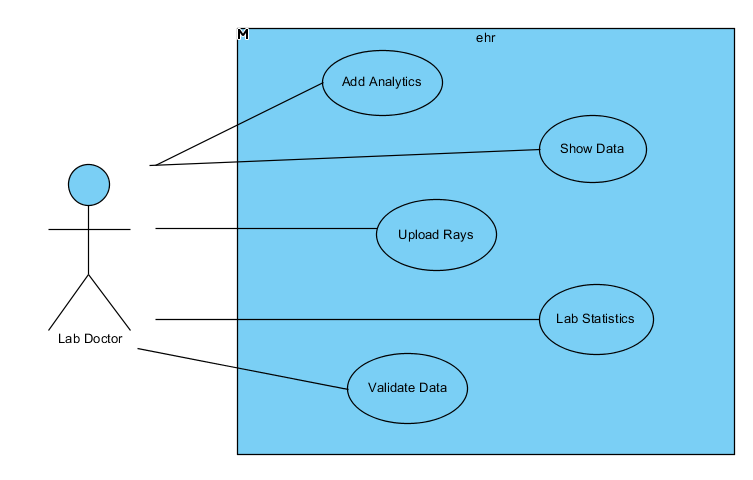
## **3.1 usecase diaram**

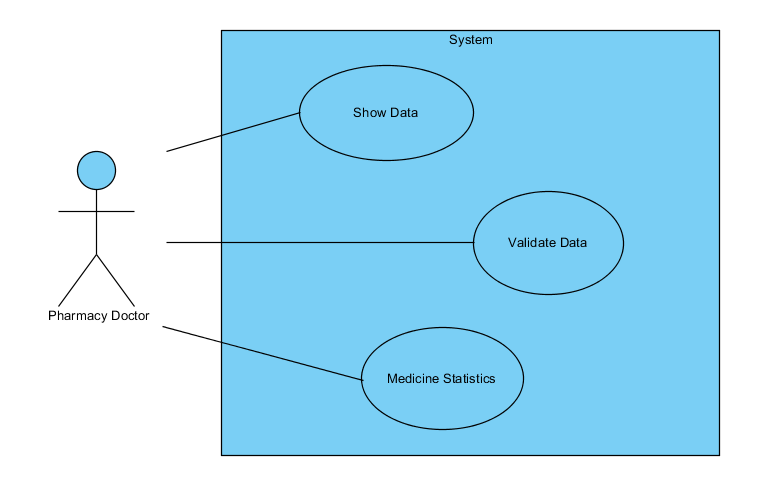






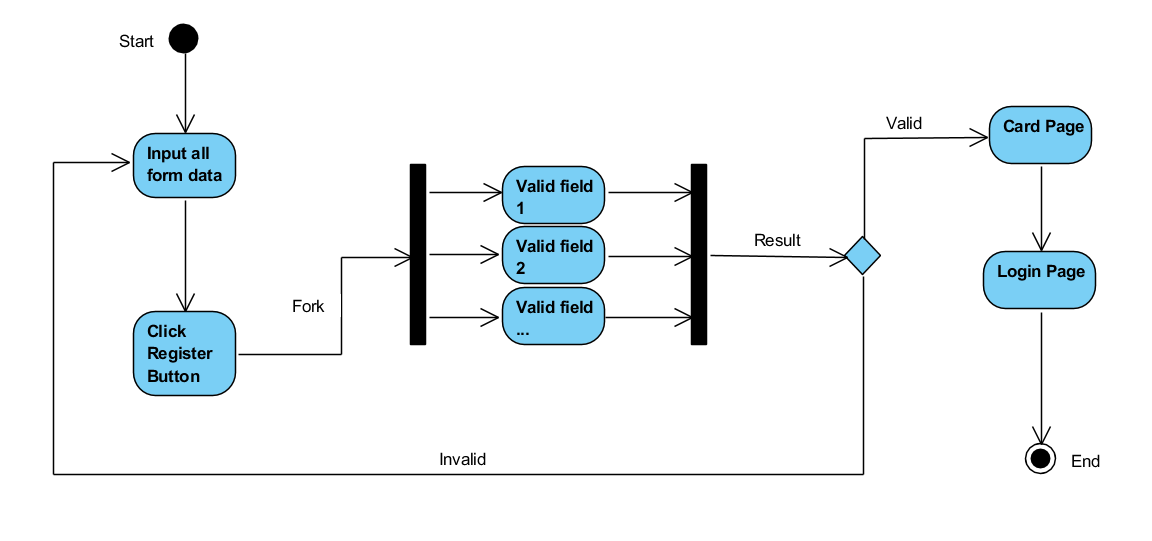


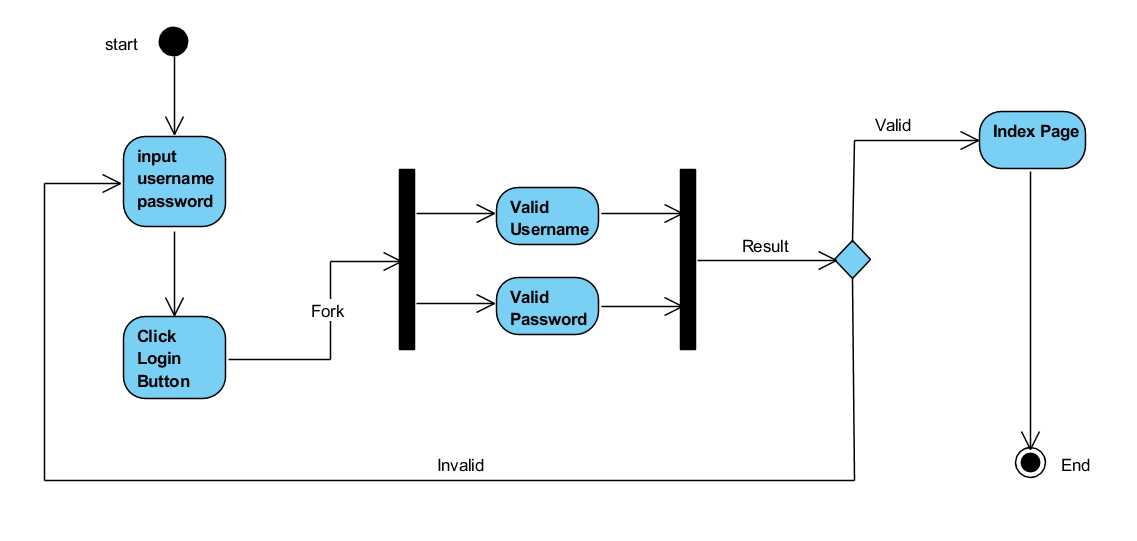


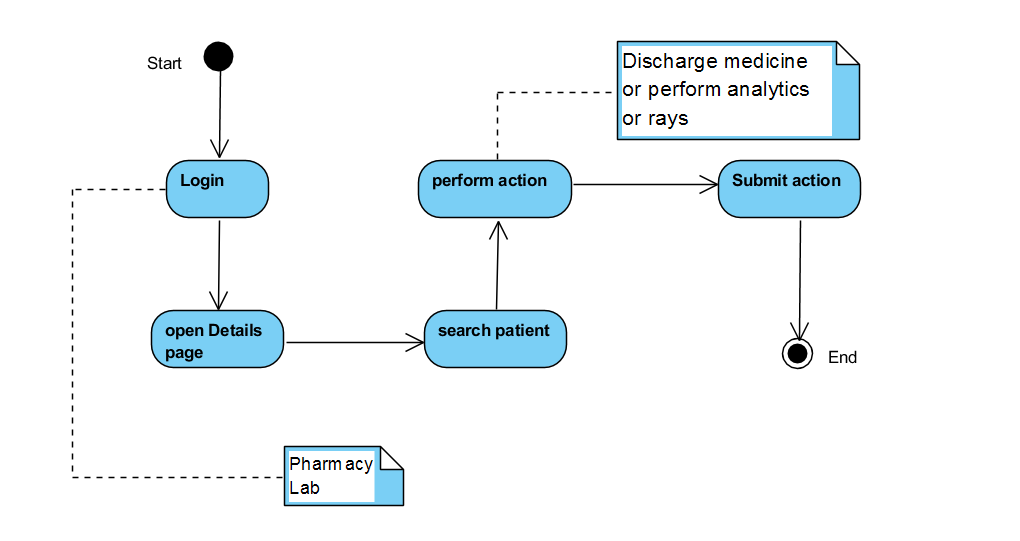


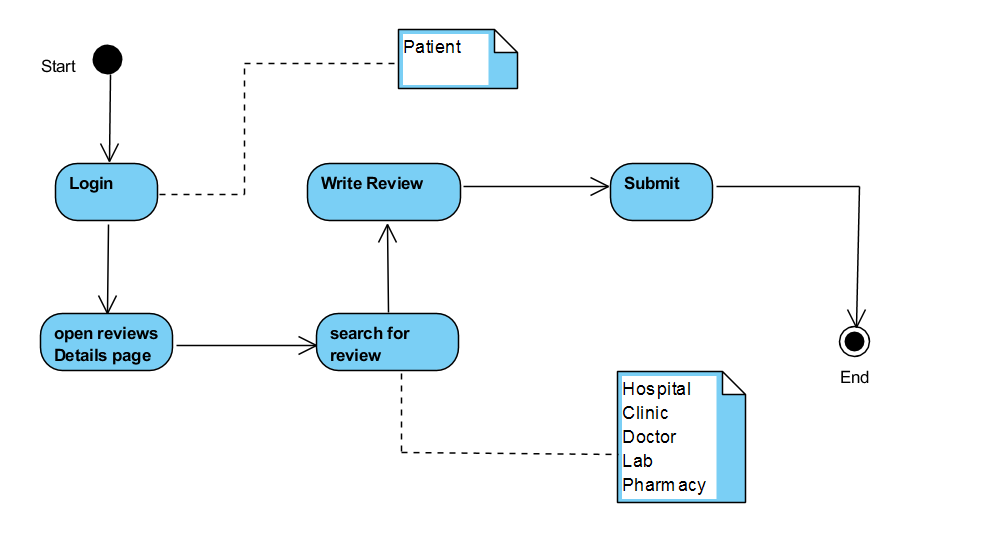


