**T.C.**

**ISTANBUL AYDIN UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**



**Hospital Guide Web Application**

**ADVISOR: PARVANEH SHAMS**

**BURAK KELEŞ**

**B1405.010018**

**GRADUATION PROJECT**

**COMPUTER ENGINEERING GRADUATION PROJECT**

**2018 – 2019 FALL**

**T.C.**

**ISTANBUL AYDIN UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

**Hospital Guide Web Application**

**ADVISOR: PARVANEH SHAMS**

**BURAK KELEŞ**

**B1405.010018**

**GRADUATION PROJECT**

**COMPUTER ENGINEERING GRADUATION PROJECT**

**2017 – 2018 FALL**

# FOREWORD

In this thesis study, I present a web project for foreign visitors that I have prepared for the use of technology that is included in our daily life in the field of health and with the necessary interface designs. Nowadays, people are available to travel any country. That can be for a vacation or business purposes. However, when people are faced with a health problem in their foreign places, they can struggle to find the right address and have trouble accessing the necessary information. My aim in this thesis is to provide a solution to these problems experienced by people and to create a useful platform. The project was presented as a website and easily accessible on every platform. For the purpose of the project, a design close to the interfaces that different health institutions prepared for them from domestic and foreign sources was used, and the foreigner attractiveness of the users when using the site was minimized. In order to reach the information they want easily, they have access to the information of the hospital in different cities from different countries. Persons can create an appointment in the hospital with the account they have created for themselves and the appointment data will keep in system. In this project, the needs of the users are considered as necessary and accordingly I provide them with access to the areas where they may need access.

To give a general overview of the project; Users create an account using their own passport numbers mail addresses phone numbers and start using the system. Personal passwords are secured by encryption. Users select the country from system that they have traveled. System will shows provinces according to that selection. After select the of province hospitals will be listed to user. Users can access the information of hospital or they can create an appointment if they want to create. In appointment section users select the clinic first and then select a doctor after that they can select a time from the Schedule that prepared for the doctor.

In this thesis, i always have a debt of gratitude to my supervisor PARVANEH SHAMS who helped me about project idea and takes action about domain study. She has been with me in every effort I have spent on the project and motivated me to do a good job.

Best regards.

BURAK KELEŞ

JANUARY 2019

# ÖNSÖZ

Bu tez çalışmamda, günlük yaşamımıza dahil olmuş teknolojinin sağlık alanında kullanımı ve gerekli ara yüz tasarımlarıyla herkesin rahatlıkla kullanabilmesi için hazırladığım ve yabancı ziyaretçiler için geliştirdiğim web projesini sunmaktayım. Günümüzde insanlar rahatlıkla dünyanın birçok yerine seyahat etme imkanı bulabiliyor veya is dolayısıyla farklı ülkelerde bulunabiliyorlar. Ancak insanlar gittikleri yabancı yerlerde bir sağlık problemi ile karşılaştıklarında doğru adresi bulmakta ve gerekli bilgilere erişmekte sıkıntı yaşayabilmektedirler. Bu tezde amacım, insanların yaşadığı bu problemlere bir çözüm sunmak ve kullanışlı bir platform oluşturmaktır. Yapmış olduğum proje bir web sitesi olarak sunulmakta ve her platformda rahatlıkla erişim imkanı sağlanmıştır. Projenin amacına yönelik olması için yerli ve yabancı kaynaklardan farklı sağlık kuruluşlarının kendileri için hazırlamış olduğu ara yüzlere yakın bir tasarım kullanılmış ve kullanıcıların siteyi kullanırken yabancılık çekmesi minimum seviyeye indirilmiştir. Kişilerin istedikleri bilgilere rahat erişebilmeleri için farklı farklı ülkelerden istedikleri şehirlerdeki hastanenin bilgilerine ulaşmaları sağlanmıştır. Kişiler istedikleri takdirde kendileri için oluşturdukları hesap ile hastaneden randevu alabilecekler ve bu randevu bilgilerini kayıt altında tutabileceklerdir. Bu projede kullanıcıların ihtiyaçları gerekli şekilde düşünülmüş buna uygun olarak erişime ihtiyaç duyabilecekleri alanları onların erişimine sunmaktayım.

Proje hakkında genel olarak bilgi vermek gerekirse; Kullanıcılar kendine özel pasaport numaraları mail adresleri telefon numaralarını kullanarak bir hesap oluştur ve sistemi kullanmaya başlar. Kişisel şifreler şifreleme yöntemi ile güvenceye alınmıştır. Kişi sistemdeki ülkelerden birinde ise ülkeyi seçip sistemde kayıtlı illere erişim imkanını elde eder. İl seçiminden sonra il de bulunan hastaneler listelenir. Kullanıcı istediği hastaneyi seçerek bilgilerini görebilir ve dilerse randevu oluşturabilir. Randevu kısmında kişi klinik ve doktor seçerek doktor için oluşturulmuş çizelgeden kendine randevu alabilmektedir.

Tez çalışmamda desteğini esirgemeyen ve beni gerekli şekilde yönlendiren proje fikri ve alan araştırmalarında adımlar atmamı sağlayan danışman hocam PARVANEH SHAMS’ a teşekkürlerimi sunarım. Proje için harcadığım her emekte yanımda olmuş ve ortaya güzel bir iş çıkarabilmem için motive etmiştir.

Saygılarımla.

BURAK KELEŞ

OCAK 2019

# ACKNOWLEDGMENT

I would like to Express my gratitude to everyone who spported me on this Project. It was very complex and i have gotten great motivation and inspiration.

I would like to thank my supervisor, PARVANEH SHAMS for her great support and guidance. She has given me Project idea and guide me to do well prepared user interface. She was very patiently at whole Project development process. Because i was struggling in many fields and she never turn me back.

This Project has shown me how should a team work would be. Because i get lots of help from many source. People share their knowledge with other people and they do not expect any return about that. I am very thankful for the people like doing stuff like that. It was very helpful and i could not complete my work without their guidance.

I would like also to thank my friends at the university for making the best of the worst times during my project study.

Finally, and most importantly, none of this would have been possible without the love and patience of my family, my father and mother. Without their unconditional love, faith and unimaginable patience, this project would never have been possible.

# ABSTRACT

Nowadays it is very rare that people have good knowledge about the country they are traveling to. My goal is to provide an idea to users about hospitals for the health conditions and to enable them to make their own choice. In addition, I have provided other patients comments and ratings for the hospital and provided them a vision to do right choice. Users can make appointments for any doctor in the hospital they want and benefit from the system in accordance with different health needs. Users register once to the system and use the system through the account they have created. Users select the country first. Then the correct province must be selected. The hospitals are listed right after the selection of province. Users can select the hospital they wants to see it is information and create the appointment to the hospital if they wish. Clinic and doctor are selected for the appointment procedure. The appointment process can be complete by choosing an hour from the doctor's schedule.

I designed the scenarios for the project first and made the steps of these scenarios by considering the steps. After logically establishing what interaction the users are launching, I put together what the script will return on the program, and what happens next, as a report in usecase and class diagrams. First a ASP .NET CORE Project created for API and then start to build database design. After finish all the backend processes a Angular Project created for frontend process and made the connection between backend. And finally create a well looking user interface through the html.

**CONTENTS**

[FOREWORD 3](#_Toc535923947)

[ÖNSÖZ 4](#_Toc535923948)

[ACKNOWLEDGMENT 5](#_Toc535923949)

[ABSTRACT 6](#_Toc535923950)

[1. INTRODUCTION 10](#_Toc535923951)

[1.1. Login Module 10](#_Toc535923952)

[1.2. Register Module 10](#_Toc535923953)

[1.3. Register Module 10](#_Toc535923954)

[1.4. Profile Module 11](#_Toc535923955)

[1.5. Appointments Module 11](#_Toc535923956)

[1.6. Homepage Module 11](#_Toc535923957)

[1.7. City Module 11](#_Toc535923958)

[1.8. Hospital Module 11](#_Toc535923959)

[1.9. Hospital Details Module 11](#_Toc535923960)

[1.10. Appointment System Module 12](#_Toc535923961)

[1.11. Search Module 12](#_Toc535923962)

[1.12. Project Motivation 12](#_Toc535923963)

[1.13. Problem statement 12](#_Toc535923964)

[1.14. Project Question 13](#_Toc535923965)

[1.15. Project Objectives 13](#_Toc535923966)

[1.16. Project Significance 13](#_Toc535923967)

[1.17. Project Scope 13](#_Toc535923968)

[2. DIAGRAMS 15](#_Toc535923969)

[2.1. Use Case Diagram 15](#_Toc535923970)

[2.1.1. Use case descriptions 16](#_Toc535923971)

[2.2. Data Flow Diagram 39](#_Toc535923972)

[2.3. Flow Chart 42](#_Toc535923973)

[2.4. Software Architecture Diagram 43](#_Toc535923974)

[2.5. UML Class Diagrams 44](#_Toc535923975)

[2.6. Database Relationship Diagram 45](#_Toc535923976)

[3. REQUIREMENT ANALYSIS 47](#_Toc535923977)

[3.1. Main Purpose of The Project. 47](#_Toc535923978)

[3.2. Requirements Development 47](#_Toc535923979)

[3.3. System Requirements 47](#_Toc535923980)

[3.4. Functional Requirements 48](#_Toc535923981)

[3.5. User Requirements 48](#_Toc535923982)

[3.6. Non Functional Requirements 48](#_Toc535923983)

[3.7. Priority Fields on Project 48](#_Toc535923984)

[4. PROJECT TOOLS 50](#_Toc535923985)

[4.1. MS SQL Server 50](#_Toc535923986)

[4.1.1. Inside sql server's architecture 50](#_Toc535923987)

[4.1.2. Usage of sql server 50](#_Toc535923988)

[4.2. Visual Studio 51](#_Toc535923989)

[4.2.1. .NET 51](#_Toc535923990)

[4.2.2. ASP.NET Core 52](#_Toc535923991)

[4.3. Angular 52](#_Toc535923992)

[4.3.1. Differences between Angular and AngularJS 52](#_Toc535923993)

[4.4. HTML 53](#_Toc535923994)

[4.5. Cascading Style Sheets (CSS) 53](#_Toc535923995)

[5. METHODS 54](#_Toc535923996)

[5.1. MVC Design 54](#_Toc535923997)

[5.1.1. MVC components 55](#_Toc535923998)

