Care & Cure Hospital

Requirements Specification and Analysis

1.0

17.11.2017

Begüm Mina Bilgin

Ekin Naz Eren

Gamze Saraç

Süleyman Murat Yiğit

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc431983052)

[1.1. Purpose of the System 1](#_Toc431983053)

[1.2. Scope of the System 1](#_Toc431983054)

[1.3. Objectives and Success Criteria of the Project](#_Toc431983055) 2

[1.4. Definitions, Acronyms, and Abbreviations](#_Toc431983056) 2

[1.5. Overview](#_Toc431983057) 2

[2. Current System](#_Toc431983058) 3

[3. Proposed System](#_Toc431983059) 4

[3.1. Overview](#_Toc431983060) 4

[3.2. Functional Requirements](#_Toc431983061) 4

[3.3. Nonfunctional Requirements](#_Toc431983062) 5

[Usability](#_Toc431983063) 5

[Reliability](#_Toc431983064) 5

[Performance](#_Toc431983065) 5

[Supportability](#_Toc431983066) 5

[Implementation](#_Toc431983067) 5

[Interface](#_Toc431983068) 5

[Packaging](#_Toc431983069) 5

[Legal](#_Toc431983070) 5

[3.4. System Models](#_Toc431983071) 6

[Scenarios](#_Toc431983072) 6-9

[Use case model](#_Toc431983073) 10-21

[Object model 22](#_Toc431983074)

[Dynamic model 2](#_Toc431983075)2-25

[User interface—navigational paths and screen mock-ups](#_Toc431983076) 26-28

## 3.5 Project Schedule ………………………………………………………………………29.

[4. Glossary](#_Toc431983077) 29

[5. References](#_Toc431983078) 29

REQUIREMENTS ANALYSIS DOCUMENT[1]

# Introduction

## Purpose of the System

The purpose of the system is to give the users, which are the hospital staff (mainly doctors) and patients an efficient, mutual way to manage their appointments online.

The online system provides the functionality of taking and managing appointments swiftly, without the patient having to go to the hospital or make a call, and doctors to

see their appointments and update the patients test results.

* The system provides functionalities to patients such as; viewing and taking appointments from the selected doctors provided their schedule is available, browsing doctors from respective departments and viewing the day and time where the doctors have an available appointment, browsing the hospital websites announcements and receive information, viewing their personal profile and their test results.
* For the doctors, the functionalities include; viewing their appointments, creating and sending the laborant the values to be tested and writing a prescription and search/view for patients.
* The laborant receives validation from the doctor to make tests for the patient, and when the tests are done, the laborant sends them back, updated to the doctor and patients profile.

## Scope of the System

The system provides service to doctors, patients and laborants. The system is maintained by an admin. There are some common functionalities for the patient and the doctor, provided they both use the appointment system. They also have functionalities that differ.

The services that the system provides to its users are as follows;

- For the services that the system provides to patients is; Viewing and browsing the hospital website to receive information via hospital news and announcements, check out departments and doctors from a specific department. The patients have to register to take appointments and create a profile, or if they already have an existing account they can login. While logged in, the patient can view and take appointments from the selected doctors provided their schedule is available, browsing doctors from respective departments and viewing the day and time where the doctors have an available appointment. They can cancel their appointments if they like. Also, they can view their personal profile and their test results.

-For the doctors, the functionalities include; After logging in, viewing their appointments, creating and sending the laborant the values to be tested and writing a prescription and search/view for patients.

-The laborant receives validation from the doctor to make tests for the patient, and when the tests are done, the laborant sends them back, updated to the doctor and patients profile.

-The admin maintains the hospital website. He must do this maintenances frequently so the website can work properly. He must also update the changes. When a new doctor joins the hospital staff, or quits the job, admin adds or deletes this doctor. The doctors schedule can also be changed by the admin, given for example their shifts are interchanged. If a new department or hospital building is added, admin can add or manage them. Admin can also add announcements, and make changes to update and fix the bugs of the hospital site. He may receive feedback from the users.

## Objectives and Success Criteria of the Project

The hospital online project will represent the website of a hospital that exists in real life. Other than giving general information from the web about the hospital, it’s main objective will aim to be an appointment management tool. It will have other objectivies to be done, such as the doctors being able to write prescriptions and the patient seeing test results. Provided their test result didn’t show anything major to be examined again, this saves time to the patient, not having to go to the hospital again. Thus, the objectives and success criteria of the project hospital online is the usability to manage appointments with a few clicks, saving time instead of spending more time going to the hospital to take appointment or wait on the phone line. Doctors and laborants should also send each other information (give validation to test result-send test result) faster and more organized, working with usability.

## Definitions, Acronyms, and Abbreviations

RAD: Requirements Analysis Document

Lab : Short for “Laboratory”

Laborant: Laboratory Assistant(to perform tests in the lab)

Via : by way of, by means of, through.

## Overview

Our RAD documentation contains Current System section, Proposed System section, Overview and Introduction of Care & Cure Hospital Online section, Functional Requirements section, Nonfunctional Requirements section, System Models section, Object Model section, Dynamic Model section and finally Glossary.

In Current System section of our RAD documentation, we’ve explained about the system functions and features of the new Care & Cure Hospital Online system to be implemented. We’ve designed it to be user-friendly, destination reachable in a few clicks and have a short response time.

In Proposed System section, we talked about our Care & Cure Hospital systems advantages it’ll provide and what developments we can make on it. Thus taking appointments and doctors to manage their appointments and send/receive test results is much easier by using our system. The system is quite efficient and provide easy payment.

In overview section, we defined overview of features, functions and details of the system.

In Functional Requirements section, we talked about functions and features of Care & Cure Hospital Online. And also, we defined functions on the part of the Patient, Doctor, Admin and the Laborant.

In Nonfunctional Requirements section, we mentioned nonfunctional parts of our new system such as usability, performance, reliability or availability and so on. We described all of nonfunctional requirements of Care & Cure Hospital Online project.

In System Models section, we described scenarios and use cases of Care & Cure Hospital Online. We defined scenarios, actors, and use cases’ flow events and so on in this part.

In Object Model section, we explained class diagrams of our system and we defined relationships between the classes of Care & Cure Hospital.

In Dynamic Model section, we mentioned sequence diagram, that is to say we explained methods and functions and their operations with actors of Care & Cure Hospital Online.

In Glossary section, we described all things of Care & Cure Hospital Online system, which can be seen as the system’s dictionary. We explained all words in RAD documentation.

# Current System

Care & Cure Hospital Online is a website of an hospital with the same name. It is mainly developed for the patients to take online appointments from the hospital’s doctors, and the doctor’s to see the patients appointments and manage their own appointments. The patients can also collect information about the hospital and view their test results. It provides a rather unique, small but efficient part to the laborants which allows them to receive validation from the doctor to update the patients test results and update the results when they’re done to the patients and doctors profile page. In summary, its main purpose is to provide an appointment and information system to the patients and a management system to the doctors and laborants. The users must register in order to provide from the appointment system.

# Proposed System

According to the current online hospital system, online hospital system that will be more efficient and easy for new users.

For example, patient with the new system the hospital of all the departments view all doctors in can easily appointment

patients can register to the site, login, get an appointment, show the their check-up test result,

show the their prescribe. Doctors can register and login to the site, edit free days and times in their schedule.