## **3.2 activity diaram**



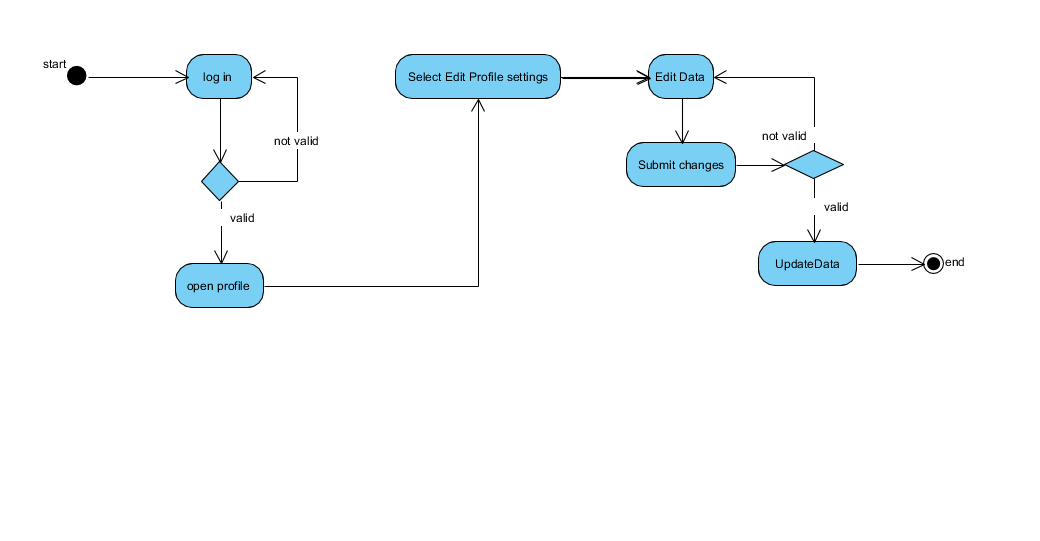


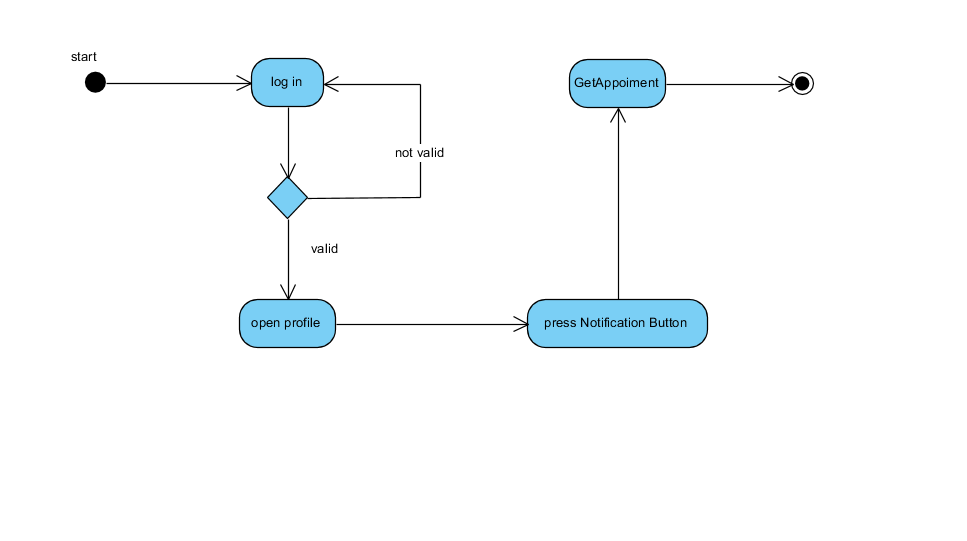


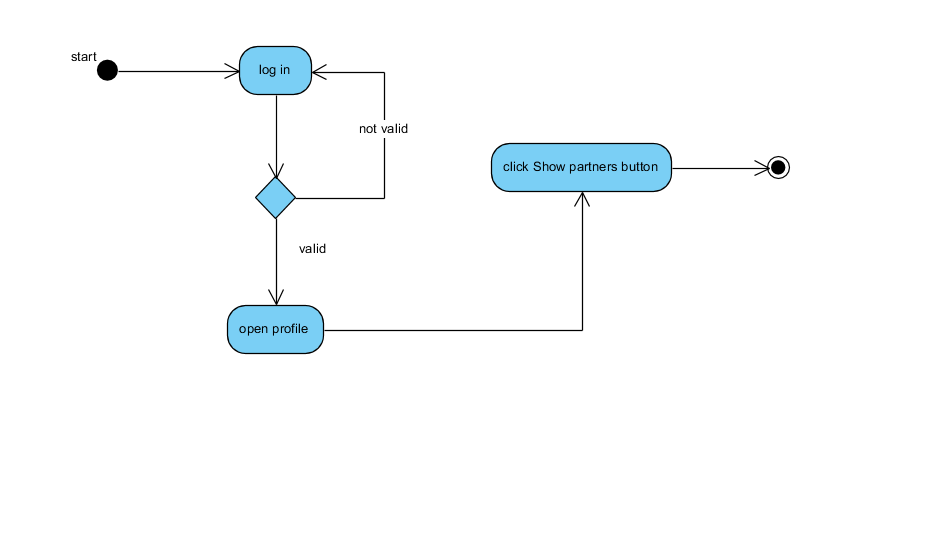


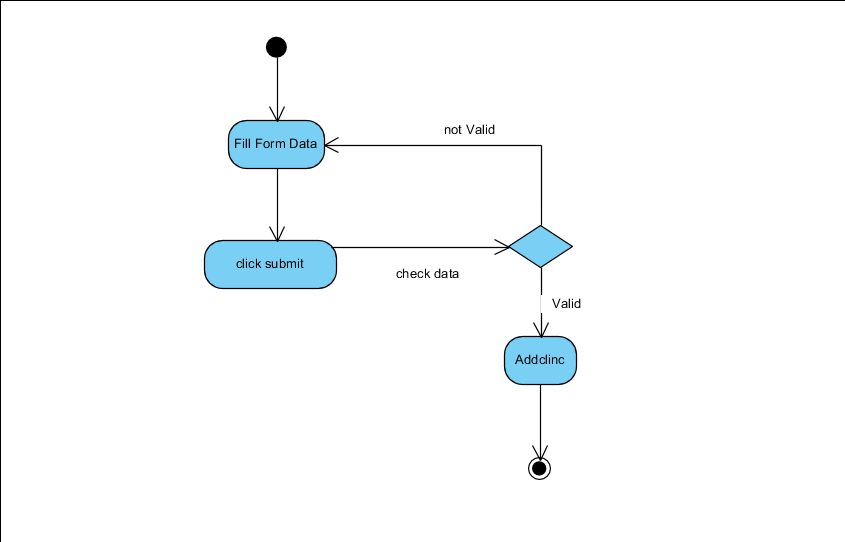


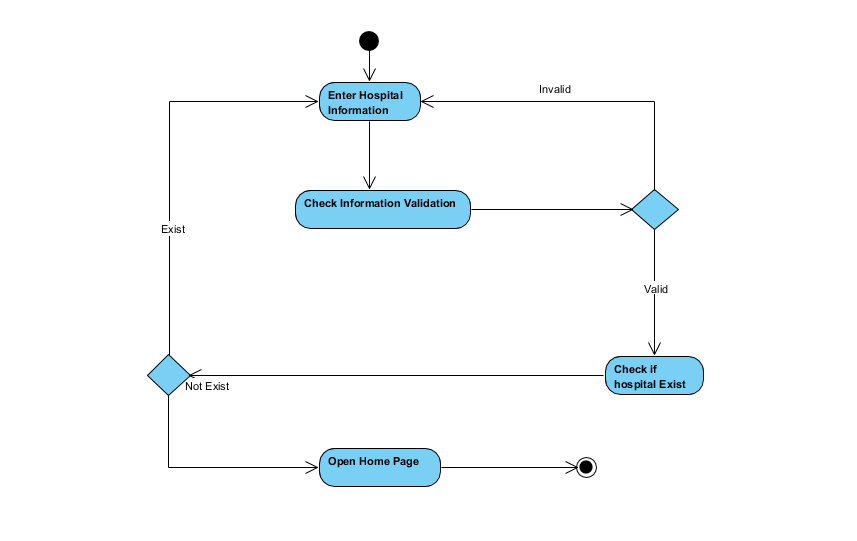


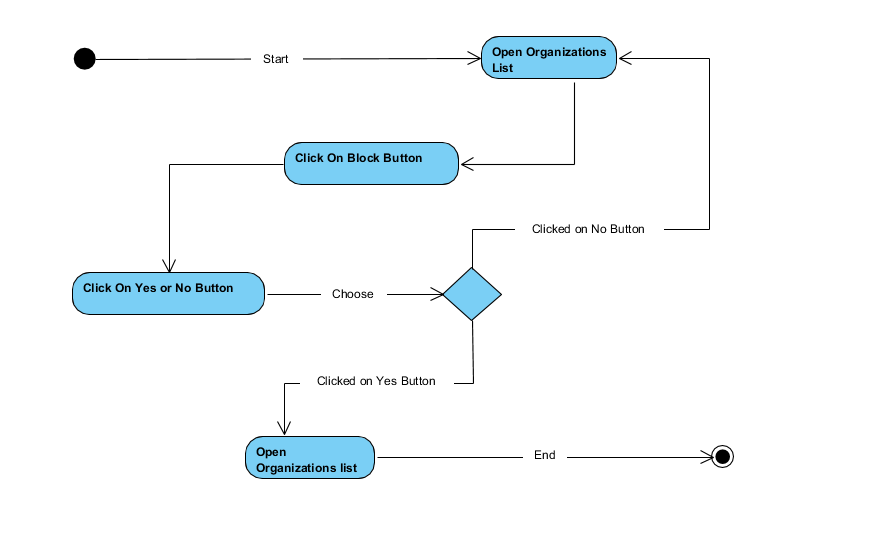


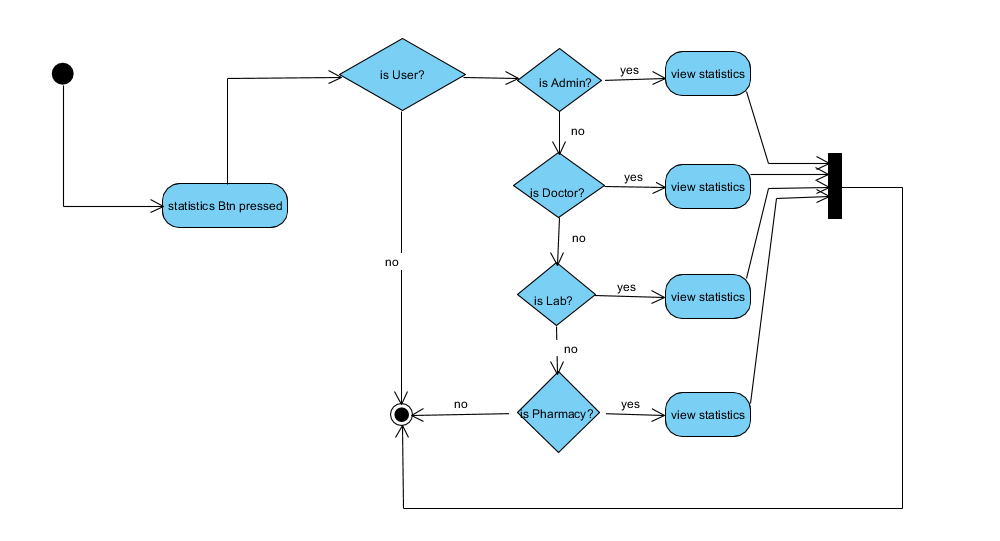