[5.2. Entity Framework 56](#_Toc535923999)

[6. USER 58](#_Toc535924000)

[6.1. Home Page 58](#_Toc535924001)

[6.2. Register 59](#_Toc535924002)

[6.3. Login 60](#_Toc535924003)

[6.4. Forget Password 61](#_Toc535924004)

[6.5. Reset Password 62](#_Toc535924005)

[6.6. My Profile 63](#_Toc535924006)

[6.7. Cities 64](#_Toc535924007)

[6.8. Hospitals 65](#_Toc535924008)

[6.9. Hospital Details 66](#_Toc535924009)

[6.10. Select Appointment 67](#_Toc535924010)

[6.11. Appointment Confirmation 68](#_Toc535924011)

[6.12. My Appointment 69](#_Toc535924012)

[6.13. Home Page Countries Search 70](#_Toc535924013)

[6.14. Cities Search 71](#_Toc535924014)

[6.15. Hospitals Serach 72](#_Toc535924015)

[7. ADMİN 73](#_Toc535924016)

[7.1. Login 73](#_Toc535924017)

[7.2. Admin Dashboard 74](#_Toc535924018)

[7.3. Add Country 75](#_Toc535924019)

[7.4. Counrty List 76](#_Toc535924020)

[7.5. Update Country 77](#_Toc535924021)

[7.6. Add City 78](#_Toc535924022)

[7.7. City List 79](#_Toc535924023)

[7.8. Update City 80](#_Toc535924024)

[7.9. Add Hospital 81](#_Toc535924025)

[7.10. Hospitals List 82](#_Toc535924026)

[7.11. Update Hospital 83](#_Toc535924027)

[7.12. Add Clinic 84](#_Toc535924028)

[7.13. Clinics List 85](#_Toc535924029)

[7.14. Update Clinic 86](#_Toc535924030)

[7.15. Add Doctor 87](#_Toc535924031)

[7.16. Doctors List 88](#_Toc535924032)

[7.17. Update Doctor 89](#_Toc535924033)

[7.18. Add Day 90](#_Toc535924034)

[7.19. Days List 91](#_Toc535924035)

[7.20. Update Day 92](#_Toc535924036)

[7.21. Add Hour 93](#_Toc535924037)

[7.22. Hours List 94](#_Toc535924038)

[7.23. Update Hour 95](#_Toc535924039)

[7.24. Users List 96](#_Toc535924040)

[7.25. Comments List 97](#_Toc535924041)

[8. TESTING 99](#_Toc535924042)

[8.1. About Unit Testing in General 99](#_Toc535924043)

[9. REFERENCES 100](#_Toc535924044)

[Figure 1 - Use Case Diagram 18](#_Toc535924169)

[Figure 2 - Data Flow Diagram for Non Member User 42](#_Toc535924170)

[Figure 3 - Data Flow Diagram for Member User 43](#_Toc535924171)

[Figure 4 - Data Flow Diagram for System Admin 44](#_Toc535924172)

[Figure 5 - Flow Chart Diagram 45](#_Toc535924173)

[Figure 6 - Architecture Diagram 46](#_Toc535924174)

[Figure 7 - Class Diagram 47](#_Toc535924175)

[Figure 8 - Database Relationship Diagram 48](#_Toc535924176)

[Figure 9 - MVC Components 58](#_Toc535924177)

[Figure 10 - Home Page 61](#_Toc535924178)

[Figure 11 - Register Page 62](#_Toc535924179)

[Figure 12 - Login Page 63](#_Toc535924180)

[Figure 13 - Forgot Password 64](#_Toc535924181)

[Figure 14 - Reset Password 65](#_Toc535924182)

[Figure 15 - My Profile Page 66](#_Toc535924183)

[Figure 16 - Cities Page 67](#_Toc535924184)

[Figure 17 - Hospitals Page 68](#_Toc535924185)

[Figure 18 - Hospital Detail Page 69](#_Toc535924186)

[Figure 19 - Create Appointment Page 70](#_Toc535924187)

[Figure 20 - Confirm Appointment 71](#_Toc535924188)

[Figure 21 - My Appointments Page 72](#_Toc535924189)

[Figure 22 - Contry Search Field 73](#_Toc535924190)

[Figure 23 - Search City Field 74](#_Toc535924191)

[Figure 24 - Search Hospital Field 75](#_Toc535924192)

[Figure 25 - Admin Login 76](#_Toc535924193)

[Figure 26 - Admin Home Page 77](#_Toc535924194)

[Figure 27 - Add Country Page 78](#_Toc535924195)

[Figure 28 - Country Page List 79](#_Toc535924196)

[Figure 29 - Update Country Page 80](#_Toc535924197)

[Figure 30 - Add City Page 81](#_Toc535924198)

[Figure 31 - City List Page 82](#_Toc535924199)

[Figure 32 - Update City 83](#_Toc535924200)

[Figure 33 - Add Hospital Page 84](#_Toc535924201)

[Figure 34 - Hospital List Page 85](#_Toc535924202)

[Figure 35 - Update Hospital Page 86](#_Toc535924203)

[Figure 36 - Add Clinic Page 87](#_Toc535924204)

[Figure 37 - Clinic List Page 88](#_Toc535924205)

[Figure 38 - Update Clinic Page 89](#_Toc535924206)

[Figure 39 - Add Doctor Page 90](#_Toc535924207)

[Figure 40 - Doctor List Page 91](#_Toc535924208)

[Figure 41 - Update Doctor Page 92](#_Toc535924209)

[Figure 42 - Add Day Page 93](#_Toc535924210)

[Figure 43 - Day List Page 94](#_Toc535924211)

[Figure 44 - Update Day Page 95](#_Toc535924212)

[Figure 45 - Add Hour Page 96](#_Toc535924213)

[Figure 46 - Hour List Page 97](#_Toc535924214)

[Figure 47 - Update Hour Page 98](#_Toc535924215)

[Figure 48 - Users List Page 99](#_Toc535924216)

[Figure 49 - Comments List Page 100](#_Toc535924217)

**CHAPTER 1:**

**INTRODUCTION**

# INTRODUCTION

The travelling interest of humans is getting bigger. Nowadays lots if people travel all araound the world. But if you are new in a country you cant know every information about city of the country. My aim is support that people to eleminate that kind of problems. My web application shows all the hospitals that is added. So users can know their choice options. After select a hospital the related information of that hospital are shown by the system. And user able to see user comments and hospital rate that done by users. If the user whishes can create a appointment by select a clinic and doctor on that clinic after that work schedule if the doctor listed and user can select a date from the schedule and save the application details.

## Login Module

In this section, the user logs in by entering the mail address and password. If the user enters the mail address and password incorrectly, appication gives information about that. If user enter to system correctly users’ user id holds in and every information about user reqests by that user id.

## Register Module

In this section, user create a membership to application. If users’ mail address already in system application shows information message about that.

## Register Module

In this section, user want to see what was the password. Application takes users’ mail address information from user and if user enter that information correct application send a e-mail to users mail address that includes user password.

## Profile Module

In this section, members see profile information and can edit information if they want to edit any information saved in system.

## Appointments Module

In this section, members can see their saved appointments for a doctor. All the information about the related appointment are listed in that section.

## Homepage Module

In this section, members see default page logo and above that counties in system listed one by one. Users can select desired country in this section.

## City Module

In this section, cities of selected country will be display. Users can select city wherever they traveled. Cities are grouped by their initial letter.

## Hospital Module

In this section, hospitals of selected city will be display. Users can select a hospital that wants to Access information.

## Hospital Details Module

In this section, hospital details will be display. Every hospital has their own detail page. And users can see Hospital name, location, phone, adress information of related hospital. Also users comments and hospital rate will be show in this section. And users can rate the hospital and make a comment to hospital.

## Appointment System Module

In this section users select a clinic from list and system will loads doctors of that clinic. After users select the doctor system will loads doctor’s work schedule. After selection of the day users will be able to select time for the related day. And the appointment can save by user click to save area.

## Search Module

There are different types search areas. In country section users can search a country from the list. In city section users can search a city just type in some letters. In hospital section users are able to search hospital by the name. For each search area different query methods is processed.

## Project Motivation

This Project is prepared to reduce foreign visitors struggling about hospital choosing. If they travel fort he first time to a city it is probably hard to do right choice. So people need a assist for their fresh experiences.

## Problem statement

Technology is rapidly growing in every field. Today most of the countries use their hopital appointment system like Turkey’s MHRS system. Citizens use their system effectively. But foreign visitors are completely strange to a new city and the national appointment systems may not meet their needs. So foreign visitors needs a hospital system that they can find information about the hospital and can create a appointment to hospital like they are a citizen of the country.

## Project Question

The focus of that Project is building a hospital web system for assist foreign visitors and questions are:

* What are the user requirements for that system.
* How to elemitane existing problems.
* How to create user friendly interface.

## Project Objectives

The main objective of this project is produce desired project to meet all the peoples' need who travel across World.

Benefits of the Project and solutions:

* Develop a usable system that covers all the countries along the World.
* Helping the people who are interested with such projects.
* Applying and merging new knowledges and merging ideas to create something new.

## Project Significance

This kind of system is never done before. It will cover all the countries in the World and users be able to access every hospital system. This will help every travelers to book appointment all around the World.

## Project Scope

The Scope of the project can be explained for large segment of people but mostly will use by people who have traveled to another country for a sort time. Heath is an important issue and people should able to know how to go health center or how to book an appointment. Todays technology is making people live easier so it is possible to produce a complitely usable system in everywhere.

**CHAPTER 2:**

**DESIGN PHASE**

# DIAGRAMS

## Use Case Diagram

I designed this use case diagram for Non member user, Member user and System Admin. Use case diagram presents the actions that actors can do.