Doctors can show appointments, prescriptions and check-up test of their patients.

In addition, the interaction between users and online hospital system, it will be better than before.

Interface will be improved with new technologies, so that users would be impressive.

## Overview

Care & Cure Hospital Online is a web-based appointment management system for patients and doctors. The patients use Care & Cure Hospital Online to reach the hospital’s information such as hospital news and announcements, view and browse departments and doctors. In addition, the patients can register if they don’t have an existing account and take appointments from doctors of a specific department. The patients must be logged in in order to take and manage their appointments, and see their test results from the hospital. Doctors use Care & Share Hospital System to…Hospital staff such as Doctors and Laborants do not have to register, since they’re recorded to the database by the admin.

## Functional Requirements

Our system works for getting an appointment and makes a pursuance of the patient and the system displays all the relationships between the patient and the doctor.

- All the users can register to the system with his personal information, name, surname, e-mail.

- All the users can login the system by credentials of e-mail and password.

- All the users can edit profile information.

- Patients can get an appointment.

- Patients and doctors can show appointments and check-up test result.

- Doctors create prescribe and check-up test.

- Doctors show check-up test result of his patients.

- Doctors edit his schedule.

- Doctors can show received appointments.

- Admin can manage doctors and departments all the system.

- Laborant enter check-up test result of the system.

## Nonfunctional Requirements

*Usability* : For a patient should make 7 clicks to get an appointment to our system which

including login, select department and doctor for clicks.

*Reliability :*

-Our system must available at any time on web browsers.

-The system must be running 90% of the time.

-The system must never crash.

*Performance :*

-The system must allow at least 500 users at the same time.

-The system update basket within a few seconds.

-System should response less than 1 second.

-All queries must return a response in less than 3 seconds.

*Supportability :*

-The system must be easily to maintenance.

-The security system must be upgradable.

*Implementation :* The system will be implemented on Eclipse platform. In addition, JAVA will be used as the programming language and html will be used as a framework in the process. User Interface should be web-based (accessible via WWW Browser).

*Interface :*

- System has user-friendly interface to use.

- All actors can use our system so easily.

*Packaging :*

- There is no need for any installations to computer.

- All users should use our system via web browser.

*Legal : www.cchospital.com*

## System Models

### **Scenarios**

**Scenario 1 :**

|  |
| --- |
| *Scenario Name:* AddDepartment |
| *Participant actor instances:* Murat:Admin |
| *Flow of events:*   1. A new department was established to the hospital. For the users to be aware of this new department, Murat must add it to the departments list in the website. 2. Murat opens the website and logs in as admin. He clicks to the departments topic. 3. Murat clicks "add new department" under the departments list. 4. Murat enters the new department's information such as "title", "explanation", "medical devices" and "doctors under this department". 5. Murat clicks "complete" button and there it is. New department added to the system. |

**Scenario 2:**

|  |
| --- |
| *Scenario Name* : *DeleteDepartment* |
| *Participant actor instances:* Murat:Admin |
| *Flow of events:*   1. A department was removed from the hospital. For the users to be aware of this, Murat must delete it from the departments list in the website. 2. Murat opens the website and logs in as admin. He clicks to the departments topic. 3. Murat clicks related department and in this section clicks to "remove this department". 4. The department was removed. |

**Scenario 3:**

|  |
| --- |
| *Scenario Name:* UpdateDoctor |
| *Participant actor instances:* Murat:Admin |
| *Flow of events:*   1. The working shifts of two doctors in the same department has been changed with each other. Murat interchanges their calendar. 2. Murat logs in as admin to the website. From departments list, he finds these two doctors. 3. Murat changes the doctors appointment times in their calendar with each other. When finished, he clicks "update calendar" to update/save the changes. |

**Scenario 4:**

|  |
| --- |
| *Scenario Name:* ListSchedule |
| *Participant actor instances:* Ahmet:Doctor |
| *Flow of events:*   1. In the time period which Ahmet needs to update his informations / empty hours on the system, he determines his hours and free days. 2. He logs in the system and opens his schedule. 3. By the time periods, he chooses his available hours and clicks on the save button. 4. System receives the schedule and saves the differences that Ahmet has done. |

**Scenario 5:**

|  |
| --- |
| *Scenario Name:* ShowTakenAppointment |
| *Participant actor instances:* Ahmet : Doctor |
| *Flow of events:*   1. Ahmet wants to see his appointments to see his next patient. 2. He logs in the system and clicks on the appointments link. 3. The list of his next appointments appears on his screen and he clicks on the patient’s name to see the informations of his patient. 4. He checks the informations and see if the patient was his patient. |

**Scenario 6:**

|  |
| --- |
| *Scenario Name:* ShowTestResult |
| *Participant actor instances:* Ahmet : Doctor Derya:Laborant |
| *Flow of events:*   1. After Ahmet treats his patients he wants them to give some samples to the laboratory and leads his patients to the laboratory. 2. After the patients go to the laboratory and give some samples to the laboratory, Derya takes the samples and examines them. 3. After Derya examines the samples, logs in the system and prepares a document for the results and updates those to the system to make the patient be able to get the results. |

**Scenario 7:**

|  |
| --- |
| *Scenario Name:* SendTestResults |
| *Participant actor instances:* Derya: Laborant Doctor: Ahmet Patient: Arif |
| *Flow of events:*   1. After the patient gave the samples in the lab, Derya has analyzed them and when the test results are out, she logs into the system as laborant. 2. From doctor's logs Derya searches for the patient which gave the samples and clicks on his name. 3. After she opens the patient's page, she logs the patient's test results. 4. After Derya logs the patient's results, she sends them to doctor and patient's pages both. |

**Scenario 8:**

|  |
| --- |
| *Scenario Name:* GetAppointment |
| *Participant actor instances:* Gamze:Patient |
| *Flow of events:*   1. Gamze is sick. She wants look up doctor. She enter for get an appointment www.cchospital.com. 2. She research all the doctors which making an appointment with doctor to decide. 3. She enters her e-mail address and password to login the site. 4. She select KBB department. After all doctors list of KBB department. She is treated by Dr. Ahmet Yılmaz. 5. Gamze select available date and hour list of doctor. 6. Gamze enter name-surname, address, telephone number and e-mail. 7. Thus patient finished appointment. |