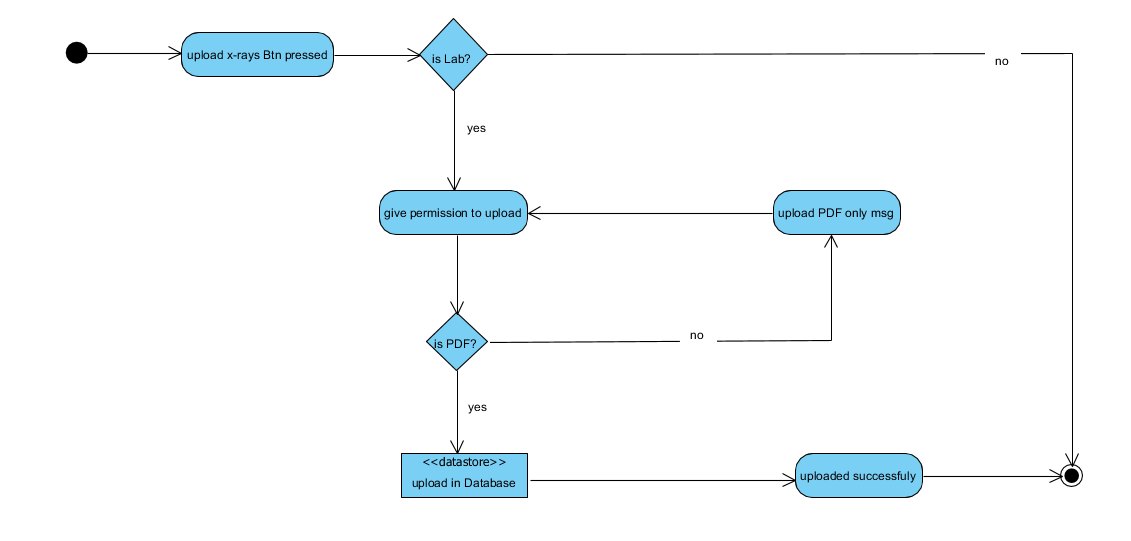




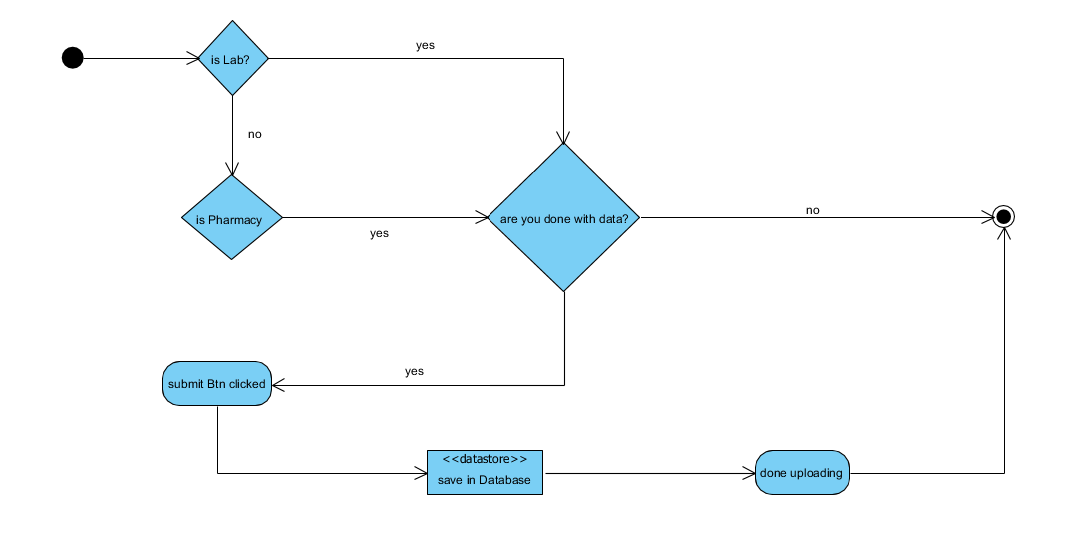


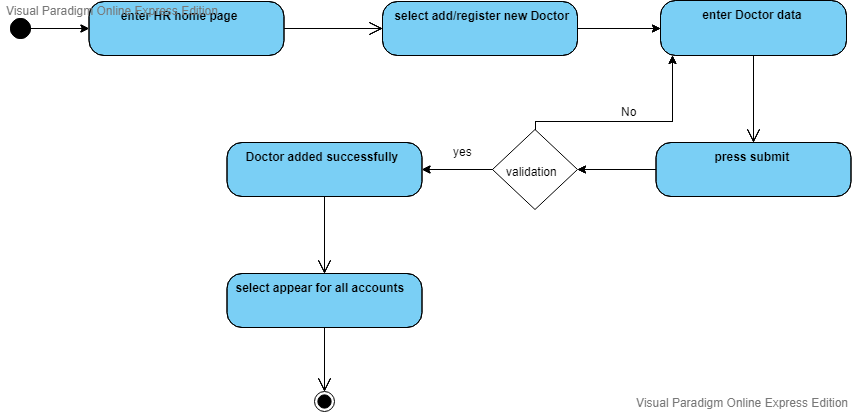


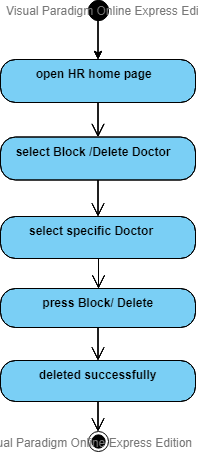


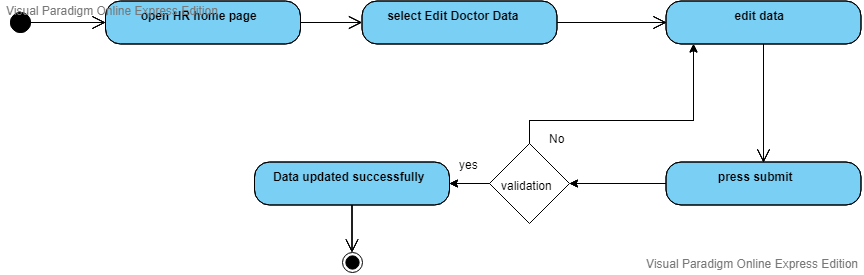


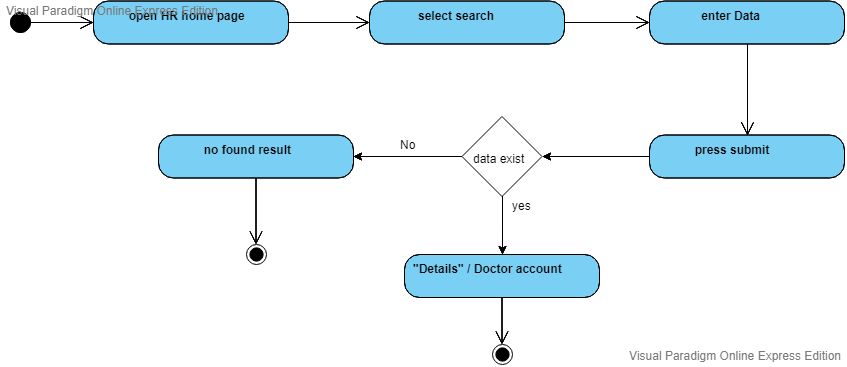




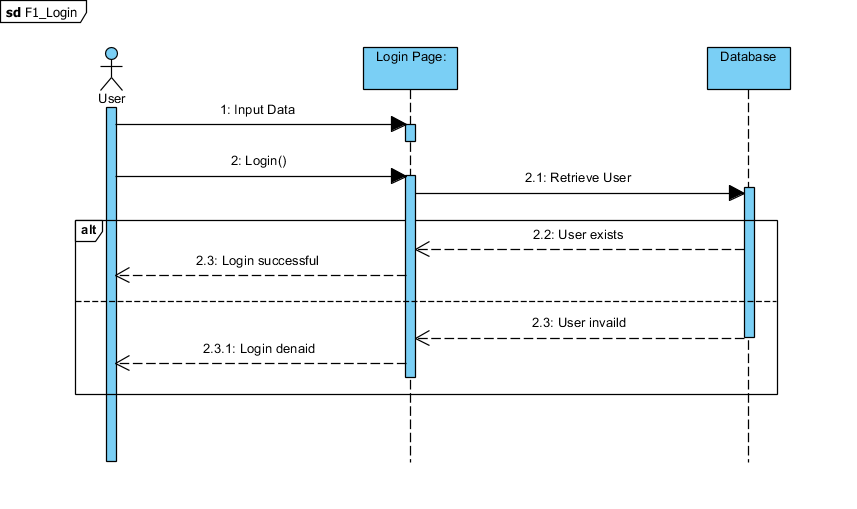