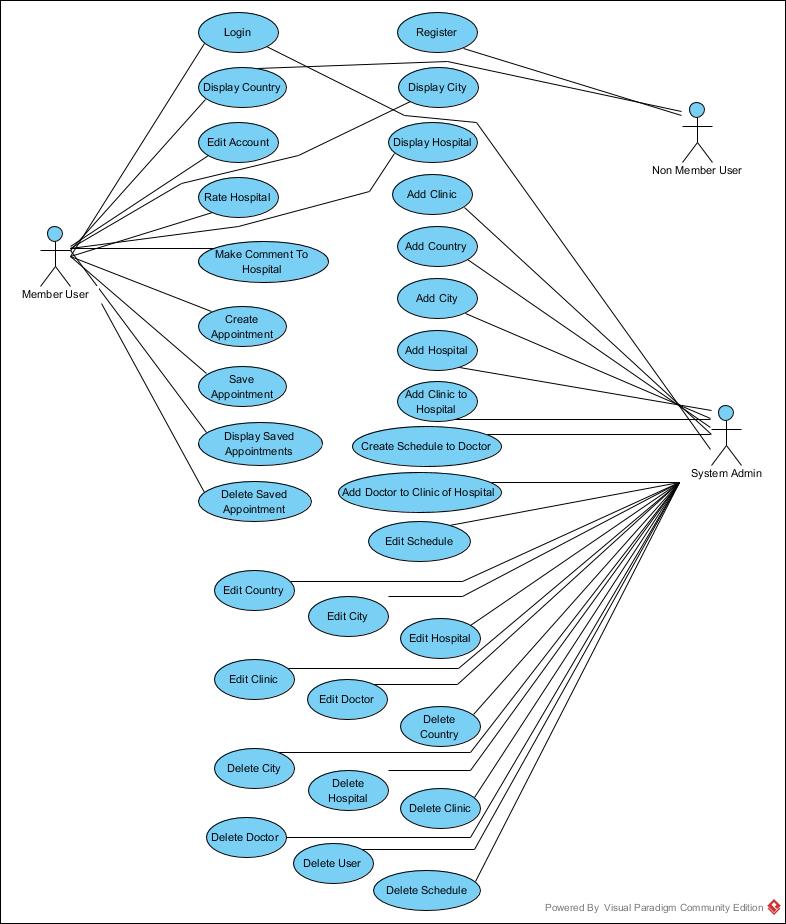


Figure 1 - Use Case Diagram

### Use case descriptions

**Use case ID:** 1

**Use case name:** Register

**Summary:** Users will create a membership so they can use all features of application.

**Actor:** Non-member users

**Precondition:** Application should work correctly. User should not be logged in.

**Main sequence:**

1. Users should click register.

2. Web site should load register page.

3. Users will enter register data to fields and click to register button.

4. Application checks user passaport id and email in database.

5. Application save register data to database.

6. Application redirect member to homepage.

**Alternative sequence:**

Step 3.1: User can cancel the process.

Step 3.2: Application load home page.

Step 4.1: If the passaport id or email is available for another user, the system displays an error message and prompts for another username or email.

**Postcondition:** The actor registers in system. Be able to loginnto system.

**Use case ID:** 2

**Use case name:** Login

**Summary:** System validates the user.

**Actor:** Member user

**Dependency:** Register.

**Precondition:** Application should work correctly, User should registered to system.

**Main sequence:**

1. Users click to Login

2. Application loads login page.

3. User enter mail address and password to fields and click to login.

4. Application checks username and password.

5. Application loads homepage if informations are correct.

**Alternative Sequence:**

Step 3.1: User can cancel the process.

Step 3.2: Application load home page.

Step 4.1: If the username or password is not correct, the system displays an error message and prompts for the correct mail address and password.

**Postcondition:** Member see account page and it can continue to use applications feature.

**Use case ID:** 3

**Use case name:** Edit account.

**Summary:** Member update account information.

**Actor:** Member user

**Dependency:** Login.

**Precondition:** Application should work correctly, User should logged in.

**Main sequence:**

1. Member click to my profile.

2. Application loads profilepage and fill the fileds with users information.

3. Member enter new data and click to edit.

4. Application update new data to database.

5. Redirect member to account page.

**Alternative sequence:**

Step 3.1: Member can cancel the operation.

Step 3.2: Application reload account page without any changes.

Step 4.1: If any error occurs application will display error message to user.

**Postcondition:** Member sees account informations in my profile page and can continue to use applications feature.

**Use case ID:** 4

**Use case name:** Display countries.

**Summary:** Member see country that added to system by system admin

**Actor:** Member user, Non member user.

**Precondition:** Application should work correctly and should load countries to related field.

**Main sequence:**

1. User enters to web site.

2. Application loads country information to field.

3. User goes to bottom of website and become able to see countries.

**Alternative sequence:**

Step 2.1: Country field information may not be loaded.

Step 2.2: Country table of database can be empty.

**Postcondition:** User become able to select country if they logged in.

**Use case ID:** 5

**Use case name:** Display cities.

**Summary:** Member see cities that added to system by system admin

**Actor:** Member user.

**Dependency:** Login.

**Precondition:** Application should work correctly and should load cities to related fields.

**Main sequence:**

1. User should select the country.

2. Application loads city page and load city information to fields.

3. Users able to use city categories on city page.

**Alternative sequence:**

Step 2.1: City field information may not be loaded.

Step 2.2: City table of database can be empty.

**Postcondition:** User become able to select city.

**Use case ID:** 6

**Use case name:** Display hospitals.

**Summary:** Member see hospitals that added to system by system admin

**Actor:** Member user.

**Dependency:** Login.

**Precondition:** Application should work correctly and should load hospitals to related fields.

**Main sequence:**

1. User should select the city.

2. Application loads hospital information to field.

3. Users become able to see hospital details on hospital page.

**Alternative sequence:**

Step 2.1: Hospital field information may not be loaded.

Step 2.2: Hospital table of database can be empty.

**Postcondition:** User become able to select hospital.

**Use case ID:** 7

**Use case name:** Display hospital details.

**Summary:** Member see hospital details.

**Actor:** Member user.

**Dependency:** Login.

**Precondition:** Application should work correctly and should load hospital information.

**Main sequence:**

1. User should select the hospital.

2. Application loads hospital information to field.

3. Users become able to see hospital information on hospital detail page.

4. Users become able to see hospital’s comments and hospital rate.

**Alternative sequence:**

Step 2.1: Hospital details may not be loaded.

Step 4.1: Hospital comment table on database may be empty.

Step 4.2: Hospital rate table on database may be empty.

**Postcondition:** User become able to select hospital details.

**Use case ID:** 8

**Use case name:** Make comment to hospital.

**Summary:** Member post a comment to displayed hospital.

**Actor:** Member user.

**Dependency:** Login.

**Precondition:** Application should work correctly and should load hospital detail page.

**Main sequence:**

1. User should select the hospital.

2. Application loads hospital information to field.

3. Users become able to see hospital information on hospital detail page.

4. Users become able to see hospital’s comments and hospital rate.

5. User goes to bottom of website and type a comment.

6. User should submit the comment to system.

**Alternative sequence:**

Step 2.1: Hospital details may not be loaded.

Step 4.1: Hospital comment table on database may be empty.

Step 4.2: Hospital rate table on database may be empty.

Step 5.1: User can make a rating to hospital if want to do.

Step 6.1: If any problem occurs application display error page.

**Postcondition:** User become able to see added information.

**Use case ID:** 9

**Use case name:** Rate Hospital

**Summary:** Member make rating to displayed hospital.

**Actor:** Member user.

**Dependency:** Login, Make comment to hospital.

**Precondition:** Application should work correctly and should load hospital detail page.

**Main sequence:**

1. User should select the hospital.

2. Application loads hospital information to field.

3. Users become able to see hospital information on hospital detail page.

4. Users become able to see hospital’s comments and hospital rate.

5. User goes to bottom of website and type a comment and select rating value.

6. User should submit the comment and rate to system.

**Alternative sequence:**

Step 2.1: Hospital details may not be loaded.

Step 4.1: Hospital comment table on database may be empty.

Step 4.2: Hospital rate table on database may be empty.

Step 6.1: If any problem occurs application display error page.

**Postcondition:** User become able to see added information.

**Use case ID:** 10

**Use case name:** Create appointment.

**Summary:** Member book an appointment to displayed hospital.

**Actor:** Member user.

**Dependency:** Login.

**Precondition:** Application should work correctly and appointment related information should load correctly to appointment page.

**Main sequence:**

1. User should click to book appointment button.

2. Application loads appointment page.

3. Users become able to select clinic.

4. Users become able to select doctor.

5. Application loads doctor’s available schedule to related field.

6. User select desired appointment day.

7. Application loads dostor’s work time according to selected day.

8. User select desired appointment time.

9. User click to confirm in order to submit the appointment process.

10. Application loads appointments details to pop-up screen and gives knowledge to user.

11. User select confirm in order to complete submitting process.

**Alternative sequence:**

Step 3.1: Clinic fields of related hospital on database may be empty.

Step 4.1: Doctor fields of related hospital on database may be empty.

Step 6.1: Appointment days field of related doctor on database may be empty.

Step 8.1: Appointment hours field of related doctor on database may be empty.

Step 9.1: Member can cancel the process.

Step 11.1: Member can cancel the process.

**Postcondition:** Created appointment going to save users My appointments fields and user become able to see saved appointment details.

**Use case ID:** 11

**Use case name:** Display saved appointments.

**Summary:** Members see their saved appointment details.

**Actor:** Member user.

**Dependency:** Login, Create appointment.

**Precondition:** Application should work correctly and member should create an appointment previously.

**Main sequence:**

1. User click to my profile.

2. Application loads profile page and load My appointments tab.

3. Appication loads appointment details to fields.

**Alternative sequence:**

Step 2.1: profile page may not be loaded.

Step 3.1: User may not create any appointment.

**Postcondition:** Users become able to see appointment details.

**Use case ID:** 12

**Use case name:** Delete saved appointments.

**Summary:** Members delete their saved appointment.

**Actor:** Member user.

**Dependency:** Login, Create appointment.

**Precondition:** Application should work correctly and member should create an appointment previously.

**Main sequence:**

1. User click to my profile.

2. Application loads profile page and load My appointments tab.

3. Appication loads appointment details to fields.

4. Member select delete process in order to delete related appointment.

**Alternative sequence:**

Step 2.1: profile page may not be loaded.

Step 3.1: User may not create any appointment.

Step 4.1: Delete process may crush and application display error page.

**Postcondition:** Users become able to see appointment details.

**Use case ID:** 13

**Use case name:** Login

**Summary:** System validates the admin.

**Actor:** System Admin.

**Precondition:** Application should work correctly.

**Main sequence:**

1. Users enter admin page url to web browser link tab in order to Login

2. Application loads admin login page.

3. User enter mail address and password to fields and click to login.

4. Application checks username and password.

5. Application loads admin panel dashboard if informations are correct.

**Alternative Sequence:**

Step 3.1: User can cancel the process.