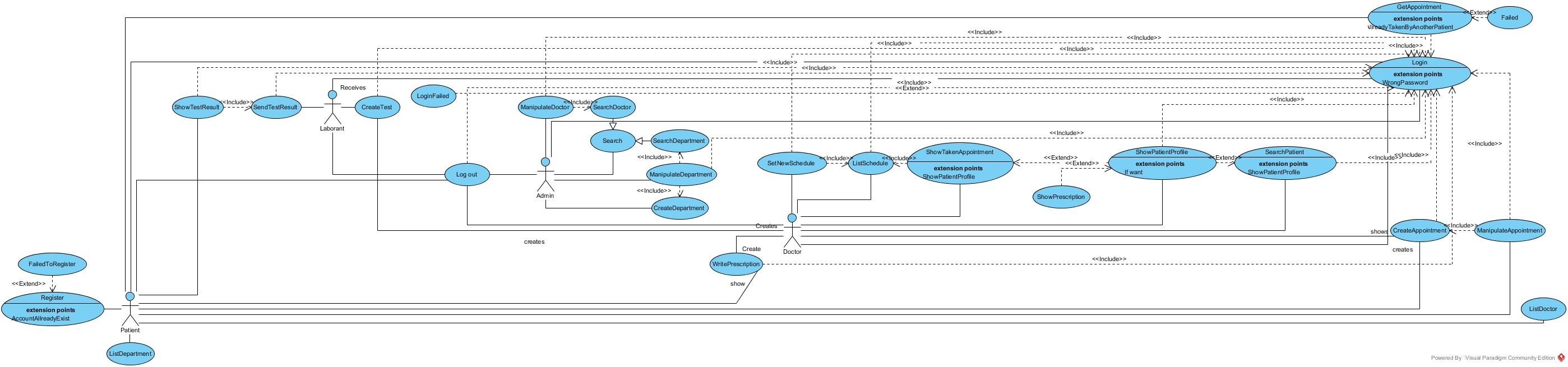
**Scenario 9:**

|  |
| --- |
| *Scenario Name:* ShowPrescription |
| *Participant actor instances:* Ali:Patient |
| *Flow of events:*   1. Ali login www.onlinehospital.com 2. He enter of his Profile. 3. He want show his prescription. 4. He click ' Show Prescription’ tab so show the all prescription. 5. In this way the patient can view all prescriptions. He may have information about the drugs used in the past. |

**Scenario 10:**

|  |
| --- |
| *Scenario Name:* ManipulateAppointment |
| *Participant actor instances:* Ayşe:Patient |
| *Flow of events:*   1. Ayşe login [www.cchospital.com](http://www.cchospital.com). 2. She take an appointment for tomorrow from cardiology but she has a meeting on the company. She will not go to the hospital. 3. She want cancel her appointment. 4. She you can cancel the appointment by manipulate the appointment from that profile. |

### **Use case mode l**



**Use Case 1:**

|  |
| --- |
| *Use case name:* Login |
| *Participant actors:* Admin OR Doctor OR Patient OR Laborant |
| *Flow of events:*   1. Patient enters the web site name on the browser.   2. SYSTEM shows the main page which includes login, sign in, and get appointment.  3. Patient selects the departments and doctors he/she would like to get appointment.  4. SYSTEM requests that the user enter his/her name and password.  5. Patient enters his/her name and password.  6. SYSTEM validates the entered name and password and logs the user into the  System. |
| *Entry Condition:* The Patient enters to login screen. |
| *Exit Condition:* The Patient is logged in OR the Patient has received an explanation indicating why he/she could not login*.* |

**Use Case 2:**

|  |
| --- |
| *Use case name:* Register |
| *Participant actors:* Initiated by Patient |
| 1. Patient enters the web site name on the browser.  2.SYSTEM shows the main page which includes login and sign in button, get appoinment.  3. Patient enters the sign in button.  4.The SYSTEM displays register page.  5. Patient enters his/her information to the register page.  6.The SYSTEM request that the user enter his/her hane, surname, e-mail,id, address and password.  7. Patient enter his/her name, surname, e-mail, address and password.  8.The SYSTEM register the entered name, surname, e-mail, id, address and password. |
| *Entry Condition:* Patient enters sing in button. |
| *Exit Condition:* SYSTEM registers*.* |

**Use Case 3:**

|  |
| --- |
| *Use case name:* Logout |
| *Participant actors:* Admin OR Doctor OR Patient OR Laborant |
| 1. Doctor enters the web site on the browser.  2.SYSTEM shows all authorized to her/his.  3. Doctor taps profile tab.  4.The SYSTEM displays her/his profile.  5. Doctor taps log-out button.  6.The SYSTEM terminates its account access. |
| *Entry Condition:* Doctor must be logged in. |
| *Exit Condition:* Doctor taps another tab OR doctor taps cancel button. |

**Use Case 4:**

|  |
| --- |
| *Use case name:* SearchDepartment |
| *Participant actors:* Initiated by Patient |
| *Flow of events*:   1. Patient login the web site url to the browser.   2. SYSTEM displays the main page.  3. Patient selects the department that she/he wants to search.  4. SYSTEM shows all the departments in the hospital. |
| *Entry Condition:* Patient select department. |
| *Exit Condition*: SYSTEM returns department. |

**Use Case 5:**

|  |
| --- |
| *Use case name:* SearchDoctor |
| *Participant actors:* Initiated by Patient |
| *Flow of events*:   1. Patient login the web site url to the browser.   2. SYSTEM displays the main page.  3. Patient selects the department that she/he wants to search.  4. SYSTEM shows all the departments in the hospital.  5. Patient want to show doctors according to department.  6. SYSTEM shows doctors according to the departments in the hospital. |
| *Entry Condition:* Patient login. |
| *Exit Condition*: SYSTEM returns doctor according to departments. |

**Use Case 6:**

|  |
| --- |
| *Use case name:* SearchPatient |
| *Participant actors:* Initiated by Doctor |
| *Flow of events*:   1. Doctor login the web site url to the browser.   2. SYSTEM displays the main page.  3. Doctor wants to find her/his patient.  4. SYSTEM shows patient to her/his. |
| *Entry Condition:* Doctor login. |
| *Exit Condition*: SYSTEM returns patient. |

**Use Case 7:**

|  |
| --- |
| *Use case name:* ListDoctor |
| *Participant actors:* Initiated by Patient |
| *Flow of events*:   1. Patient enters the web site name to the browser.   2. SYSTEM displays main page.  3. Patient select department that she/he wants to appointment.    4. SYSTEM displays all departments in the hospital.  5. Patient select the doctor according to department.    6. SYSTEM shows the doctor and information page. |
| *Entry Condition:* Patient selects the department. |
| *Exit Condition*: SYSTEM list all doctors. |

**Use Case 8:**

|  |
| --- |
| *Use case name:* ListDepartment |
| *Participant actors:* Initiated by Patient |
| *Flow of events*:   1. Patient enters the web site name to the browser.   2. SYSTEM displays main page.  3. Patient select department that she/he wants to appointment.    4. SYSTEM list all department in the hospital. |
| *Entry Condition:* Patient login in web site. |
| *Exit Condition*: SYSTEM list all departments. |

**Use Case 9:**

|  |
| --- |
| *Use case name:* ListSchedule |
| *Participant actors:* Initiated by Doctor |
| *Flow of events*:   1. Doctor enters the site name to the browser.   2. SYSTEM displays main page.  3. Doctor wants to list your schedule for set new schedule and show taken appointment.  4. SYSTEM list schedule to doctor. |
| *Entry Condition:* Doctor login. |
| *Exit Condition*: SYSTEM list schedule. |

**Use Case 10:**

|  |
| --- |
| *Use case name:* ManipulateAppointment |
| *Participant actors:* Initiated by Patient |
| *Flow of events*:   1. Patient enters the site name to the browser.   2. SYSTEM displays main page.  3. Patient clicks appointments.  4. SYSTEM shows taken appointment to the patient.   1. Patient wants to cancel taken appointment so goes profile page and clicks appointments. |
| *Entry Condition:* Patient must login and have a created appointment |
| *Exit Condition*: SYSTEM cancel appointment. |