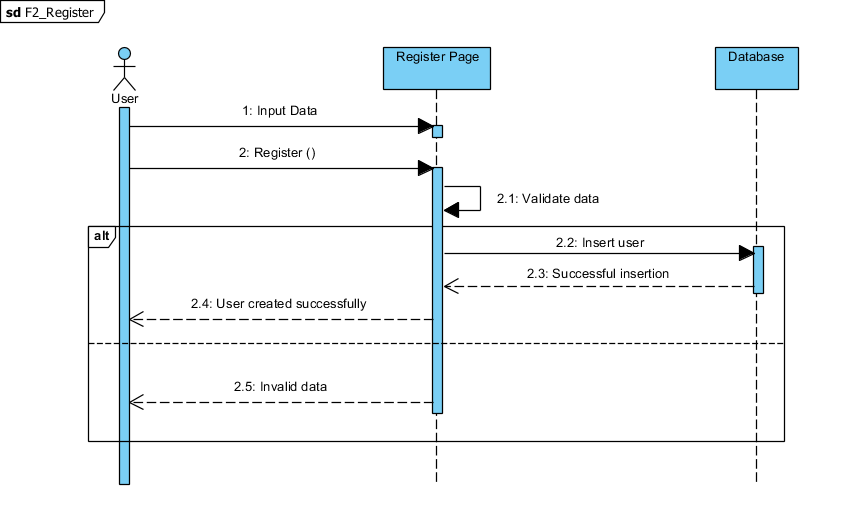


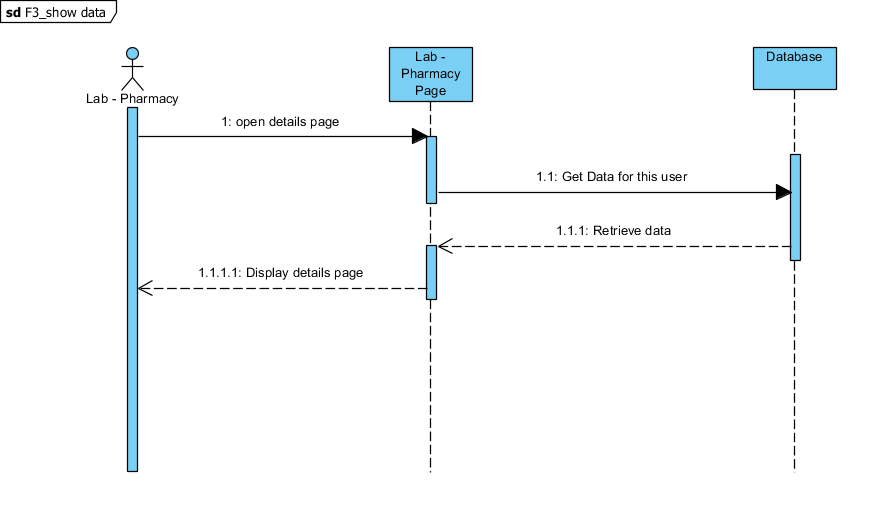


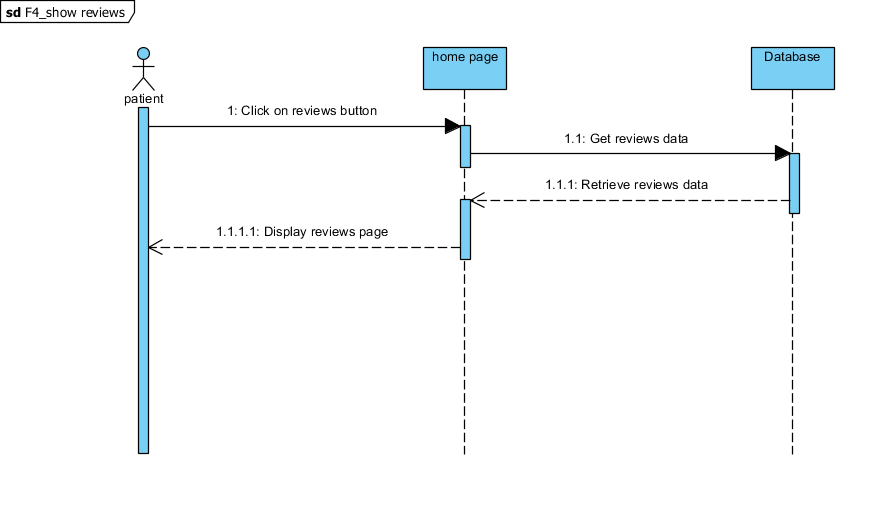


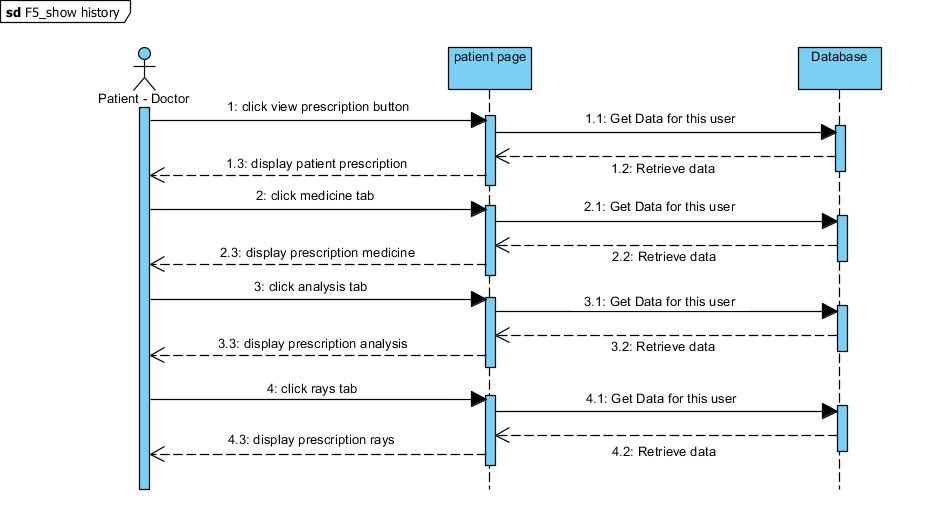
## **3.3 sequence diaram**

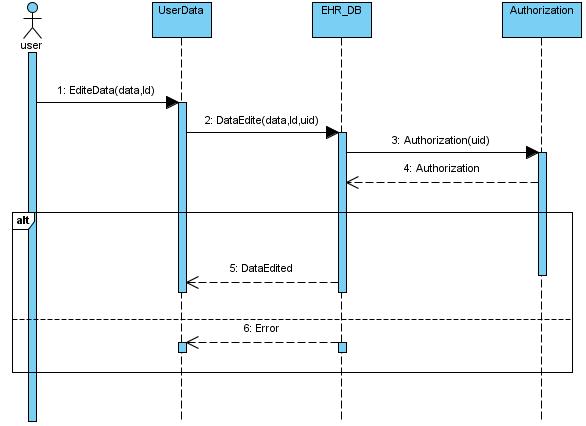


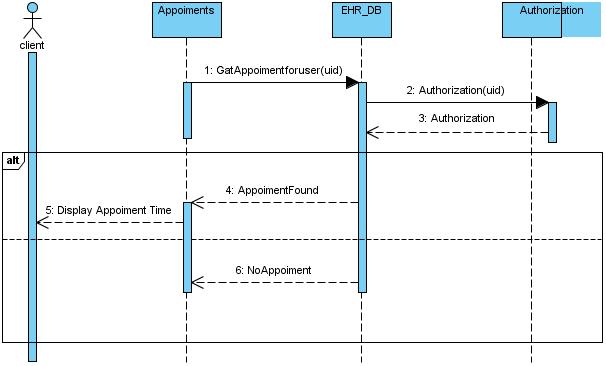


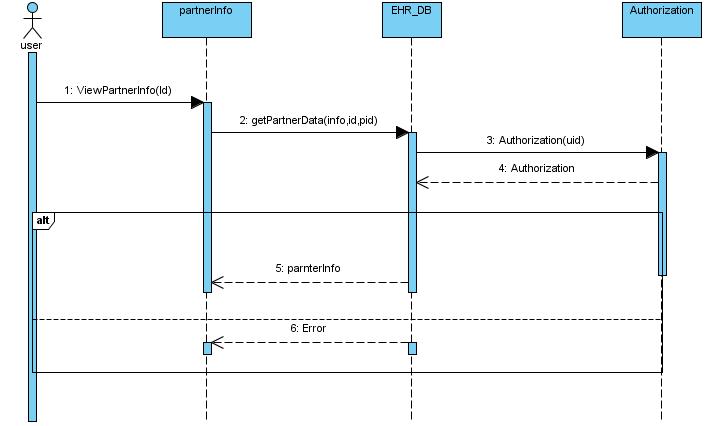