Step 4.1: If the username or password is not correct, the system displays an error message and prompts for the correct mail address and password.

**Postcondition:** Admin see admin panel dashboard and it can continue to use applications feature.

**Use case ID:** 14

**Use case name:** Add country

**Summary:** Admin add a new country to system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly.

**Main sequence:**

1. Admin select add country process from dashboard.

2. Application loads add country page.

3. Admin fills the required fields.

4. Admin click to save.

5. Application saves new country to database.

**Alternative Sequence:**

Step 4.1: Admin can cancel the process.

Step 5.1: if any error occurs application display error page.

**Postcondition:** Country added successfully and users are able to reach that country also admin able to see that country.

**Use case ID:** 15

**Use case name:** Add city

**Summary:** Admin add a new city to created country in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Country should added.

**Main sequence:**

1. Admin select add city process from dashboard.

2. Application loads add city page.

3. Admin fills the required fields.

4. Admin click to save.

5. Application saves new city to database.

**Alternative Sequence:**

Step 4.1: Admin can cancel the process.

Step 5.1: if any error occurs application display error page.

**Postcondition:** City added successfully and users are able to reach that city also admin able to see that city.

**Use case ID:** 16

**Use case name:** Add hospital

**Summary:** Admin add a new hospital to created city in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. City should added.

**Main sequence:**

1. Admin select add hospital process from dashboard.

2. Application loads add hospital page.

3. Admin fills the required fields.

4. Admin click to save.

5. Application saves new hospital to database.

**Alternative Sequence:**

Step 4.1: Admin can cancel the process.

Step 5.1: if any error occurs application display error page.

**Postcondition:** Hospital added successfully and users are able to reach that hospital also admin able to see that hospital.

**Use case ID:** 17

**Use case name:** Add clinic.

**Summary:** Admin add a new clinic to system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly.

**Main sequence:**

1. Admin select add clinic process from dashboard.

2. Application loads add clinic page.

3. Admin fills the required fields.

4. Admin click to save.

5. Application saves new clinic to database.

**Alternative Sequence:**

Step 4.1: Admin can cancel the process.

Step 5.1: if any error occurs application display error page.

**Postcondition:** Clinic added successfully and admin able to add that clinic to hospital.

**Use case ID:** 18

**Use case name:** Add doctor to clinic of hospital

**Summary:** Admin add a new doctor to created hospital and clinic in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Hospital and clinic should added.

**Main sequence:**

1. Admin select add doctor process from dashboard.

2. Application loads add doctor page.

3. Admin fills the required fields.

4. Admin click to save.

5. Application saves new doctor to database.

**Alternative Sequence:**

Step 4.1: Admin can cancel the process.

Step 5.1: if any error occurs application display error page.

**Postcondition:** Doctor added successfully and users are able to reach that doctor also admin able to see that doctor.

**Use case ID:** 19

**Use case name:** Create Schedule to doctor

**Summary:** Admin add a new work day to created doctor in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Doctor should added.

**Main sequence:**

1. Admin select add days process from dashboard.

2. Application loads add day page.

3. Admin fills the required fields.

4. Admin click to save.

5. Application saves new appointment day to database.

**Alternative Sequence:**

Step 4.1: Admin can cancel the process.

Step 5.1: if any error occurs application display error page.

**Postcondition:** Schedule added successfully and users are able to reach that schedule also admin able to see that schedule.

**Use case ID:** 20

**Use case name:** Edit country

**Summary:** Admin edit country information in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Country should exist.

**Main sequence:**

1. Admin select list of countries process from dashboard.

2. Application loads list of countries page.

3. Admin select update process.

4. Admin fills the required fields.

5. Admin click to save.

6. Application saves changes on related country on database.

**Alternative Sequence:**

Step 5.1: Admin can cancel the process.

Step 6.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related country and new data can see by admin and user.

**Use case ID:** 21

**Use case name:** Edit city

**Summary:** Admin edit city information in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. City should exist.

**Main sequence:**

1. Admin select list of cities process from dashboard.

2. Application loads list of cities page.

3. Admin select update process.

4. Admin fills the required fields.

5. Admin click to save.

6. Application saves changes on related city on database.

**Alternative Sequence:**

Step 5.1: Admin can cancel the process.

Step 6.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related city and new data can see by admin and user.

**Use case ID:** 22

**Use case name:** Edit hospital

**Summary:** Admin edit hospital information in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Hospital should exist.

**Main sequence:**

1. Admin select list of hospitals process from dashboard.

2. Application loads list of hospitals page.

3. Admin select update process.

4. Admin fills the required fields.

5. Admin click to save.

6. Application saves changes on related hospital on database.

**Alternative Sequence:**

Step 5.1: Admin can cancel the process.

Step 6.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related hospital and new data can see by admin and user.

**Use case ID:** 23

**Use case name:** Edit clinic

**Summary:** Admin edit clinic information in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Clinic should exist.

**Main sequence:**

1. Admin select list of clinics process from dashboard.

2. Application loads list of clinics page.

3. Admin select update process.

4. Admin fills the required fields.

5. Admin click to save.

6. Application saves changes on related clinic on database.

**Alternative Sequence:**

Step 5.1: Admin can cancel the process.

Step 6.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related clinic and new data can see by admin and user.

**Use case ID:** 24

**Use case name:** Edit doctor

**Summary:** Admin edit doctor information in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Doctor should exist.

**Main sequence:**

1. Admin select list of doctors process from dashboard.

2. Application loads list of doctors page.

3. Admin select update process.

4. Admin fills the required fields.

5. Admin click to save.

6. Application saves changes on related doctor on database.

**Alternative Sequence:**

Step 5.1: Admin can cancel the process.

Step 6.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related doctor and new data can see by admin and user.

**Use case ID:** 25

**Use case name:** Delete country

**Summary:** Admin delete country information from system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Country should exist.

**Main sequence:**

1. Admin select list of countries process from dashboard.

2. Application loads list of countries page.

3. Admin select delete process.

4. Application deletes related country from database.

**Alternative Sequence:**

Step 4.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related country and new data can see by admin and user.

**Use case ID:** 26

**Use case name:** Delete city

**Summary:** Admin delete city information from system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. City should exist.

**Main sequence:**

1. Admin select list of cities process from dashboard.

2. Application loads list of cities page.

3. Admin select delete process.

4. Application deletes related city from database.

**Alternative Sequence:**

Step 4.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related city and new data can see by admin and user.

**Use case ID:** 27

**Use case name:** Delete clinic

**Summary:** Admin delete clinic information from system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Clinic should exist.

**Main sequence:**

1. Admin select list of clinics process from dashboard.

2. Application loads list of clinics page.

3. Admin select delete process.

4. Application deletes related clinic from database.

**Alternative Sequence:**

Step 4.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related clinic and new data can see by admin and user.

**Use case ID:** 28

**Use case name:** Delete hospital

**Summary:** Admin delete hospital information from system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Hospital should exist.

**Main sequence:**

1. Admin select list of hospitals process from dashboard.

2. Application loads list of hospitals page.

3. Admin select delete process.

4. Application deletes related hospital from database.

**Alternative Sequence:**

Step 4.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related hospital and new data can see by admin and user.

**Use case ID:** 29

**Use case name:** Delete doctor

**Summary:** Admin delete doctor information from system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Doctor should exist.

**Main sequence:**

1. Admin select list of doctors process from dashboard.

2. Application loads list of doctors page.

3. Admin select delete process.

4. Application deletes related doctor from database.

**Alternative Sequence:**

Step 4.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related doctor and new data can see by admin and user.

**Use case ID:** 30

**Use case name:** Delete user

**Summary:** Admin delete user information from system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. User should exist.

**Main sequence:**

1. Admin select list of users process from dashboard.

2. Application loads list of users page.

3. Admin select delete process.

4. Application deletes related user from database.

**Alternative Sequence:**

Step 4.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related user and new data can see by admin and user.

**Use case ID:** 31

**Use case name:** Delete schedule

**Summary:** Admin delete doctor schedule information from system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. Appointment day should exist.

**Main sequence:**

1. Admin select list of days process from dashboard.

2. Application loads list of days page.

3. Admin select delete process.

4. Application deletes related days from database.

**Alternative Sequence:**

Step 4.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related appointment day and new data can see by admin and user.

**Use case ID:** 32

**Use case name:** Edit schedule

**Summary:** Admin edit schedule information of doctor in system.

**Actor:** System Admin.

**Dependency:** Login.

**Precondition:** Application should work correctly. schedule should exist.

**Main sequence:**

1. Admin select list of days process from dashboard.

2. Application loads list of days page.

3. Admin select update process.

4. Admin fills the required fields.

5. Admin click to save.

6. Application saves changes on related day on database.

**Alternative Sequence:**

Step 5.1: Admin can cancel the process.

Step 6.1: if any error occurs application display error page.

**Postcondition:** Information successfully uppdated on related days and new data can see by admin and user.

## Data Flow Diagram

I designed data flow diagrams for Non Member User, Member User and System Admin which can use main scenarios of application and follow all the steps in application. Here is the data flow diagrams that shows processes and steps of application that users follow.