**Use Case 11:**

|  |
| --- |
| *Use case name:* ManipulateDepartment |
| *Participant actors:* Initiated by Admin |
| *Flow of events*:   1. Admin enters the site name to the browser.   2. SYSTEM displays main page.  3. Admin clicks departments and then clicks add department.  4. SYSTEM shows the department information screen right after the department list screen.  5. Admin fills the information blanks to create new department then clicks save.  6. SYSTEM creates new department. |
| *Entry Condition:* Admin must login. |
| *Exit Condition*: SYSTEM creates department |

**Use Case 12:**

|  |
| --- |
| *Use case name:* ManipulateDoctor |
| *Participant actors:* Initiated by Admin |
| *Flow of events*:   1. Admin login web site.   2. SYSTEM displays main page.  3. Admin clicks doctor list and then clicks add doctor.  4. SYSTEM shows the doctor information screen right after the doctor list screen.  5. Admin fills the information blanks to add a new doctor then clicks save.  6. SYSTEM adds a new doctor. |
| *Entry Condition:* Admin must login. |
| *Exit Condition*: SYSTEM add doctor. |

**Use Case 13:**

|  |
| --- |
| *Use case name:* ShowPatientProfile |
| *Participant actors:* Initiated by Patient or Doctor |
| *Flow of events*:   1. Doctor login web site.   2. SYSTEM displays main page.  3. Doctor clicks patient list and then clicks related patient to saw related patient’s informations.  4. SYSTEM shows the patient information screen right after the patient list screen. |
| *Entry Condition:* Doctormust login. |
| *Exit Condition*: SYSTEM shows patient profile |

**Use Case 14:**

|  |
| --- |
| *Use case name:* ShowTakenAppointment |
| *Participant actors:* Initiated by Doctor or Patient |
| *Flow of events*:   1. Doctor enters the site name to the browser.   2. SYSTEM displays main page.  3. Doctor clicks show appointments to see taken appointments.  4. SYSTEM shows the appointment list to related doctor. |
| *Entry Condition:* Doctor must login. |
| *Exit Condition*: SYSTEM show taken appointments. |

**Use Case 15:**

|  |
| --- |
| *Use case name:* showTestResults |
| *Participant actors:* Initiated by Patient or Doctor |
| *Flow of events*:   1. Patients enters the web site to the browser.   2. SYSTEM displays main page.  3. Patient clicks “test results”.  4. SYSTEM shows the related test result screen. |
| *Entry Condition:* Patient must login and have a resulted test. |
| *Exit Condition*: SYSTEM show test results. |

**Use Case 16:**

|  |
| --- |
| *Use case name:* showPrescription |
| *Participant actors:* Initiated by Patient or Doctor |
| *Flow of events*:   1. Patients enters the site name to the browser.   2. SYSTEM displays main page.  3. Patient clicks “prescriptions” to see his prescription list.  4. SYSTEM shows the prescription list screen. |
| *Entry Condition:* Patient must login and have a prescription. |
| *Exit Condition*: SYSTEM show prescription. |

**Use Case 17:**

|  |
| --- |
| *Use case name:* LoginFailed |
| *Participant actors:* Initiated by Patient, Doctor, Admin or Laborant |
| *Flow of events*:   1. Laborant enters the site on the browser then fills the login informations.   2. SYSTEM displays login failed screen because of wrong input. |
| *Entry Condition:* Laborant must enters the login screen. |
| *Exit Condition*: SYSTEM shows login failed screen. |

**Use Case 18:**

|  |
| --- |
| *Use case name:* RegisterFailed |
| *Participant actors:* Initiated by Patient |
| *Flow of events*:   1. Patient enters the site on the browser then clicks register then fills the information blanks and clicks complete button.   2. SYSTEM displays register screen then creates a new patient. |
| *Entry Condition:* Laborant must enters the login screen. |
| *Exit Condition*: SYSTEM shows login failed screen. |

**Use Case 19:**

|  |
| --- |
| *Use case name:* getAppointment |
| *Participant actors:* Initiated by Patient |
| *Flow of events*:   1. Patients enters the site name to the browser.   2. SYSTEM displays main page.  3. Patient clicks “get appointment” button to start the operation for getting one.  4. SYSTEM shows the department list screen.  5. Patient selects the related department and then the doctor that wanted to be examined.  6. SYSTEM shows doctor list under the related department then shows suitable calendar dates.  7. Patient selects the date.  8. SYSTEM shows the suitable hours.  9. Patient selects the suitable hour.  10. SYSTEM creates an appointment. |
| *Entry Condition:* Patient must login. |
| *Exit Condition*: SYSTEM creates an appointment. |

**Use Case 20:**

|  |
| --- |
| *Use case name:* SetNewSchedule |
| *Participant actors:* Initiated by Doctor |
| *Flow of events*:   1. Doctor enters the site name to the browser.   2. SYSTEM displays main page.  3. Doctor clicks “set monthly schedule” then fills the schedule in a suitable form and clicks “send” button.  4. SYSTEM shows an empty schedule. Then creates a new monthly schedule. |
| *Entry Condition:* Doctor must login. |
| *Exit Condition*: SYSTEM creates a new monthly schedule. |

**Use Case 21:**

|  |
| --- |
| *Use case name:* WritePrescription |
| *Participant actors:* Initiated by Doctor |
| *Flow of events*:   1. Doctor enters the site name to the browser.   2. SYSTEM displays main page.  3. Doctor searches the related patient in the search bar. Then clicks new prescription and writes one and clicks done.  4. SYSTEM shows the patients informations. Then opens prescription screen and creates it after accepts it. |
| *Entry Condition:* Doctor must login. |
| *Exit Condition*: SYSTEM creates a new prescription. |