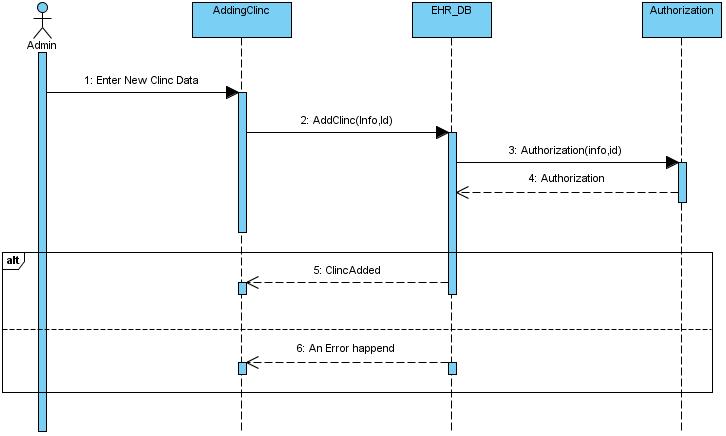


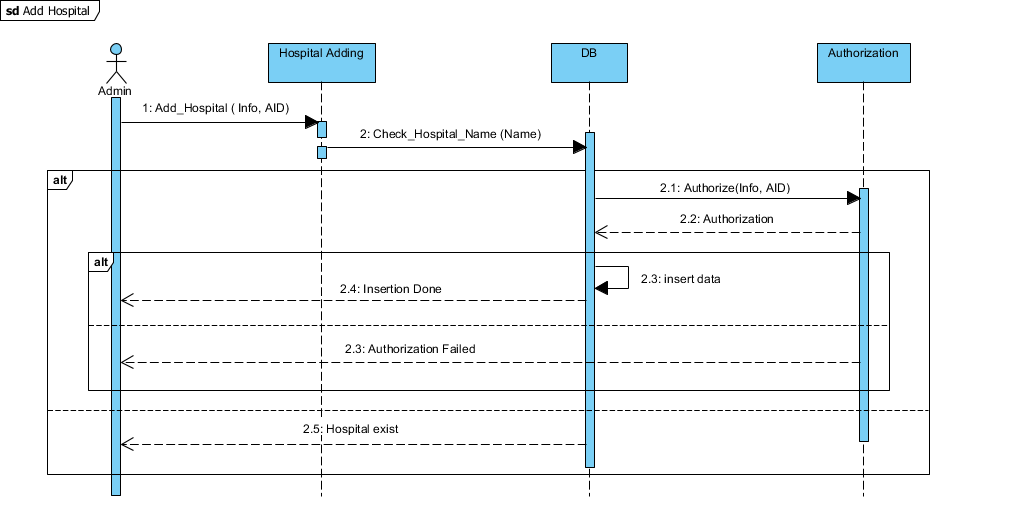


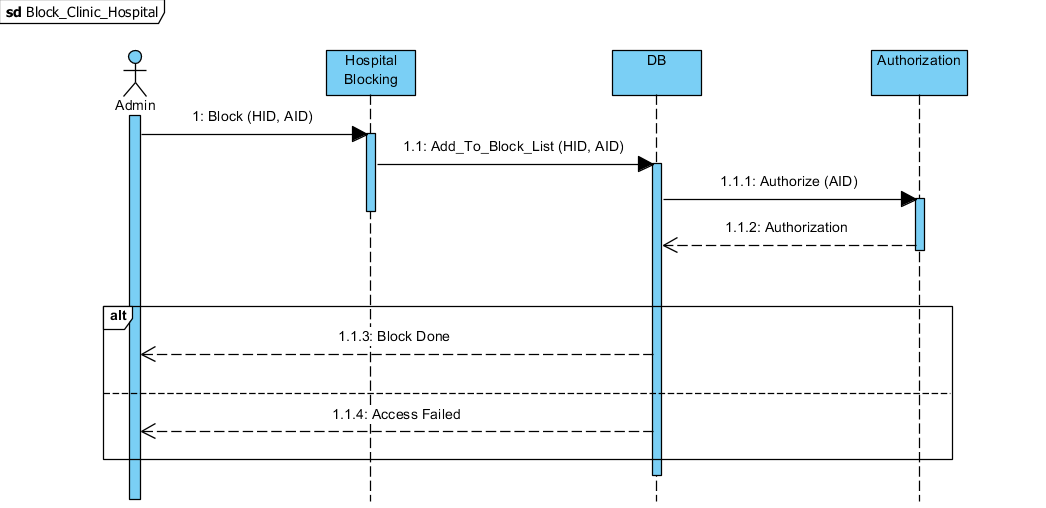


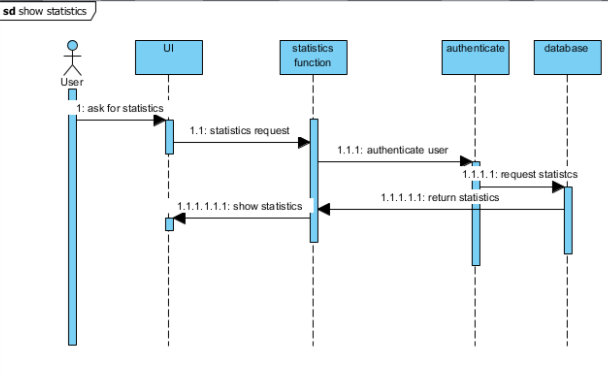


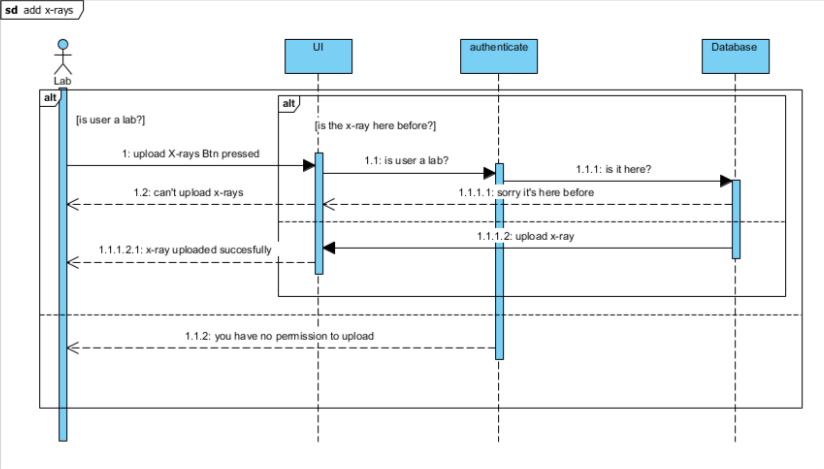


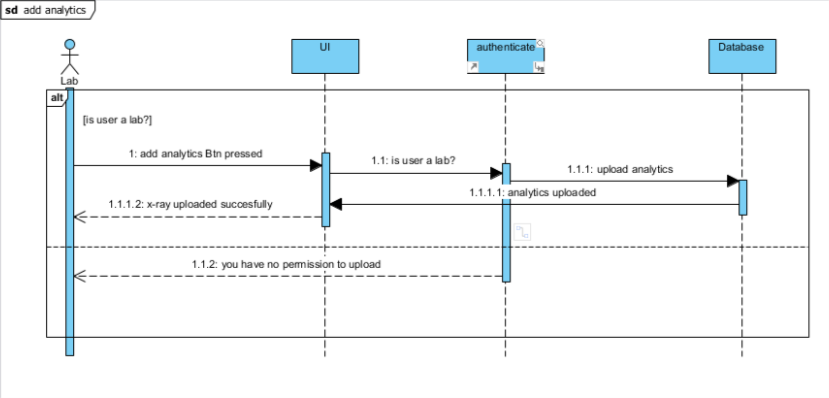


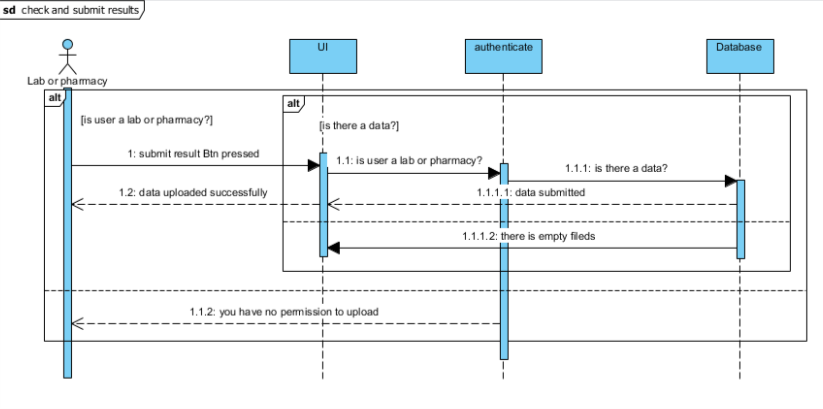