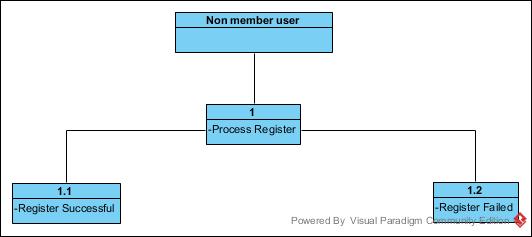


Figure 2 - Data Flow Diagram for Non Member User

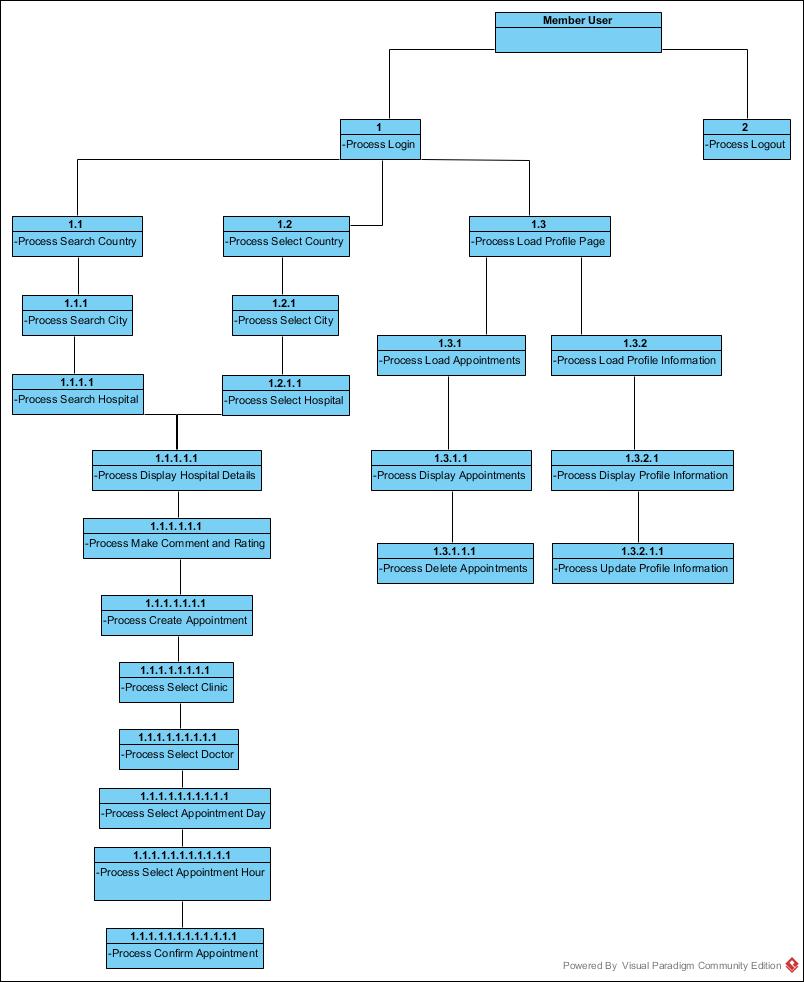


Figure 3 - Data Flow Diagram for Member User

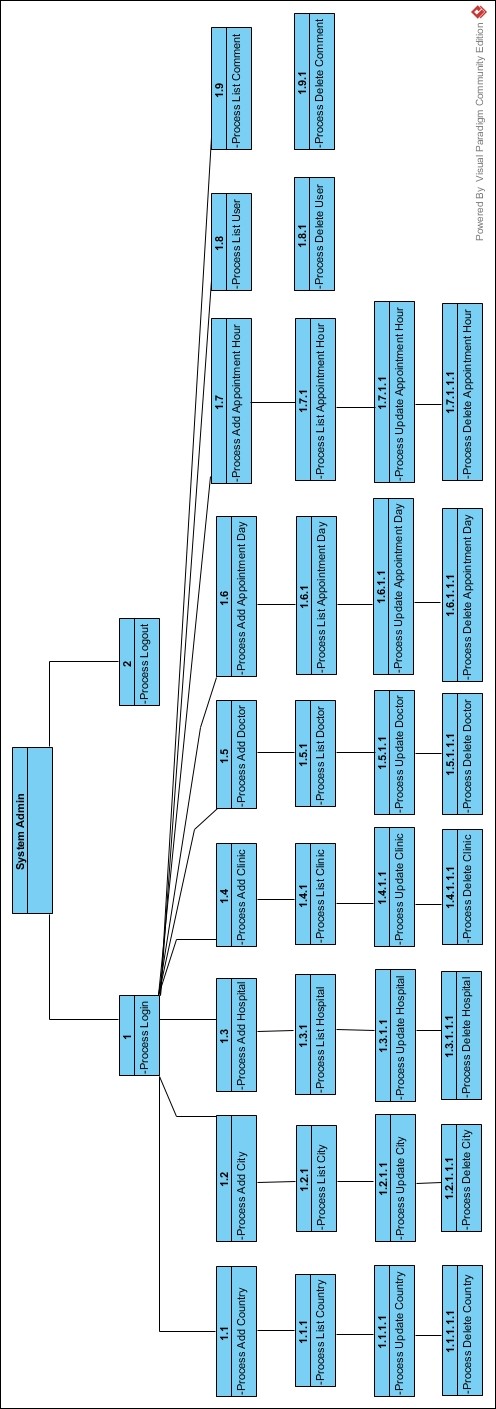


Figure 4 - Data Flow Diagram for System Admin

## Flow Chart

I designed this flow chart to present user actions. It presents Member User actions. Decision tree shapes next event of user can do. Events starts when user enters the website and then the decision trees control some options that can happen. When ever user comes to end process is end or user can finish directly by terminate the website.

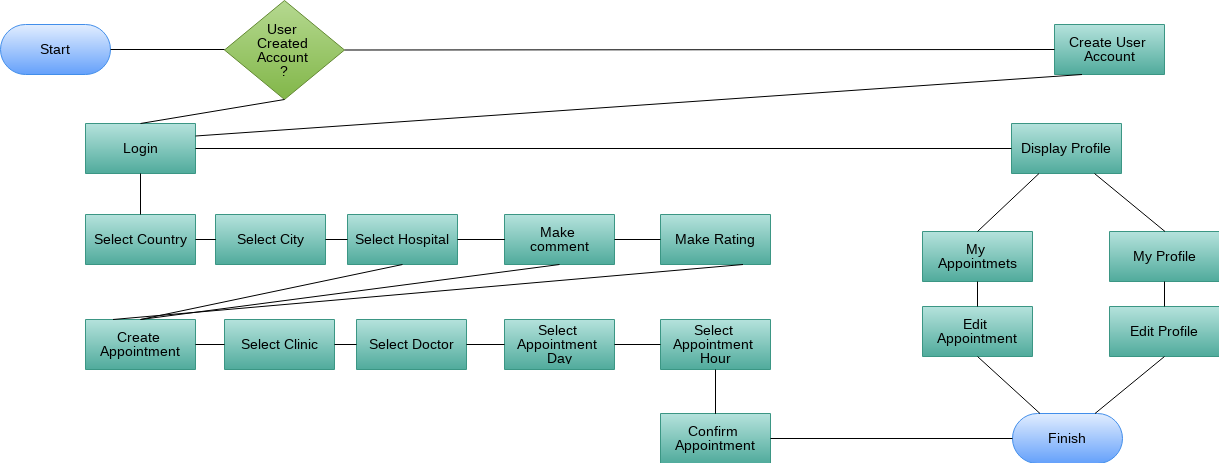


Figure 5 - Flow Chart Diagram

## Software Architecture Diagram

I present system’s architecture design on that diagram. It basicly refers the all the layers that using when the application run on internet host.

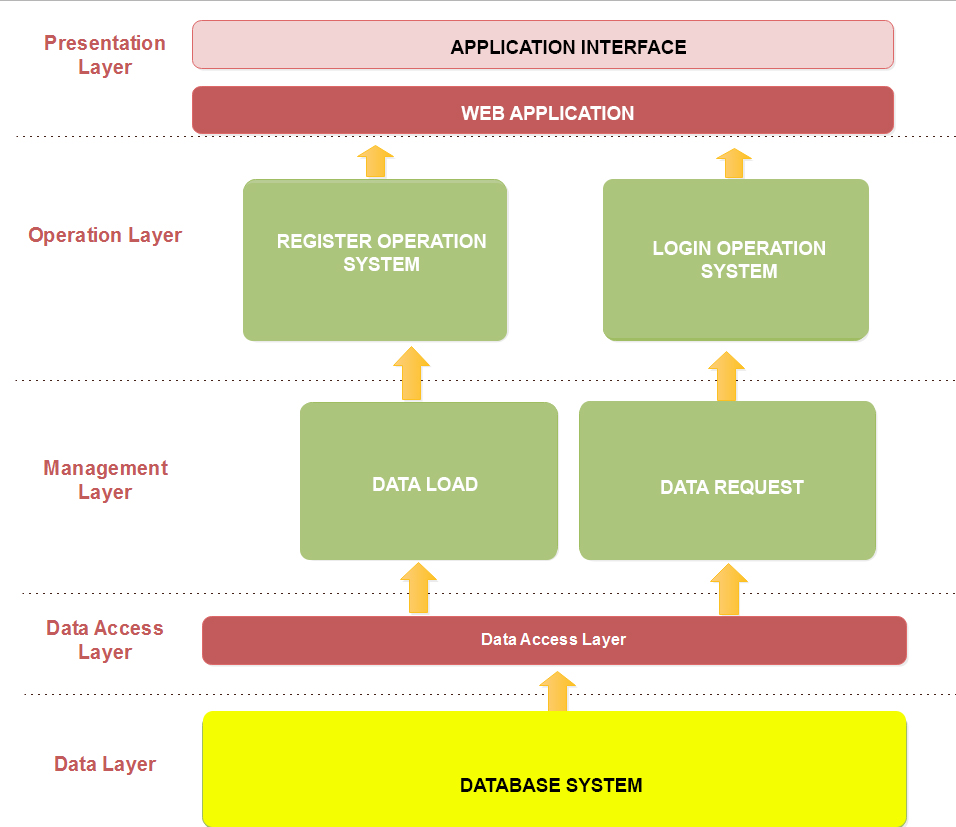


Figure 6 - Architecture Diagram

## UML Class Diagrams

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

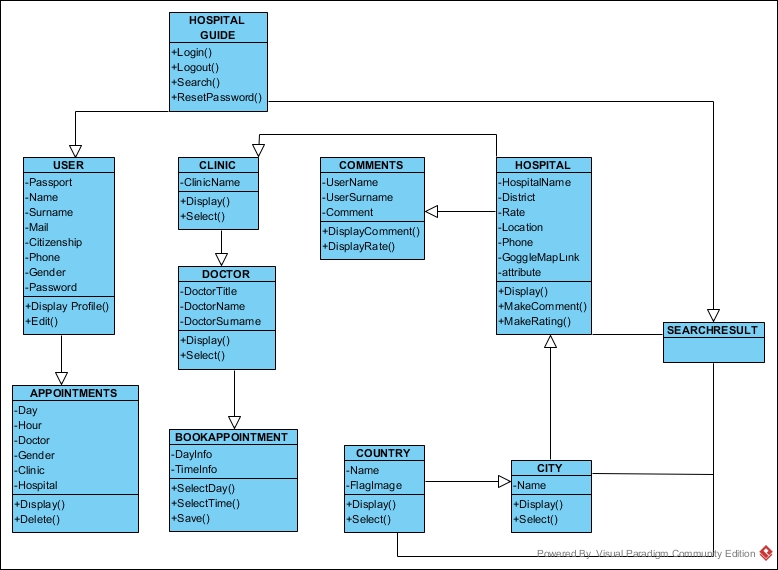


Figure 7 - Class Diagram

## Database Relationship Diagram

I designed a database relationship diagram. This figure shows it in detail. You can see tables, primary keys, relations and fields in database.