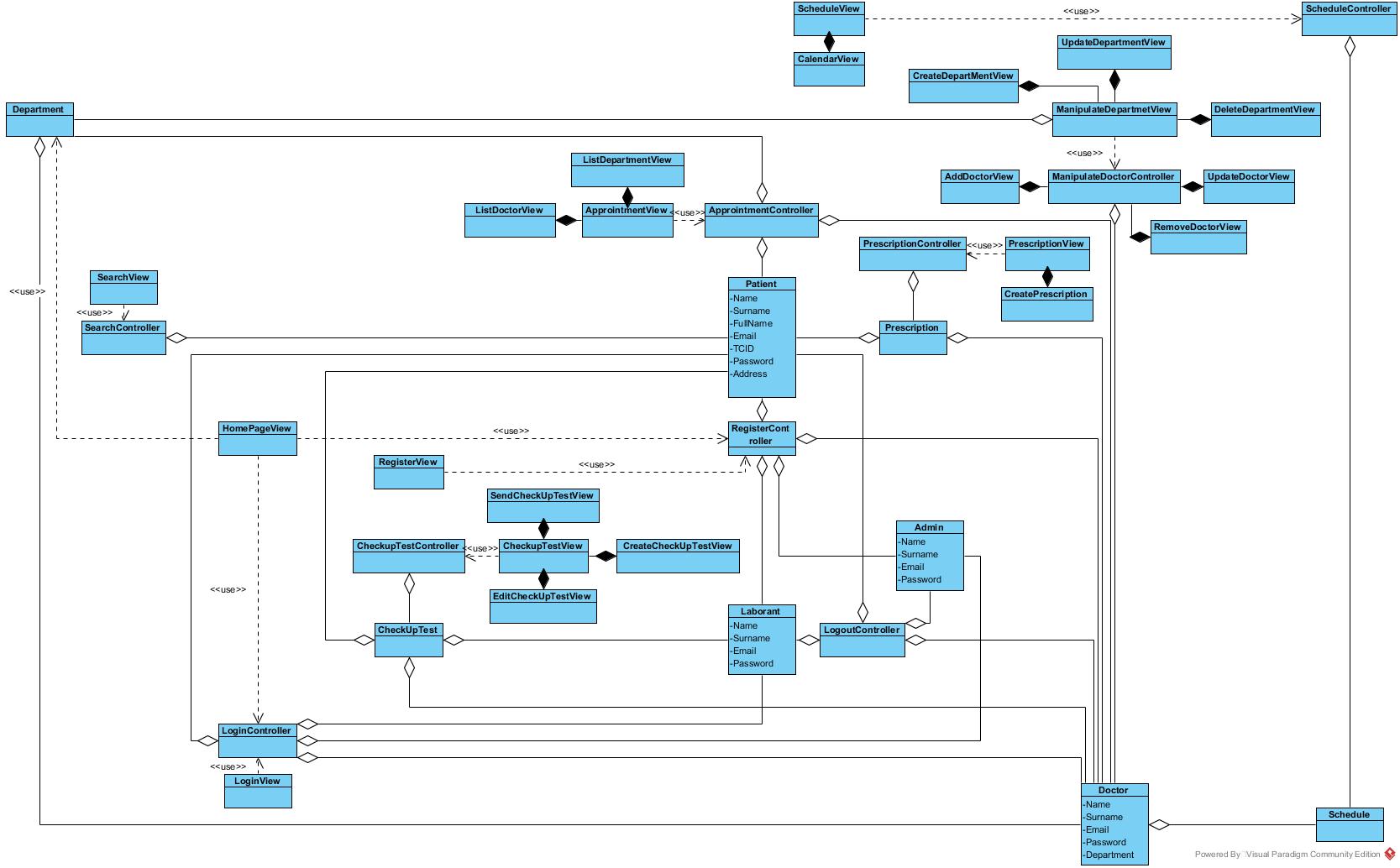
**Use Case 22:**

|  |
| --- |
| *Use case name:* SendTestResult |
| *Participant actors:* Initiated by Laborant |
| *Flow of events*:   1. Laborant enters the site name to the browser.   2. SYSTEM displays main page.  3. Laborant fills the test and receives it back.  4. SYSTEM receives the create test. |
| *Entry Condition:* Laborant must login and have a test that will be resulted. |
| *Exit Condition*: SYSTEM receives test result. |