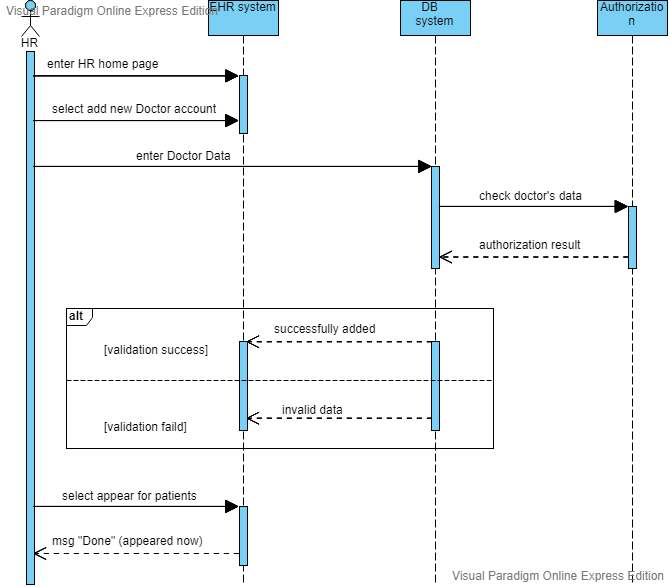


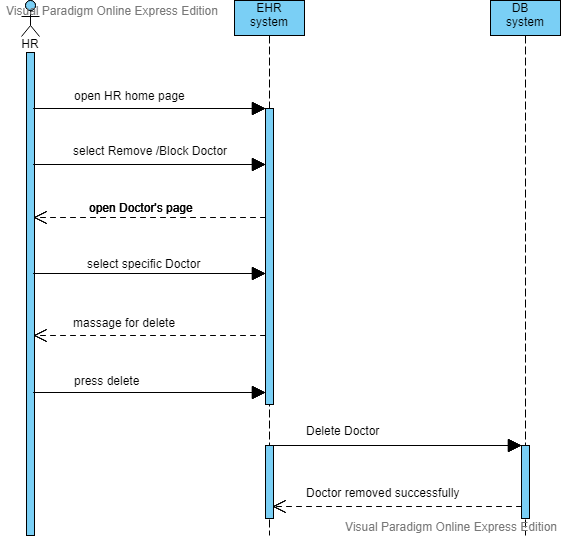




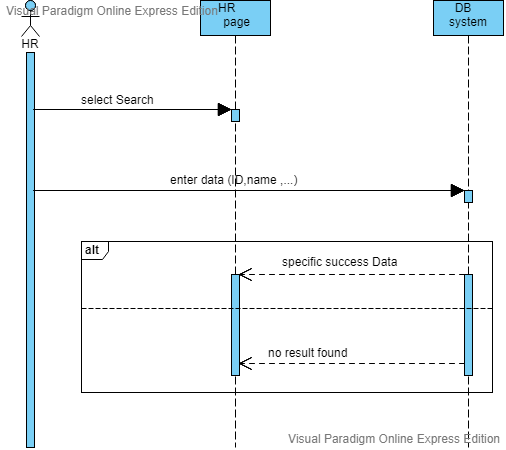




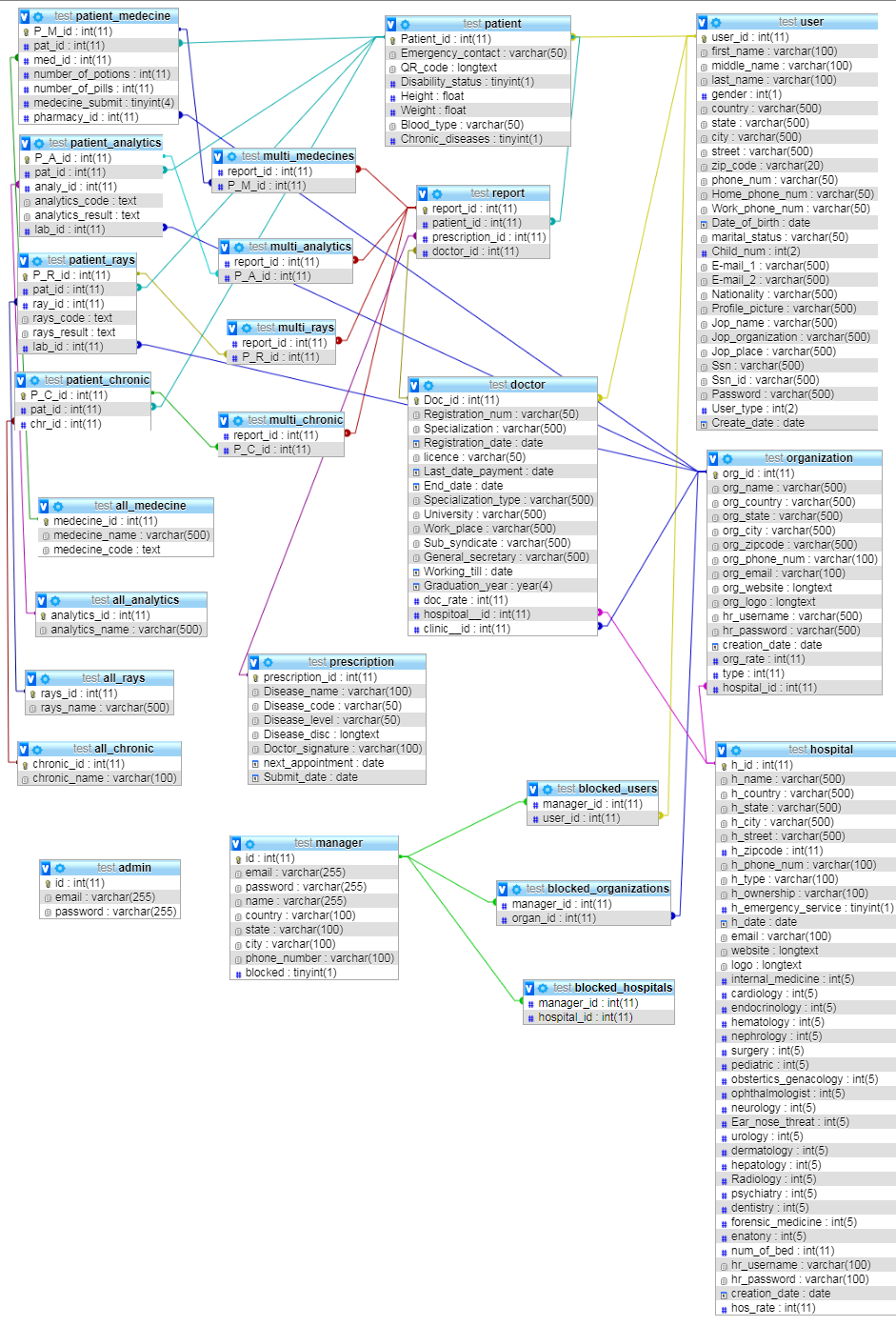






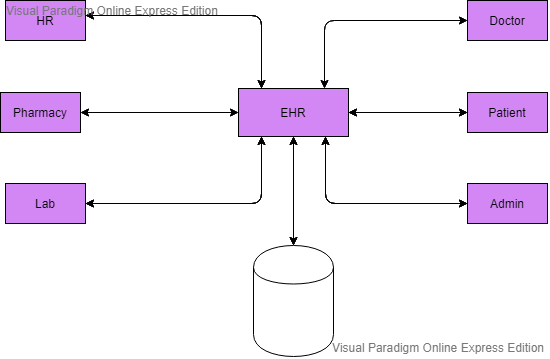


## **3.4 database schema**



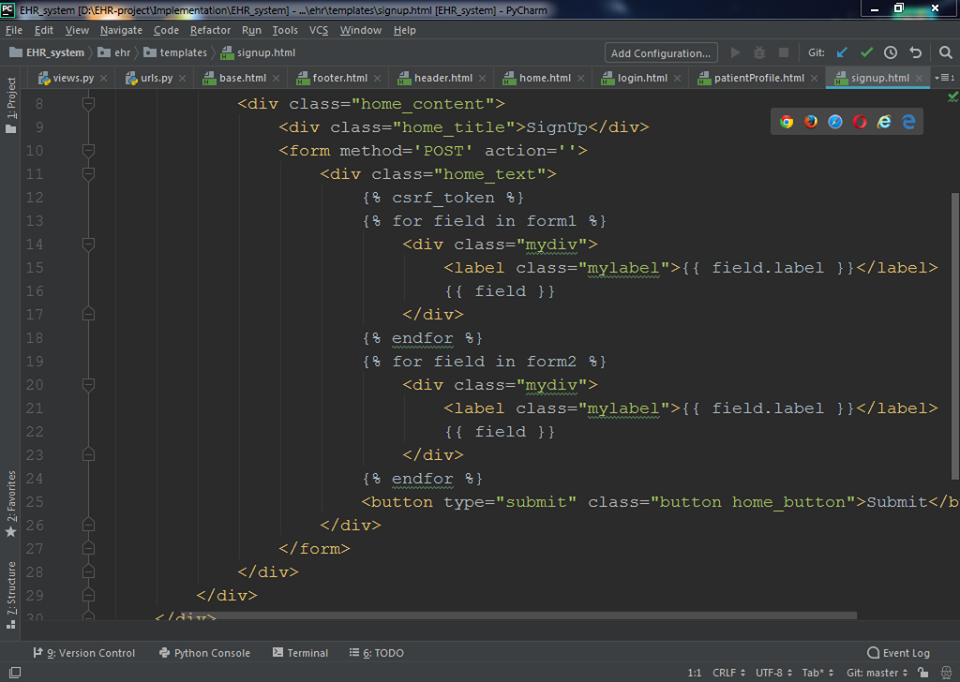
# Chapter 4 : project "implemention "

