Life Hospital System

Object Design Document

1.0

27.12.2017

Selin Atalay

Tolga Şıvgın

Murat Akalın

Hasan Cihan Akpınar

Prepared for

SE301 Software Engineering



Table of Contents

[Introduction 1](#_Toc502179773)

[Object Design Trade-Offs 1](#_Toc502179774)

[Interface Documentation Guidelines 2](#_Toc502179775)

[Definitions, Acronyms, and Abbreviations 2](#_Toc502179776)

[References 2](#_Toc502179777)

[Packages 3](#_Toc502179778)

[Models 3](#_Toc502179779)

[Controllers 3](#_Toc502179780)

[Views 4](#_Toc502179781)

[Class Interfaces 4](#_Toc502179782)

OBJECT DESIGN DOCUMENT

# Introduction

We have not been in a trade-off between purchasing and development as we have learned the software development phases. If we were between a tradeoff between purchasing and development, we would not be able to do our job, because it would not give us the source code, and it would be waste of time and money.

## Object Design Trade-Offs

**Reliability**

We tested all the steps of the software, we developed it with different inputs and tried to improve the experience of the users by fixing all the mistakes. We tested the system after every new feature we added. When we encountered any problem, we figured out where we made a mistake.

**Expandability**

While designing our system, we paid attention to extensibility. Thanks to its modular structure, our system is ready for new technologies that can be added later.

**Programmability**

In our project, we tried to create a simple, functional product using the PHP Laravel Framework. The reason we use the Framework is that simple operations such as logging in and signing up are readily available and accordingly we can spend more time on other important parts of the software.

**Maintainability**

After we finished our project, we gave importance to give the necessary technical support. When we sold our restaurant reservation system to a business in our daily life we needed to explain the operation of the system over a period of time, so we tried to provide all the features with an understandable interface.

**Compatibility**

Another reason why we use PHP in our project is that we know that it will not hurt us about compatibility. The Laravel framework is currently the most used framework on PHP andhas a lot of resources. Of course new features can be added to the system later, so we chose to use a software language that is known to survive the problem and make the software long-lived.

**Adaptability**

The software we have developed is designed to adapt to changing conditions. Of course, there are no restrictions from the beginning, since later searches for restaurants and

new features can be added to the database section to make some additions and deletions, as

needed.

**Availability**

Availability is the ratio of time a system or component is functional to the total time it is required or expected to function. The system we developed must be accessible 24/7 and we designed it this way.

## Interface Documentation Guidelines

**Top Navigation**

At the top of the screen we have logo of our site and the buttons that allow users to login and register.

**Icons**

We focused primarily on functionality on our site, but we did not forget to skip some small details. We added the necessary icons to the Profile page section.

**Buttons**

We used bootstrap to build the buttons on the site. We chose to use the ready button designs and measures found here because Laravel automatically supports it. In order to better understand the code we have written, we've put some rules in naming it, you can find these rules in the following table.

## Definitions, Acronyms, and Abbreviations

**Acronyms**

There are no any acronyms.

**Abbreviation**

We used abbreviations in the addresses of the restaurants “e.g. Sk. / Mah. / Cad.”

## References

1. <https://www.zomato.com>
2. <https://www.quandoo.com.tr>
3. <https://www.rezervin.com/tr>

# Packages

# Models

|  |  |
| --- | --- |
| package Online Hospital; | LabResult.cs  Patient.cs  Personnel.cs  Doctor.cs  Appointments.cs  Policlinic.cs  Department.cs  Work\_hours.cs  Admin.cs |

# Controllers

|  |  |
| --- | --- |
| package app\Http\Controller; | AdminListDepartment.aspx.cs  AdminListDoctor.aspx.cs  AdminListPersonnel.aspx.cs  AdminListPolicinic.aspx.cs  PatientBookAppointment.aspx.cs  PatientCancelAnAppointment.aspx.cs  PatientLabResult.aspx.cs  PatientListAppointment.aspx.cs  PersonnelEnterLabResult.aspx.cs  DoctorSetNotWorks.aspx.cs  DoctorListAppointment.aspx.cs  DoctorShowPatientLabResult.aspx.cs  LoginController.aspx.cs  RegisterController.aspx.cs  ForgetPasswordController.aspx.cs |

# Views

|  |  |
| --- | --- |
| package resources\views\admin; | Listdepartment.aspx  ListDoctor.aspx  ListPersonnel.aspx  ListPolicinic.aspx |
| package resources\views\Patient; | BookAppointment.aspx  CancelAnAppointment.aspx  LabResult.aspx  ListAppointment.aspx |
| package resources\views\Personnel; | EnterLabResult.aspx |
| package resources\views\Doctor; | DoctorSetNotWorks.aspx  DoctorListAppoitnemtn.aspx  DoctorShowPatientLabResult.aspx |
| package resources\views\auth; | login.aspx  register.aspx |

# Class Interfaces

**Model Name: User (Patient)**

**Model Attributes/Explanation/Dependencies:**

* **‘name’ : Name of the patient**
* **‘surname : Surname of the patient**
* **‘TC\_no’ : citizenship number of the patient**
* **‘Password : Password of the patient**
* **‘Age’ : Age of the patient**
* **‘Gender’ : Gender of the patient**
* **‘Email’ : Email of the patient**
* **‘Role’ : TC which is given by system database according to type of authenticated user**