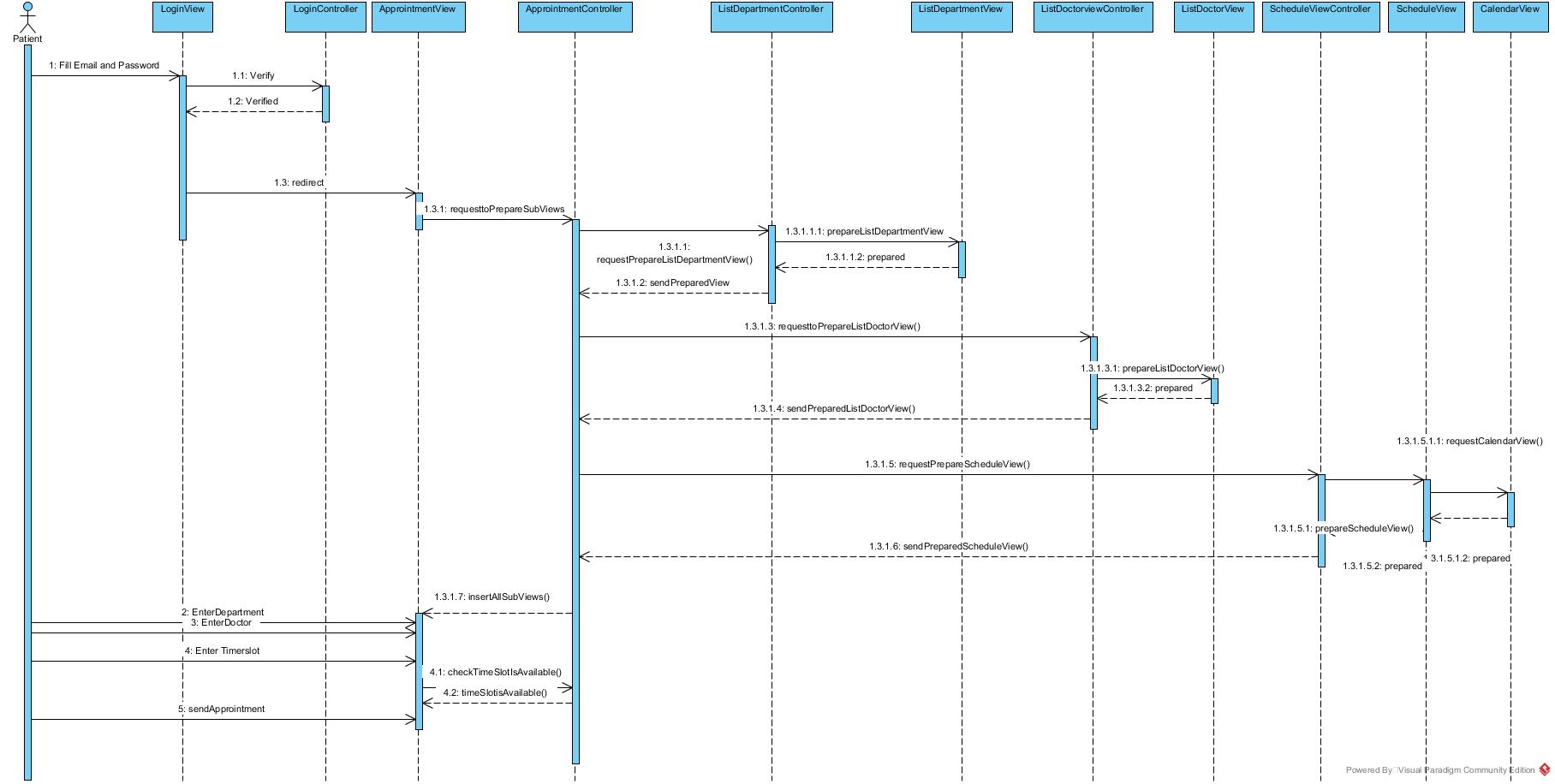
**Use Case 23:**

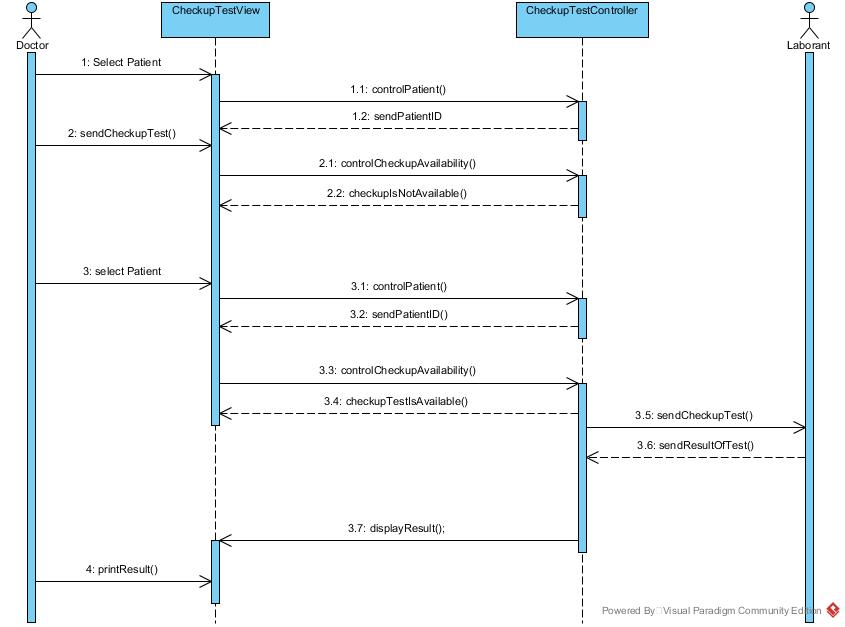
|  |
| --- |
| *Use case name:* CreateTest |
| *Participant actors:* Initiated by Doctor |
| *Flow of events*:   1. Doctor enters the site name to the browser.   2. SYSTEM displays main page.  3. Doctor fills the test and receives it back.  4. SYSTEM receives the create test. |
| *Entry Condition:* Laborant must login and have a test that will be resulted. |
| *Exit Condition*: SYSTEM receives test result. |