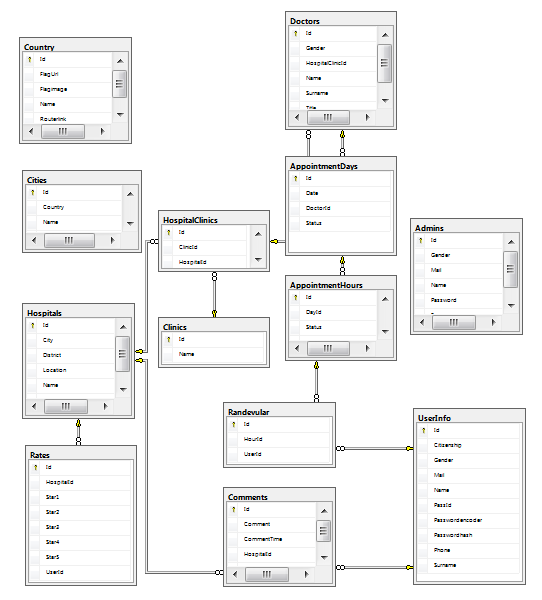


Figure 8 - Database Relationship Diagram

**CHAPTER 3:**

**ANALYSIS**

# REQUIREMENT ANALYSIS

This chapter will provide full description about the Project requirement analysis, requirement development, system requirements, functional requirements, project purpose and phases before start to build project.

## Main Purpose of The Project.

This project have one main purpose which is support foreign visitors in different countries all around the World. In the beginning of that project the idea was very simple. People have problems on that field and a solutionary system should solve their problems. I develop that project to give full support all around the World. If countries support the system it can really help all the people that travel any country. The logic and usage of the system is very simple and i create lots of functions to get and use every information that i need to access.

## Requirements Development

To develop the functional and non­functional requirements of the proposed system, brainstorming sessions were held with supervisor of that project. Through these sessions, the system was analyzed and some of the requirements were generated according to the needs.

## System Requirements

Requirements are the necessary attributes in the system, a statement that identifies a capability, characteristic or quality factor of the system in order to have value and utility to the users. Once the requirements are set, developers can initiate the other technical work including system design, development, testing, implementation, and operation. For any system, there are functional and non­functional requirements to be considered while determining the requirements of the system On the other hand, nonfunctional requirements are requirements that describe how the system will do what it is supposed to do, for example, security, reliability and maintainability.

## Functional Requirements

The functional requirements have studied by doing domain research on existing similar project. It show me what is important and what kind of action should think carefully. Similar project gave me lots of perspective and created me a road map on that project. But as i mentioned previously that kind of system is not exist. So i filled the blank on my domain researh. Actually, functional requirements describe what the system should do.It is main purpose.

## User Requirements

Main user requirements to use the project is a internet connection and a device. This can be a mobil phone, tablet, computer, etc. When people have these kind of things they can Access the system. But for start to use the system users should have a passport and mail adress. So they can create an account and use the application.

## Non Functional Requirements

Mainly it can considered as fuction or features that system should not loss its main purpose. This should be considered very carefully because if I apply wrong function priorty to application it may affect the system purpose in negatively. So cause to extra effort to do job. Main ideas about Non functional requirements Unnecessary fuctions should considired and work on another field.

## Priority Fields on Project

* Performance
* Usability
* Security
* Reliability
* Interoperability
* Compatibility
* Effectiveness
* Supportability

**CHAPTER 4:**

**TOOLS AND TECHNOLOGY**

# PROJECT TOOLS

## MS SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

### Inside sql server's architecture

Like other RDBMS technologies, SQL Server is primarily built around a row-based table structure that connects related data elements in different tables to one another, avoiding the need to redundantly store data in multiple places within a database. The relational model also provides referential integrity and other integrity constraints to maintain data accuracy; those checks are part of a broader adherence to the principles of atomicity, consistency, isolation and durability collectively known as the [ACID properties](https://searchsqlserver.techtarget.com/definition/ACID) and designed to guarantee that database transactions are processed reliably.

### Usage of sql server

* To create databases.
* To maintain databases.
* To analyze the data through SQL Server Analysis Services (SSAS).
* To generate reports through SQL Server Reporting Services (SSRS).
* To carry out ETL operations through SQL Server Integration Services (SSIS).

## Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle

### .NET

NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (in contrast to a hardware environment) named Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. As such, computer code written using .NET Framework is called "managed code". FCL and CLR together constitute the .NET Framework.

FCL provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their source code with .NET Framework and other libraries. The framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

### ASP.NET Core

ASP.NET Core is a free and open-source web framework, and higher performance than ASP.NET, developed by Microsoft and the community. It is a modular framework that runs on both the full .NET Framework, on Windows, and the cross-platform .NET Core.

The framework is a complete rewrite that unites the previously separate ASP.NET MVC and ASP.NET Web API into a single programming model.

Despite being a new framework, built on a new web stack, it does have a high degree of concept compatibility with ASP.NET MVC. ASP.NET Core applications supports side by side versioning in which different applications, running on the same machine, can target different versions of ASP.NET Core. This is not possible with previous versions of ASP.NET.

## Angular

Angular is a TypeScript-based open-source front-end web application framework led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS.

### Differences between Angular and AngularJS

Angular was a ground-up rewrite of AngularJS.

* Angular does not have a concept of "scope" or controllers, instead it uses a hierarchy of components as its primary architectural characteristic.
* Angular has a different expression syntax, focusing on "[ ]" for property binding, and "( )" for event binding
* Modularity – much core functionality has moved to modules
* Angular recommends the use of Microsoft's TypeScript language.
* TypeScript is a superset of ECMAScript 6 (ES6), and is backwards compatible with ECMAScript 5 (i.e.: JavaScript).
* Dynamic loading
* Asynchronous template compilation
* Iterative callbacks provided by RxJS. RxJS limits state visibility and debugging, but these can be solved with reactive add-ons like ngReact or ngrx.
* Support
* Support Angular Universal, a technology that runs your Angular application on the server
* Has its own suite of modern UI components that work across the web, mobile and desktop, called Angular Material

## HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

## Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

# METHODS

## MVC Design

Model-View-Controller (MVC) is a term you often hear in the world of Software, Web and Mobile development, but it can be confusing if you do not understand what it means.

Model-View-Controller is essentially a design pattern that separates the different aspects of a piece of Software. This separation promotes code reusability and a more structured application architecture.

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

So, for example, you could breakdown a mobile app into 3 main components.

• Firstly, the “View” is what is presented to the User. This is storyboard in a view.

• Secondly, the “Model” is how the mobile app talks to the database.

• Thirdly, the “Controller” is what connects the “View” and the “Model”. Therefore, when a User requests something from the database, the “Controller” takes that request from the “View” and sends it to the “Model”

### MVC components

Following are the components of MVC



Figure 9 - MVC Components

#### Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

#### View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

#### Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

## Entity Framework

Entity Framework is an open-source ORM framework for .NET applications supported by Microsoft. It enables developers to work with data using objects of domain specific classes without focusing on the underlying database tables and columns where this data is stored. With the Entity Framework, developers can work at a higher level of abstraction when they deal with data, and can create and maintain data-oriented applications with less code compared with traditional applications.

Official Definition: “Entity Framework is an object-relational mapper (O/RM) that enables .NET developers to work with a database using .NET objects. It eliminates the need for most of the data-access code that developers usually need to write.”

**CHAPTER 5:**

**PROJECT DESIGN**

# USER

## Home Page

This is the homepage of website. When the users enters the direct link they access to that page. Main scenario starts from here. Users can Access to Login Register Contact fields directly see countries on system.

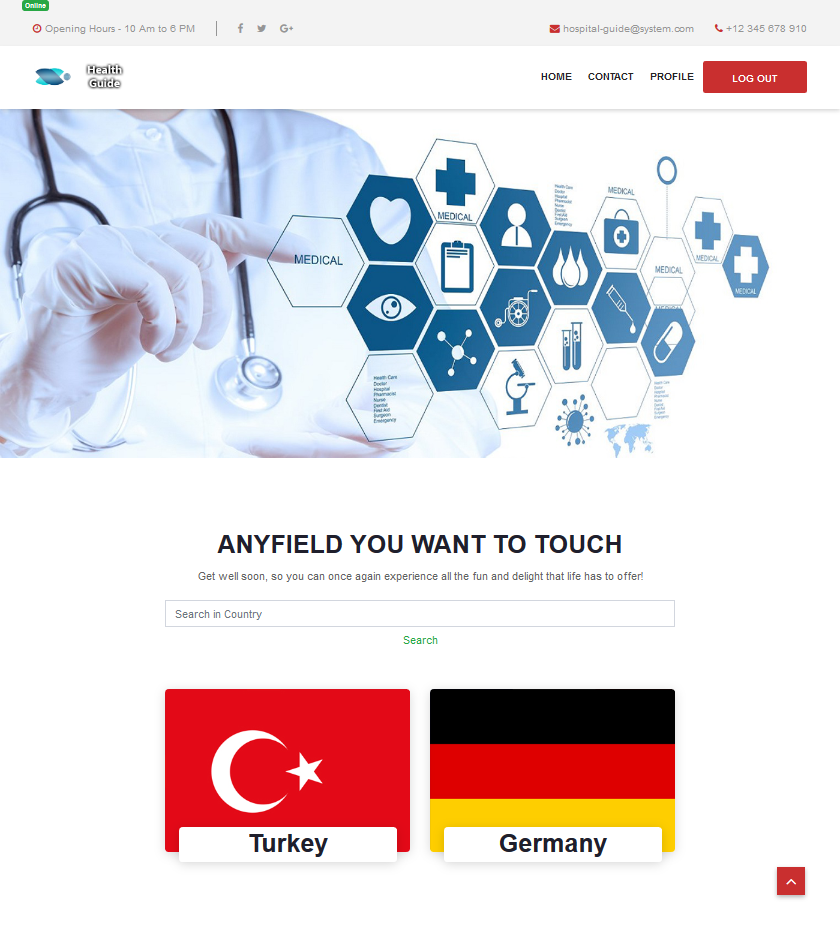


Figure - Home Page

## Register

Registerpage allows users to create an account to use the website fully. In this page user enter Passport number, Name, Surname, Mail, Citizenship, Phone, Password and Gender information.