**Model Operations:**

* **PatientBookAppointment@index():Lists all available policlinics.**
* **PatientBookAppointment@SelectPoliclinic():Lists all available departments of selected policlinic.**
* **PatientBookAppointment@SelectDepartment():Lists all available doctors of selected department**
* **PatientBookAppointment@SelectDoctor():Lists all available date/dates of selected doctor.**
* **PatientBookAppointment@SelectAvailableDate():If everystep true, confirm the book appointment.**
* **PatientListAppointment@index(): Lists all appointments that belongs to current patient**
* **PatientCancelAppointment@index():Lists all booked appointments**
* **PatientCancelAppointment@SelectAppointment():Select the date then remove the appointment.**
* **PatientLabResult@showResult():Lists all lab results that belongs to current patient**

**Model Name: Policlinic**

**Model Attributes/Explanation/Dependencies:**

* **‘Pname’ : name of the policlinic**
* **‘Paddress’ : address of the policlinic.**
* **‘PoliclinicID’ : ID of the policlinic**
* **‘phone’ : Phone of the policlinic**

**Model Operations:**

**This model does not contain any distinctive operations at any controller.**

**Model Name: Department**

**Model Attributes/Explanation/Dependencies:**

* **‘name’ : Name of the department**
* **‘DepartmentID’ : ID of the department**
* **‘PoliclinicID’: Policlinic which has department**
* **‘BuildingNo’: no of the department**

**Model Operations:**

**This model does not contain any distinctive operations at any controller.**

**Model Name: WorkHours**

**Model Attributes/Explanation/Dependencies:**

* **‘Type’: working hours of the doctors**
* **‘WorkHours’: 9.00-17.00 working hours**

**Model Operations:**

**This model does not contain any distinctive operations at any controller.**

**Model Name: User (Administrator)**

**Model Attributes/Explanation/Dependencies:**

* **‘name’ : Name of the patient**
* **‘surname : Surname of the patient**
* **‘TC\_no’ : citizenship number of the patient**
* **‘Password : Password of the patient**
* **‘Age’ : Age of the patient**
* **‘Gender’ : Gender of the patient**
* **‘Email’ : Email of the patient**
* **‘Role’ : TC which is given by system database according to type of authenticated user**

**Model Operations:**

* **AdminListDepartment@index():Lists all available departments**
* **AdminListDoctor@index():Lists all available doctors**
* **AdminListPersonnel@index():Lists all available personnels**
* **AdminAddDoctor@index():Lists all available departments**
* **AdminAddDoctor@create():adding doctor’s information and confirm**
* **AdminAddDepartment@index():Lists all available policlinics**
* **AdminAddDepartment@create():adding department’s information and confirm**
* **AdminAddPersonnel@index():Lists all available personnels**
* **AdminAddPersonnel@create():adding personnel information and confirm**

**Model Name: LabResult**

**Model Attributes/Explanation/Dependencies:**

* **‘PersonnelTc\_no’ : Tc of the Personnel**
* **‘LabResult : Description about the LabResult**
* **‘DepartmentID’: ID of the Department**

**Model Operations:**

**This model does not contain any distinctive operations at any controller.**

**Model Name: User (Personnel)**

**Model Attributes/Explanation/Dependencies:**

* **‘name’ : Name of the patient**
* **‘surname : Surname of the patient**
* **‘TC\_no’ : citizenship number of the patient**
* **‘Password : Password of the patient**
* **‘Age’ : Age of the patient**
* **‘Gender’ : Gender of the patient**
* **‘Email’ : Email of the patient**
* **‘Role’ : TC which is given by system database according to type of authenticated user**
* **‘PolicinicID’: Personnel work on which policlinic**

**Model Operations:**

* **PersonnelEnterLabResult@index():enter TC number of the patient**
* **PersonnelEnterLabResult@create():adding patient information then confirm**

**Model Name: User (Doctor)**

**Model Attributes/Explanation/Dependencies:**

* **‘name’ : Name of the patient**
* **‘surname : Surname of the patient**
* **‘TC\_no’ : citizenship number of the patient**
* **‘Password : Password of the patient**
* **‘Age’ : Age of the patient**
* **‘Gender’ : Gender of the patient**
* **‘Email’ : Email of the patient**
* **‘Role’ : TC which is given by system database according to type of authenticated user**
* **“DepID’: Doctor works on which department**
* **‘Not\_work’: What days the doctor does not work**
* **‘PolicinicID’: Doctor works on which Policlinic**

**Model Operations:**

* **DoctorSetNotWorks@index():Lists all available works date belongs to current doctor**
* **DoctorSetNotWorks@delete():Select the date/dates for not working then confirm.**
* **DoctorListAppointment@index():Lists all available appointments belongs to current doctor.**
* **DoctorShowPatientLabResult@index():Enter the patient’s TC number**
* **DoctorShowPatientLabResult@show():Lists lab results of the patient**

**Model Name: Appointment**

**Model Attributes/Explanation/Dependencies:**

* **‘AppID’: Every Appointments has individual appointment id**
* **‘PatientID’ : Which patient the appointment belongs to.**
* **‘DepartmentID’ : Patient received the appointment from which department**
* **‘PolicinicID’: Patient received the appointment from which policlinic**
* **‘DoctorID’: Patient received the appointment from who Doctor**
* **‘date’: Patient received the appointment from which Date**
* **‘Type’: The time the patient received the appointment**

**Model Operations:**

**This model does not contain any distinctive operations at any controller.**