### Object model



### Dynamic model

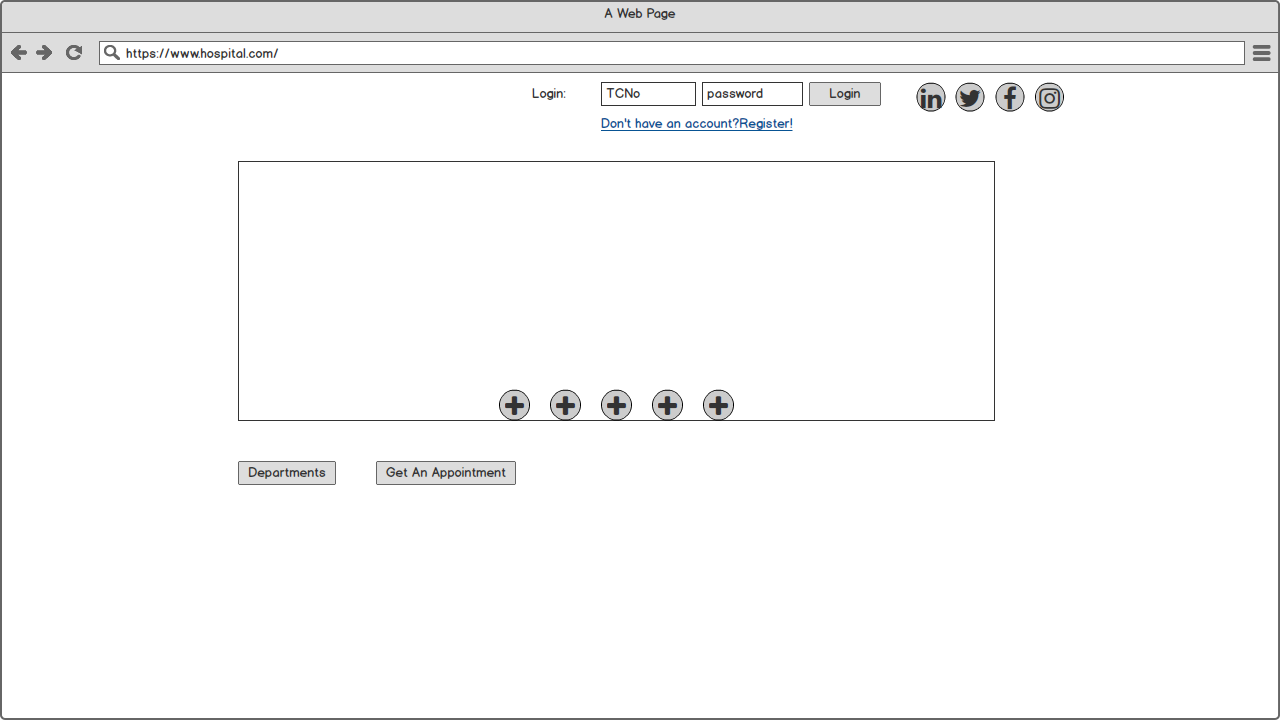
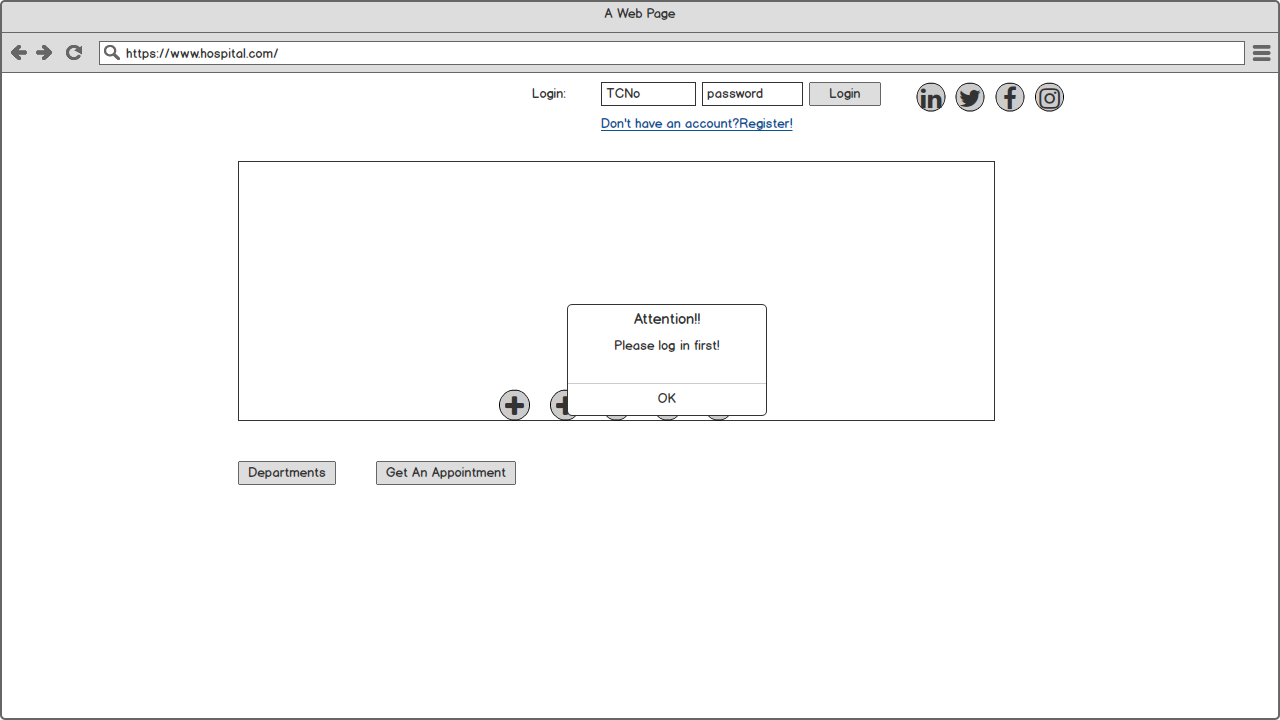
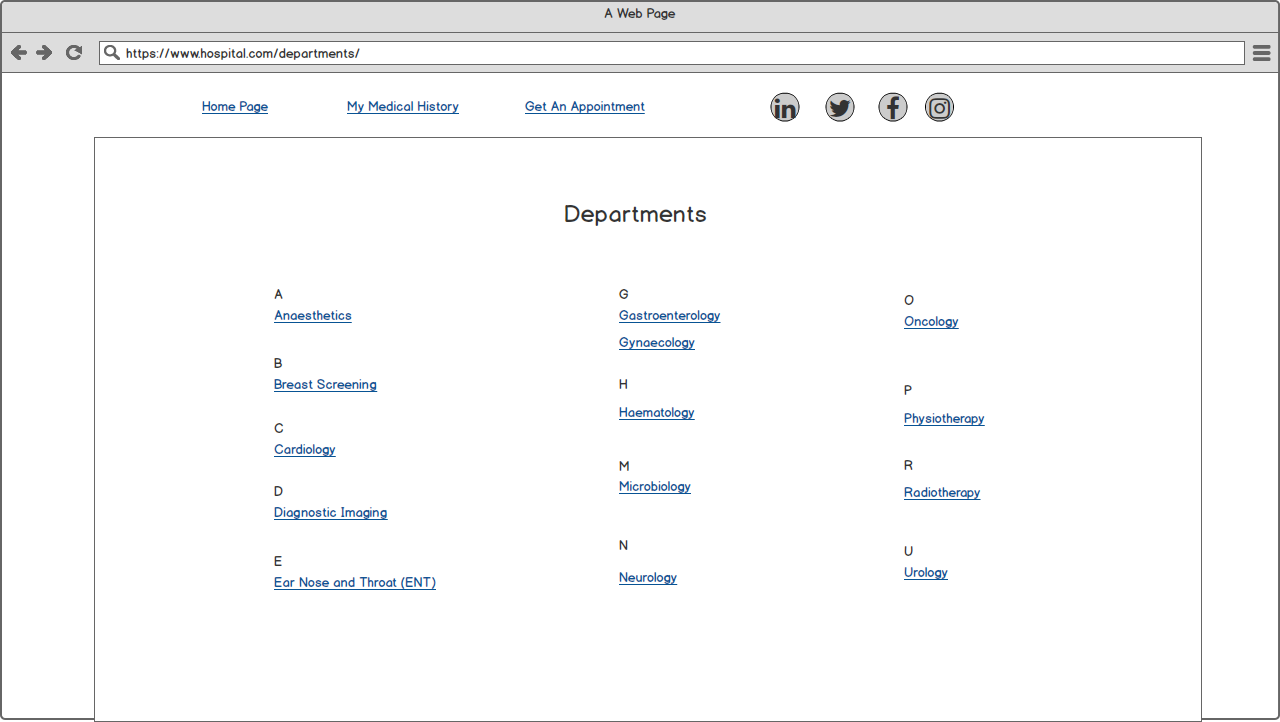
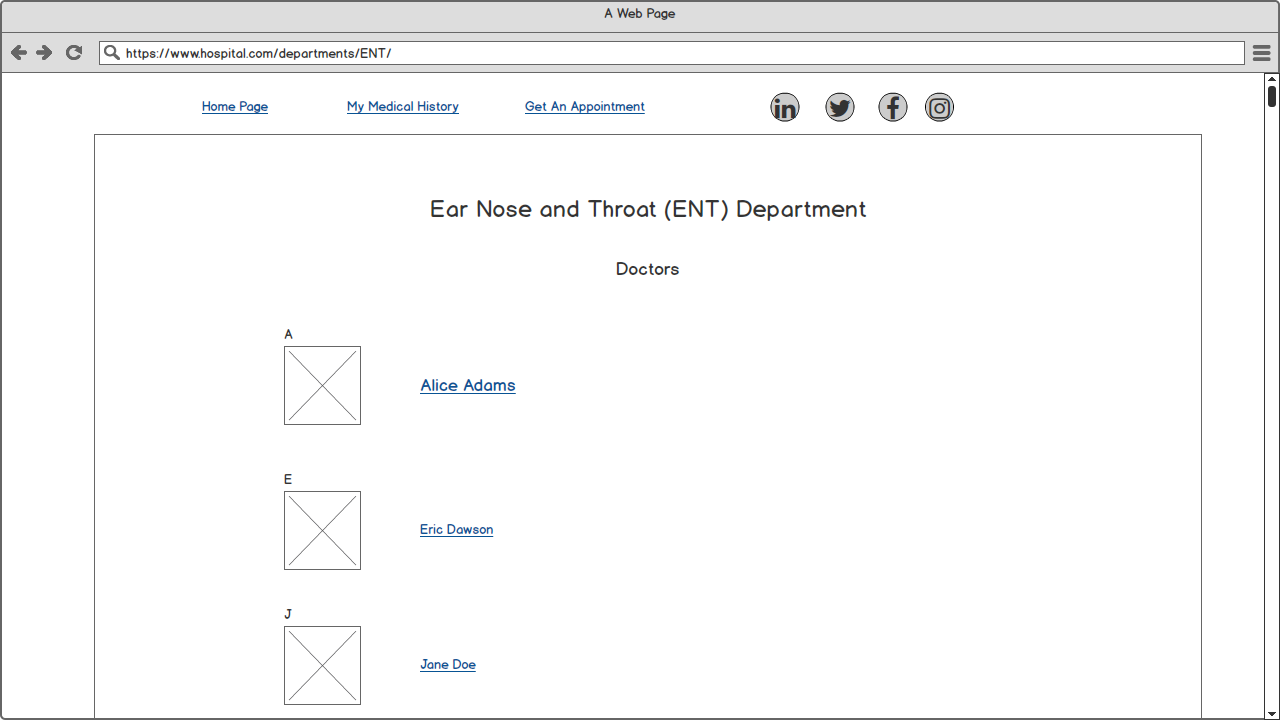
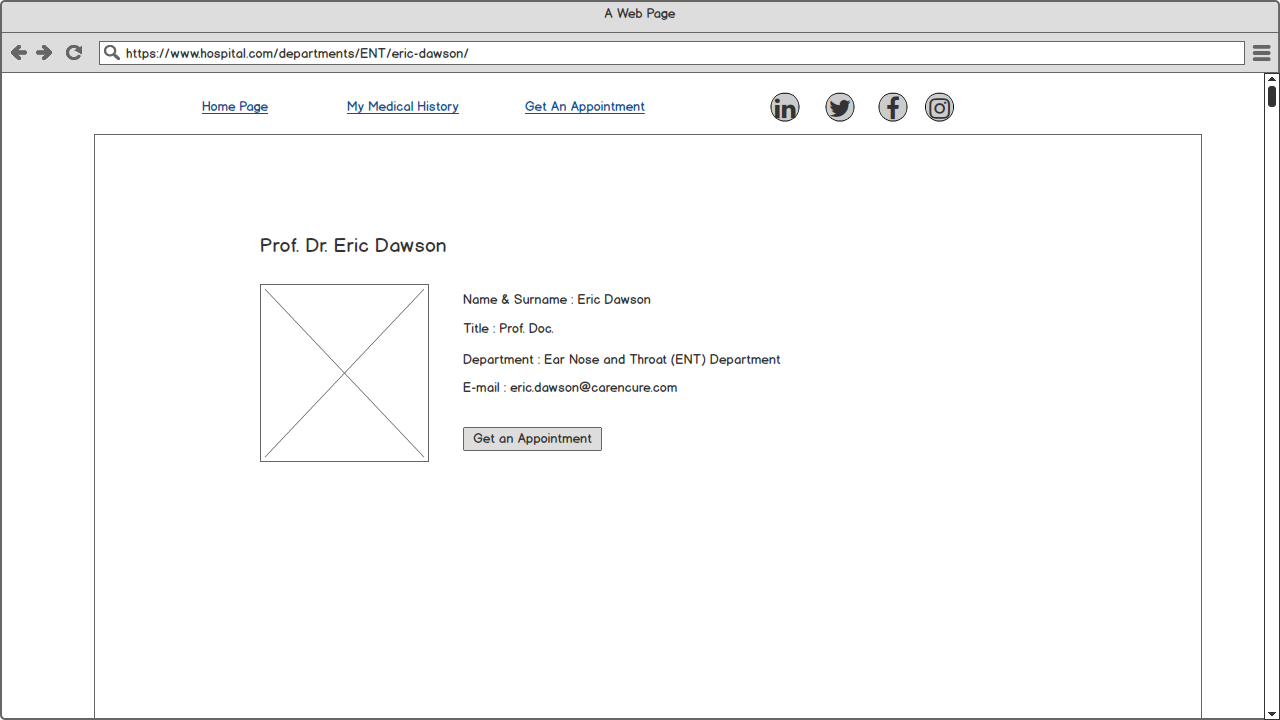
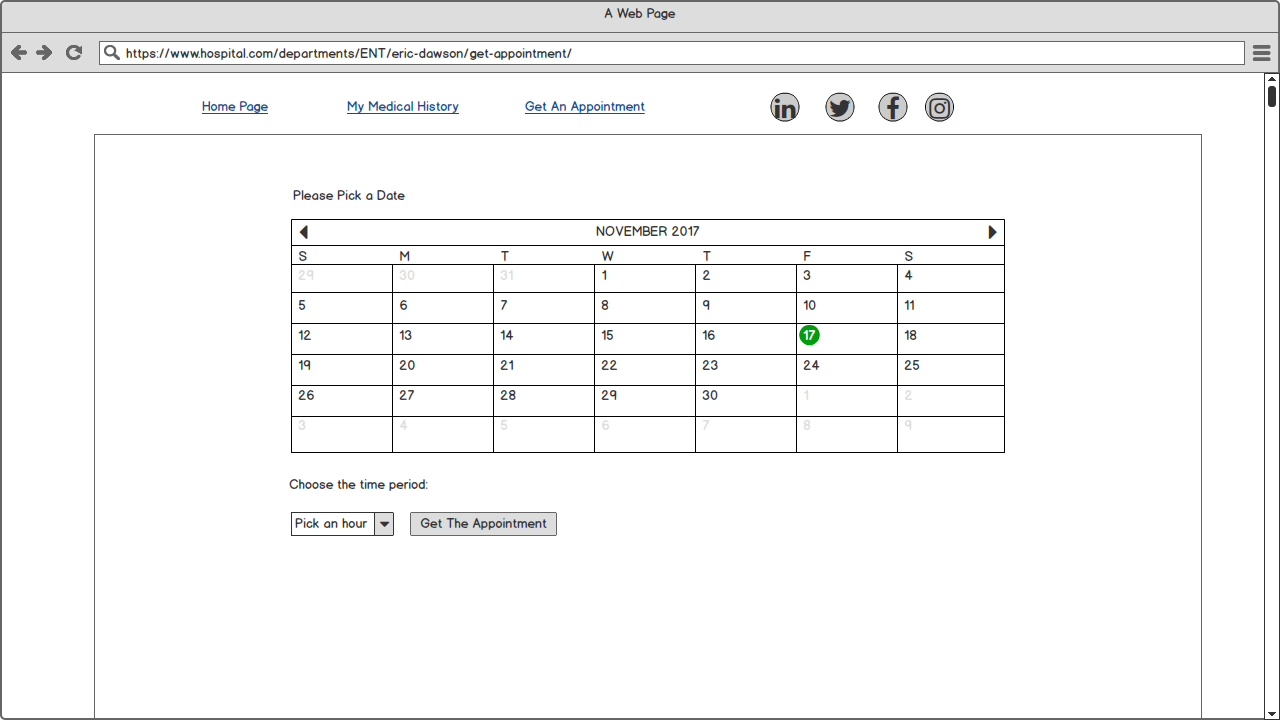