## **4.1 hardware archeticture**



## **4.2 software archeticture**

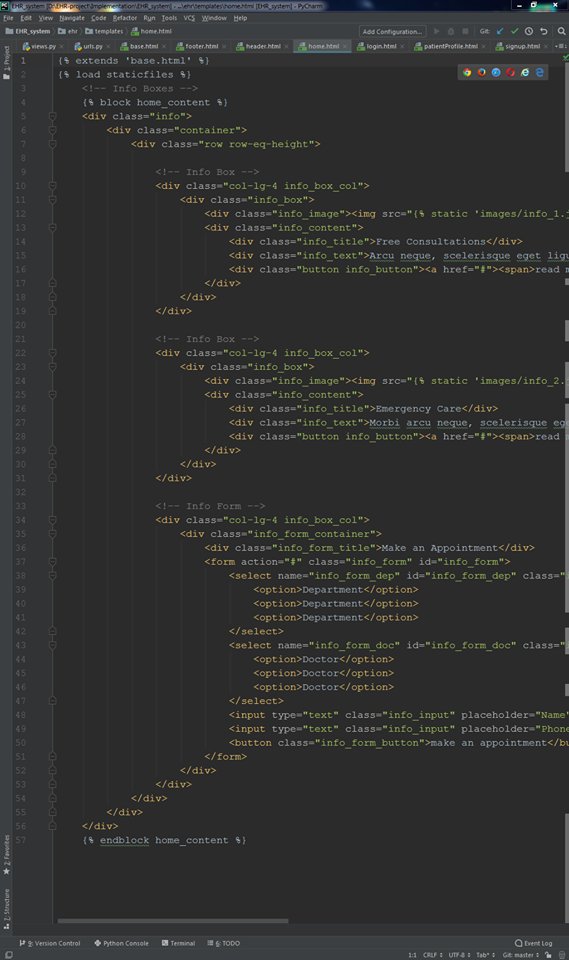
1. Sign up



1. Login



1. Home



## **5. TESTING**

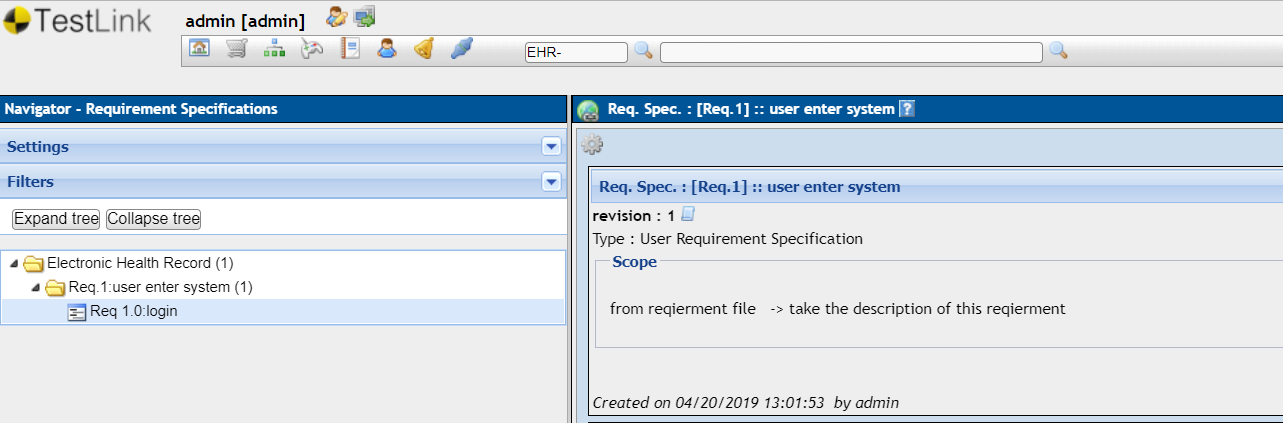
1. Project

## 

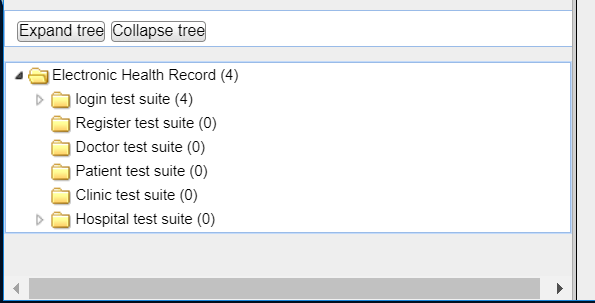
2. Test Plan



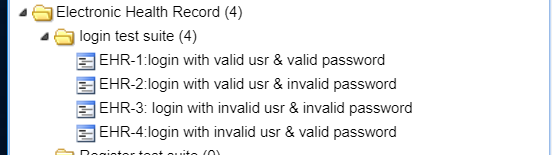
3. Test Requirement

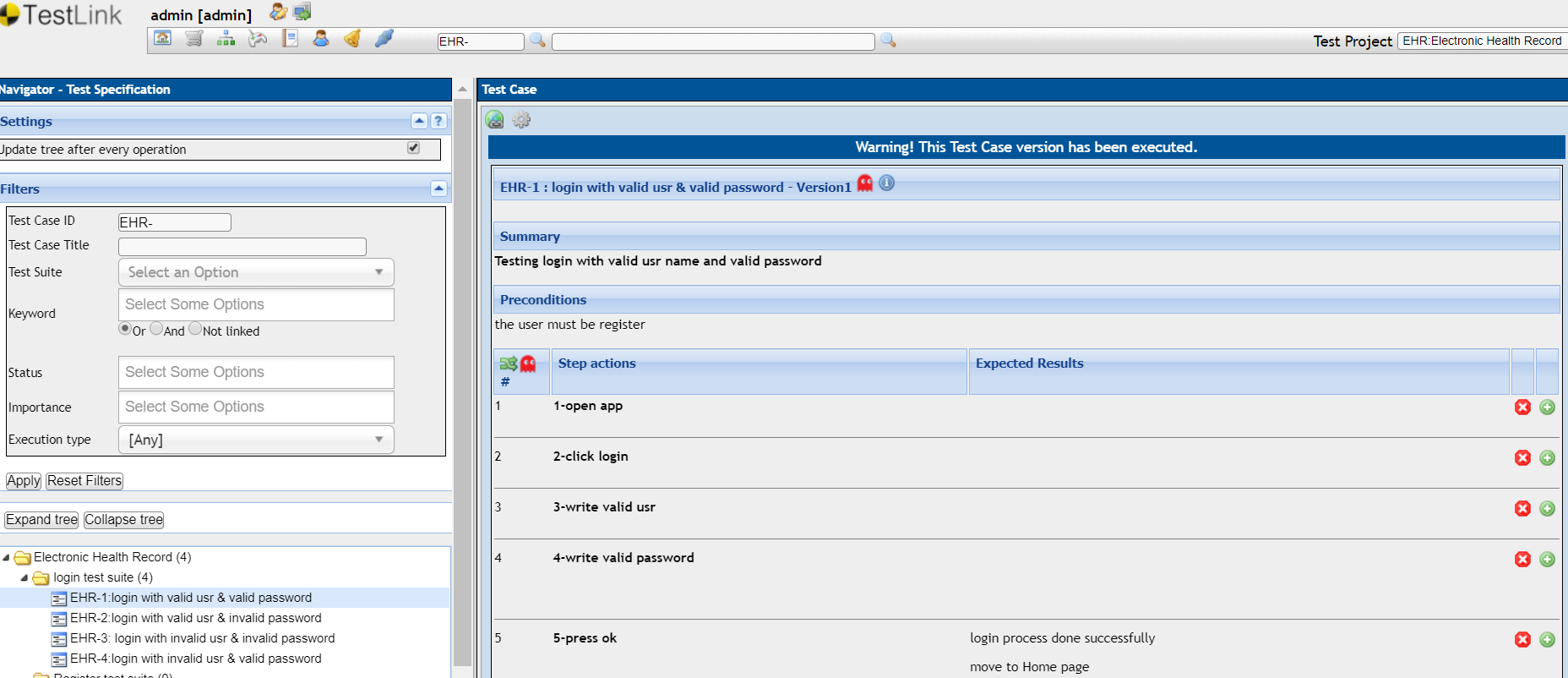