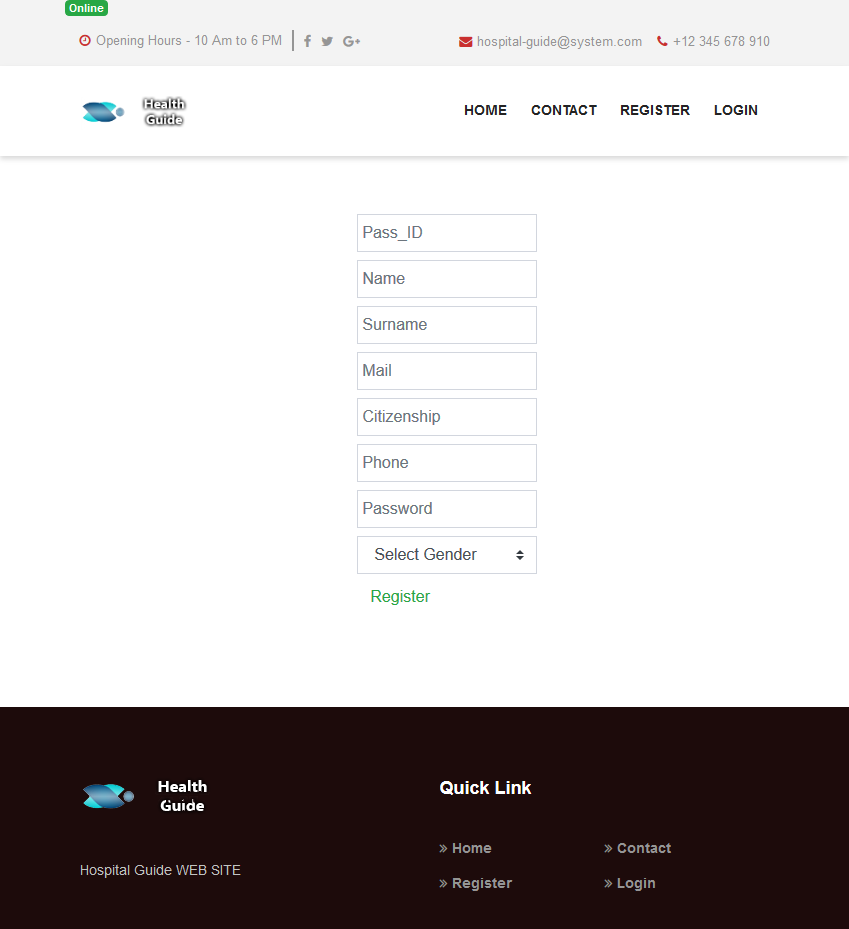


Figure - Register Page

## Login

In loginpage users enter their mail address and password that used while creating an account. In this section if users cant remember their password they can click to Forgot password section.

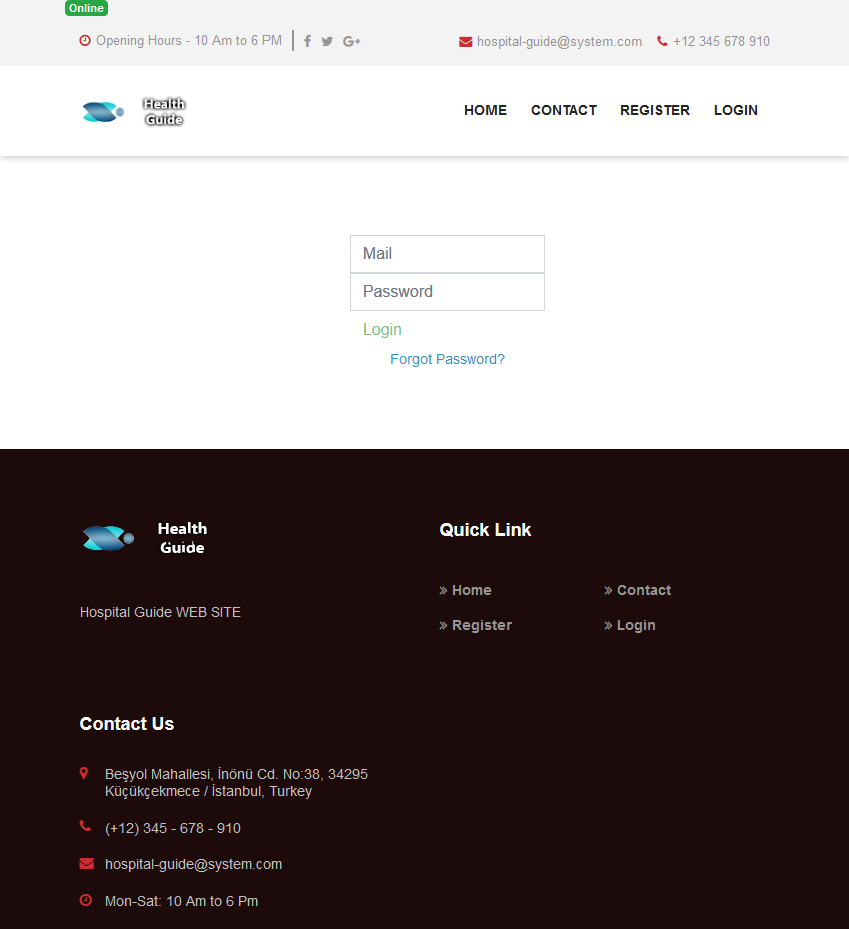
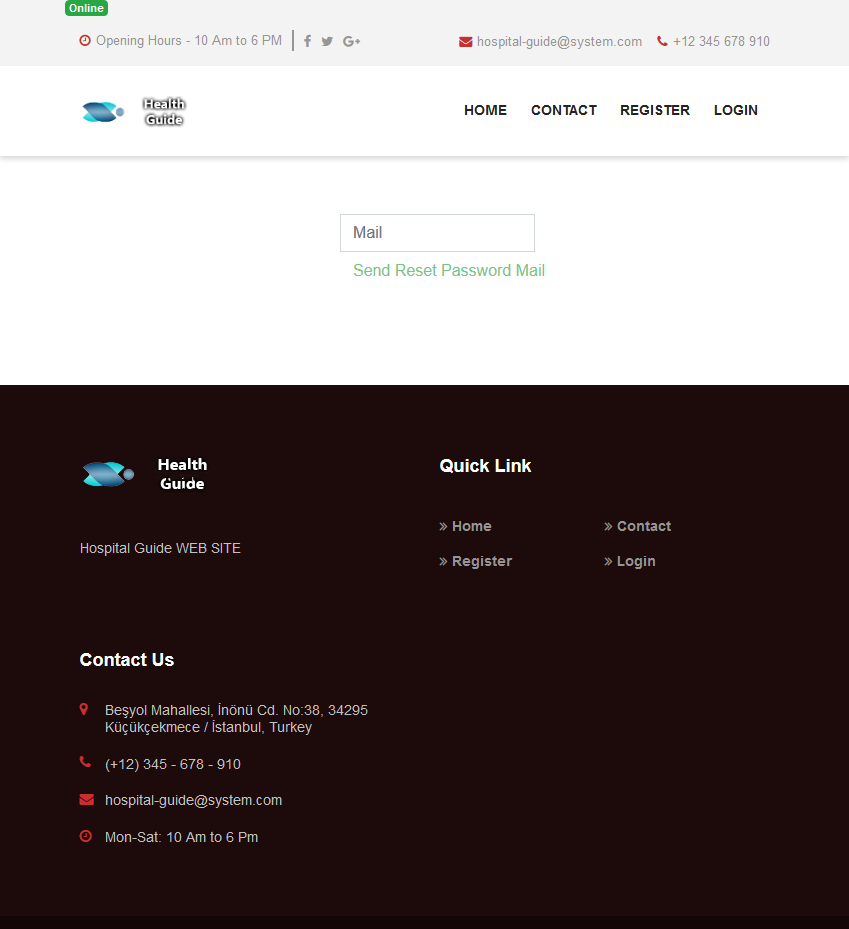


Figure - Login Page

## Forget Password

In this page users enter their mail address used in register section. And click to Send reset password mail button. After that application generate an email for that user and sends it to users’ mail adress. By using the link in the mail users can reset their password.



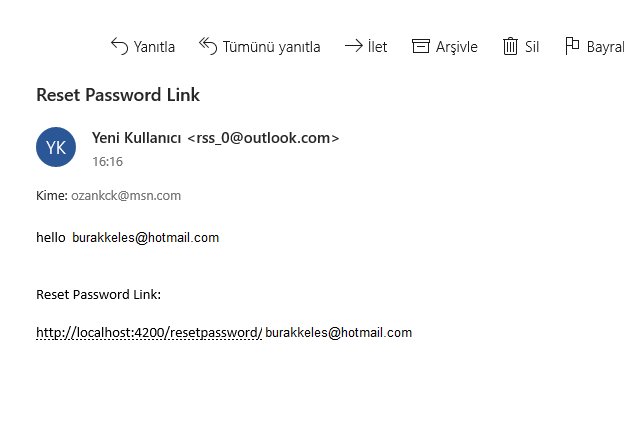
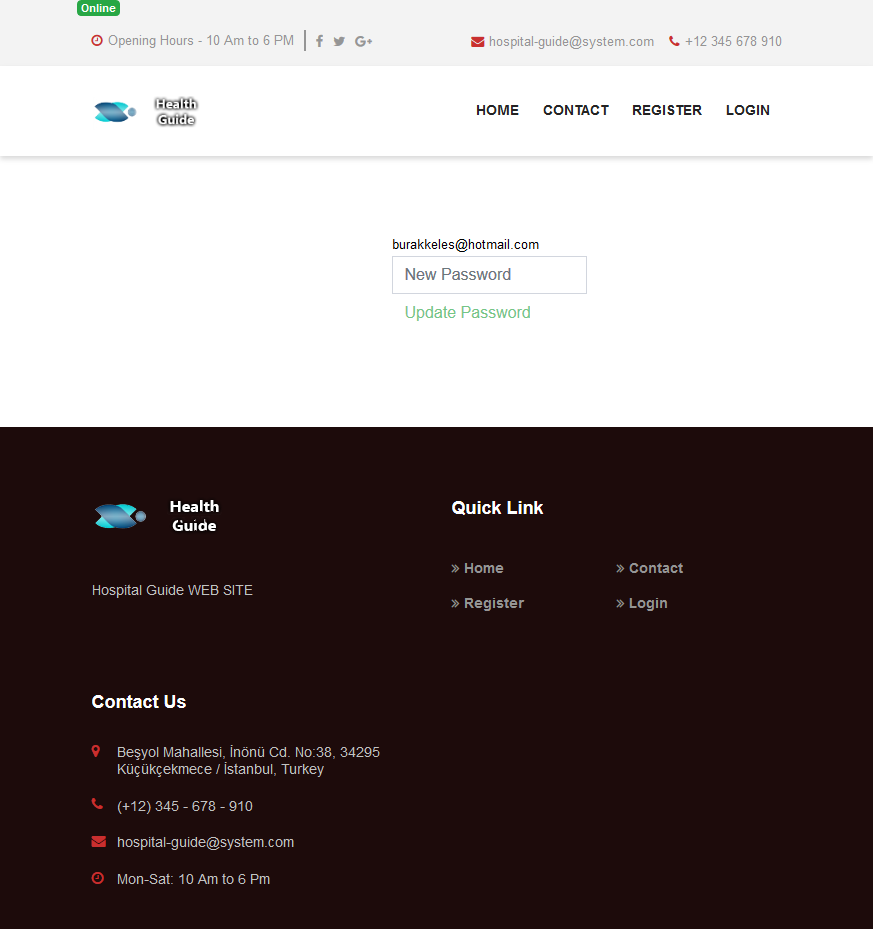