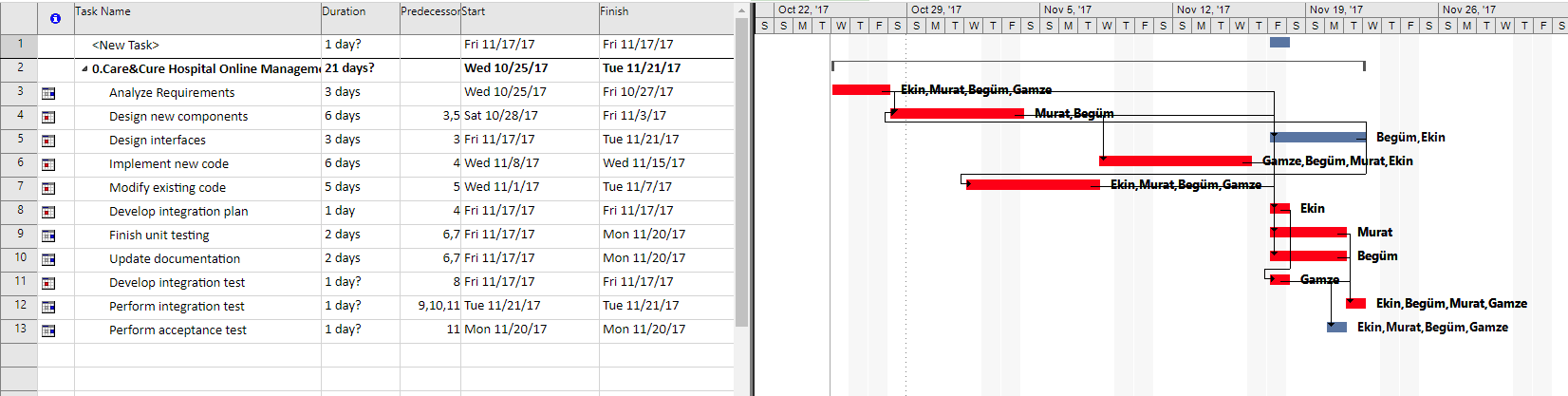
### C:\Users\Begüm\Desktop\301diagramresimleri\PatientPrescription.jpg

### C:\Users\Begüm\Desktop\301diagramresimleri\prescription.jpg

### User interface—navigational paths and screen mock-ups



## Project Schedule



# Glossary

CheckupTest contains patient name, id, date, collestrol, haemogram, fasting blood sugar, liver function, kidney function, lipid profile, uric acid, chest extray, ecg, treadmill test, hepatitis b test.

# References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. https://edisciplinas.usp.br/pluginfile.php/2150022/mod\_resource/content/1/1429431793.203Software%20Engineering%20by%20Somerville.pdf