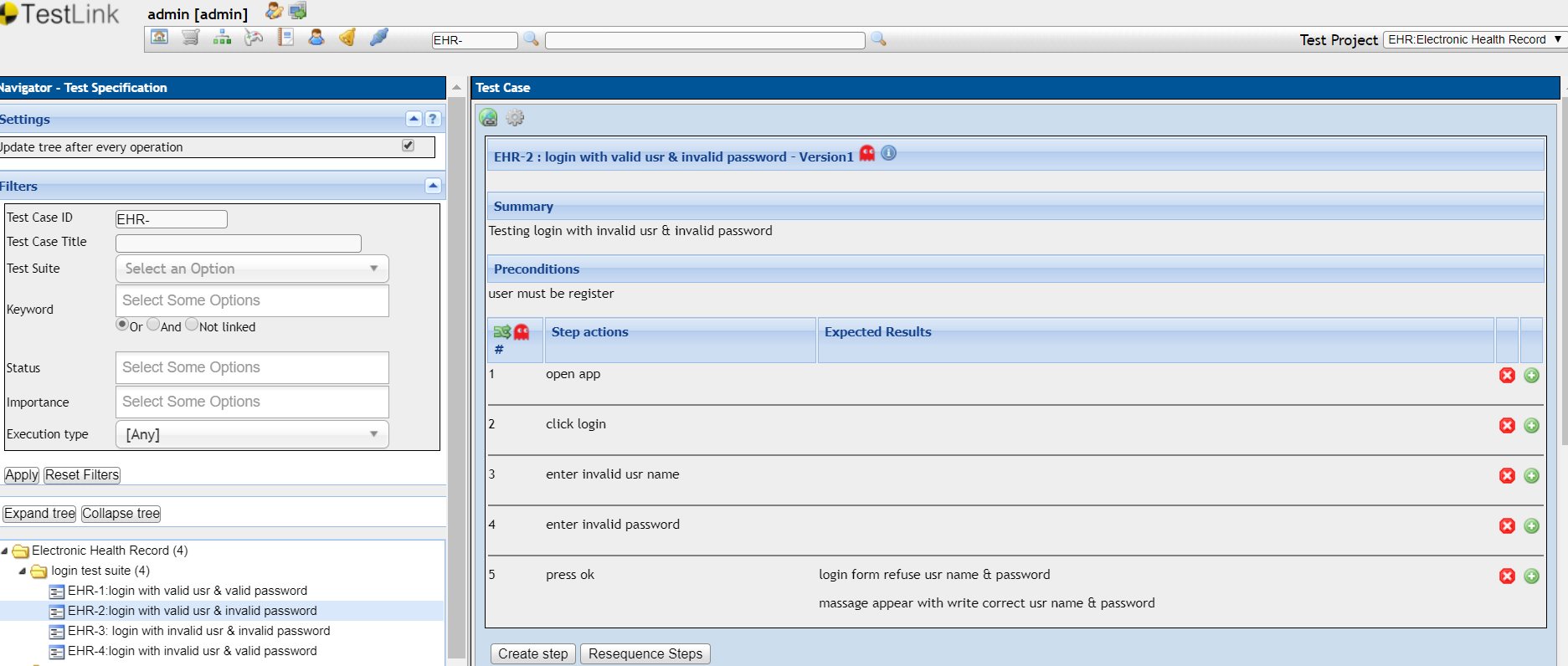
4. Test Suite

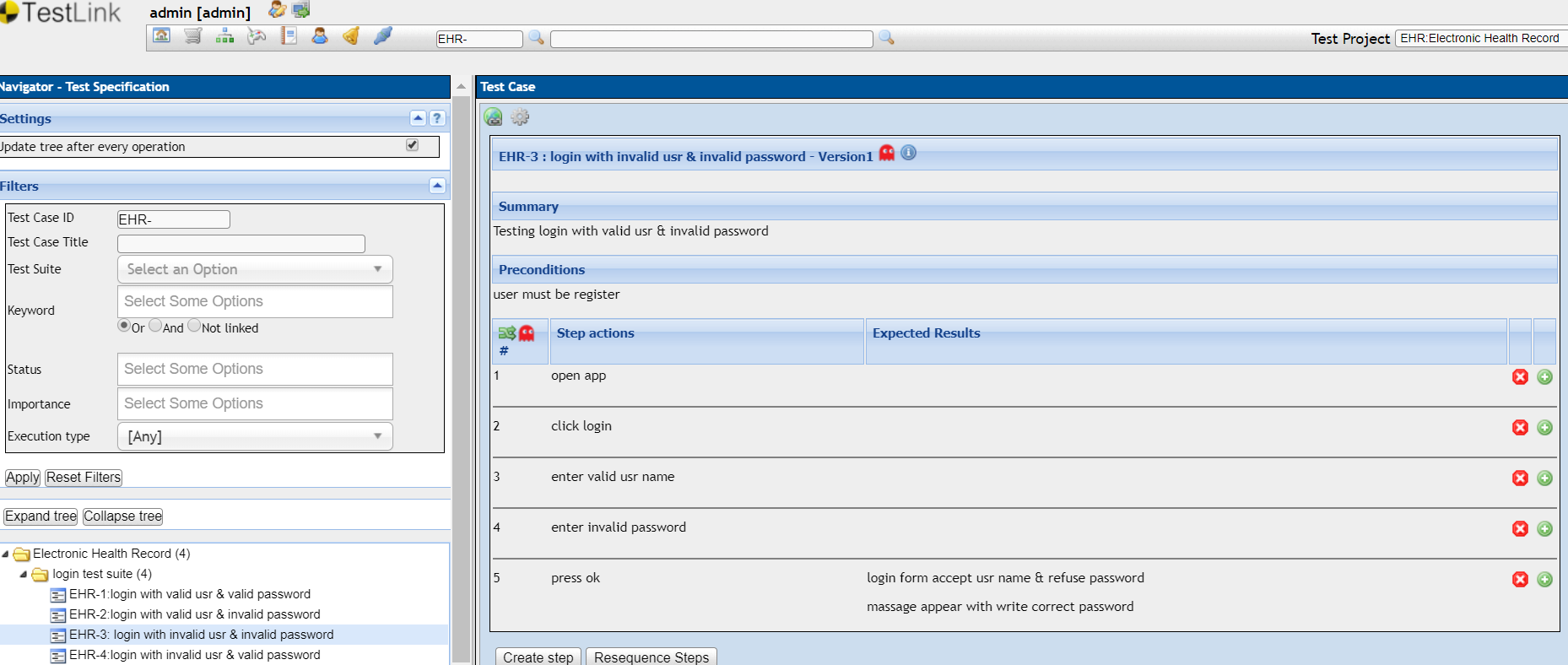


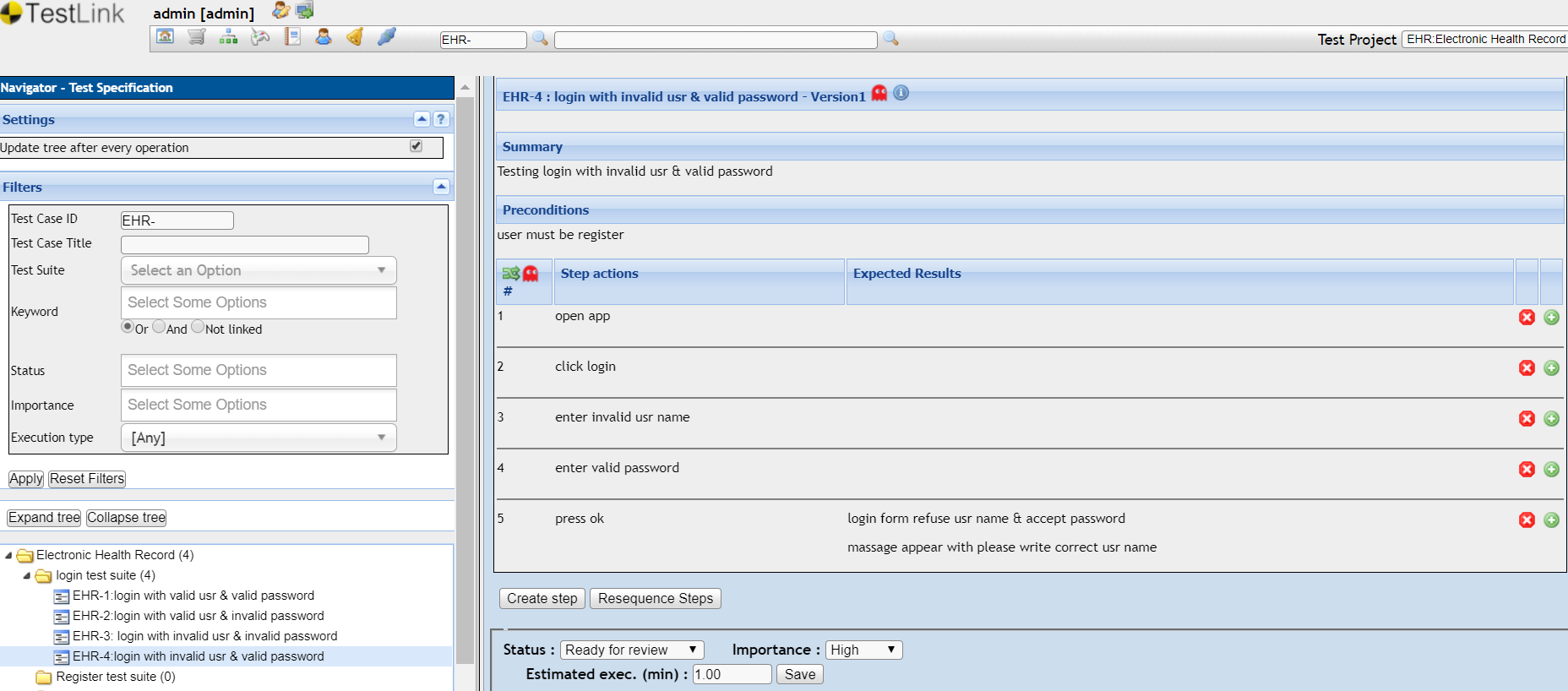
5. Test Case



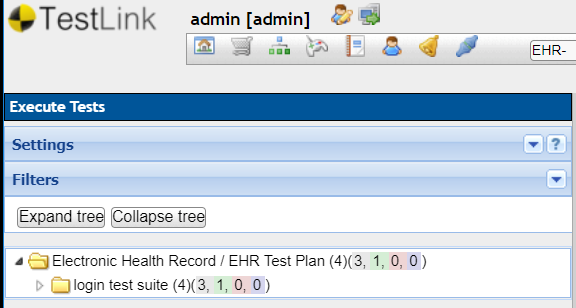




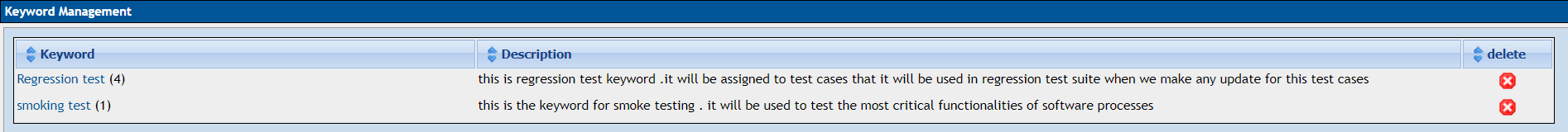




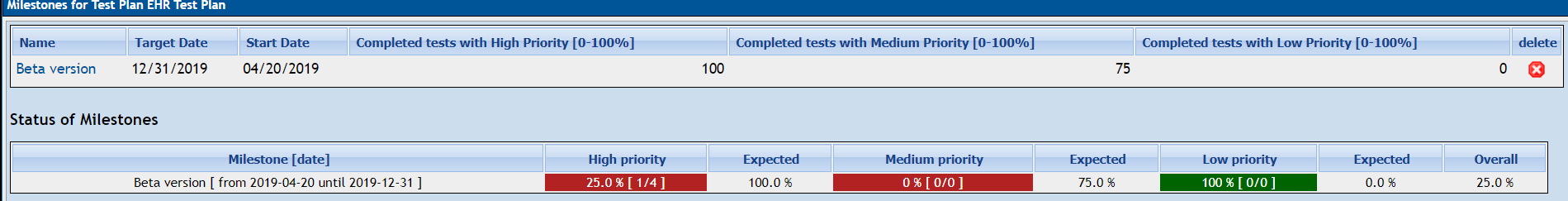
1. Test Execute



1. Test Management



1. Test Reports & Matrix





1. Test Results