Figure - Forgot Password

## Reset Password

After users click the link in mail that resetpassword page will loads. Users enter a new password to field and click to update. After process complete application display an information page and users become able to enter to system by using new password.



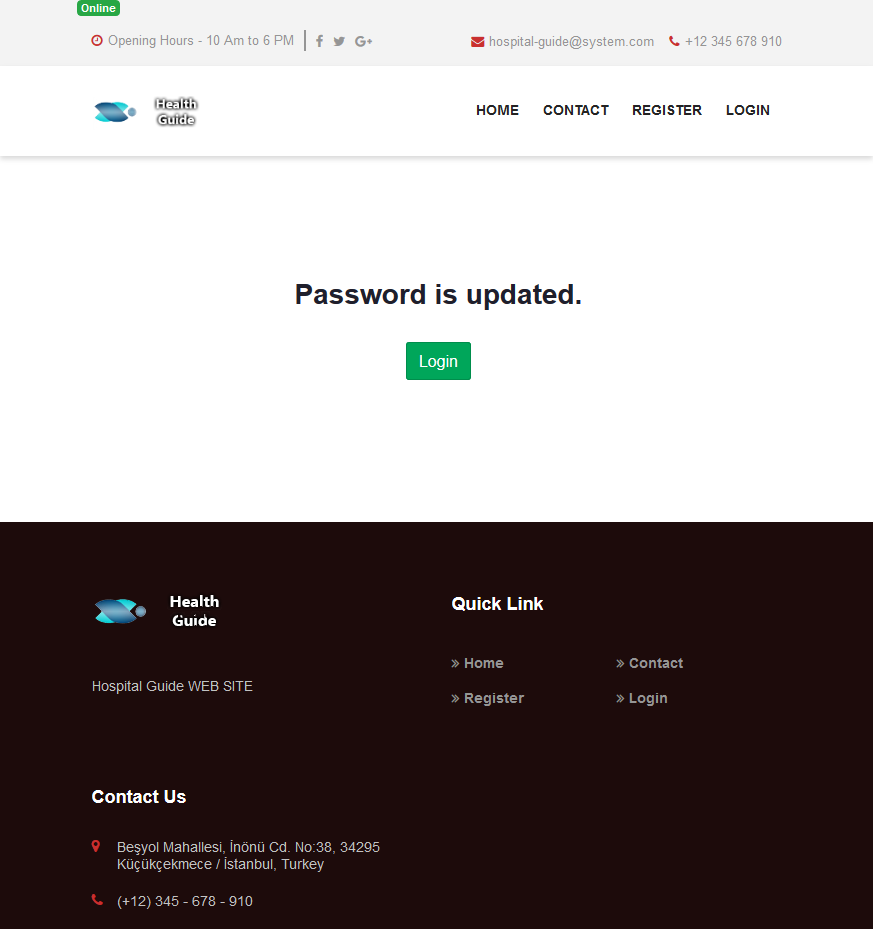


Figure - Reset Password

## My Profile

In this page users able to see their account information. If they want to change any information click to related field and enter new data to field and click update.

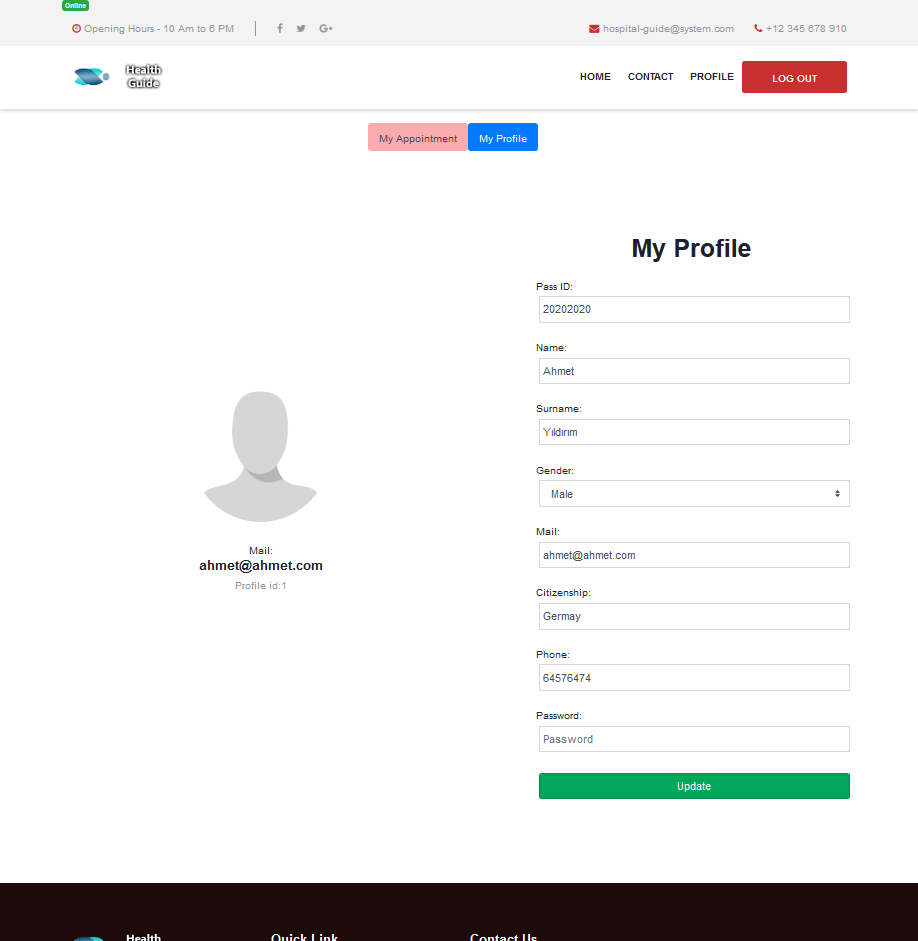


Figure - My Profile Page

## Cities

After users select the country they become able to select city in citiespage There are 5 tabs that helps user to categories cities by their name.

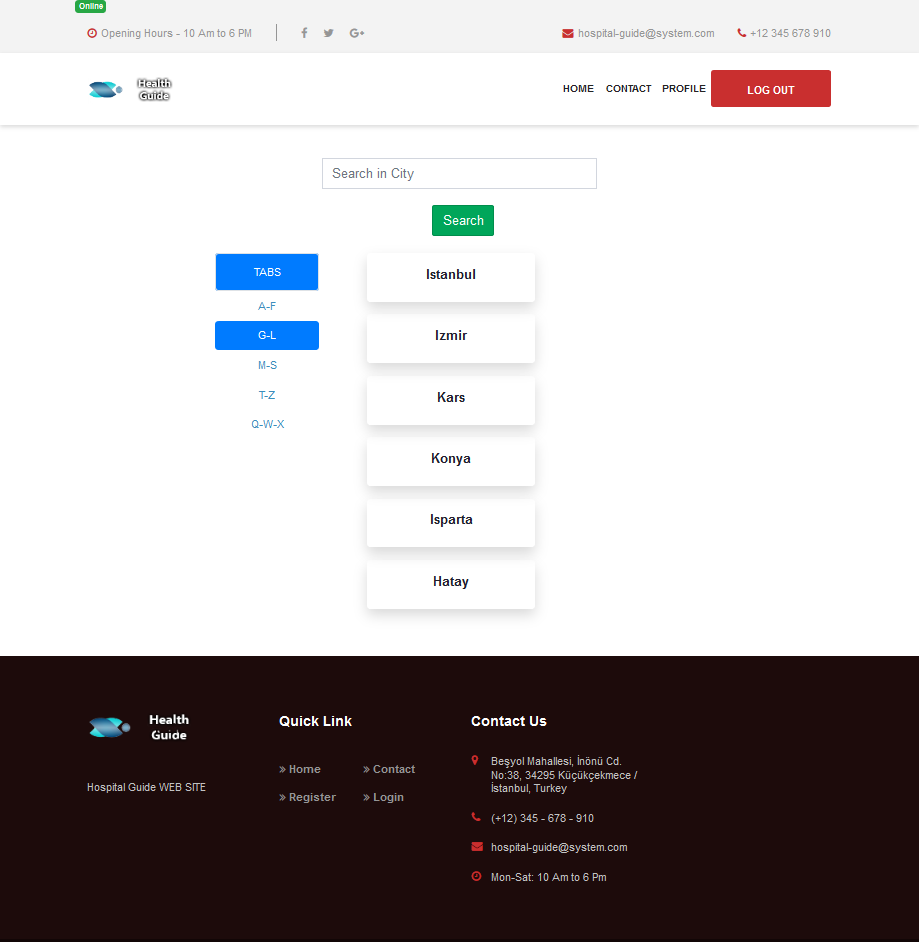


Figure - Cities Page

## Hospitals

After select city. Hospitalspage loads hospitals in that city. Users see the Hospital name, Its District and Rate information.

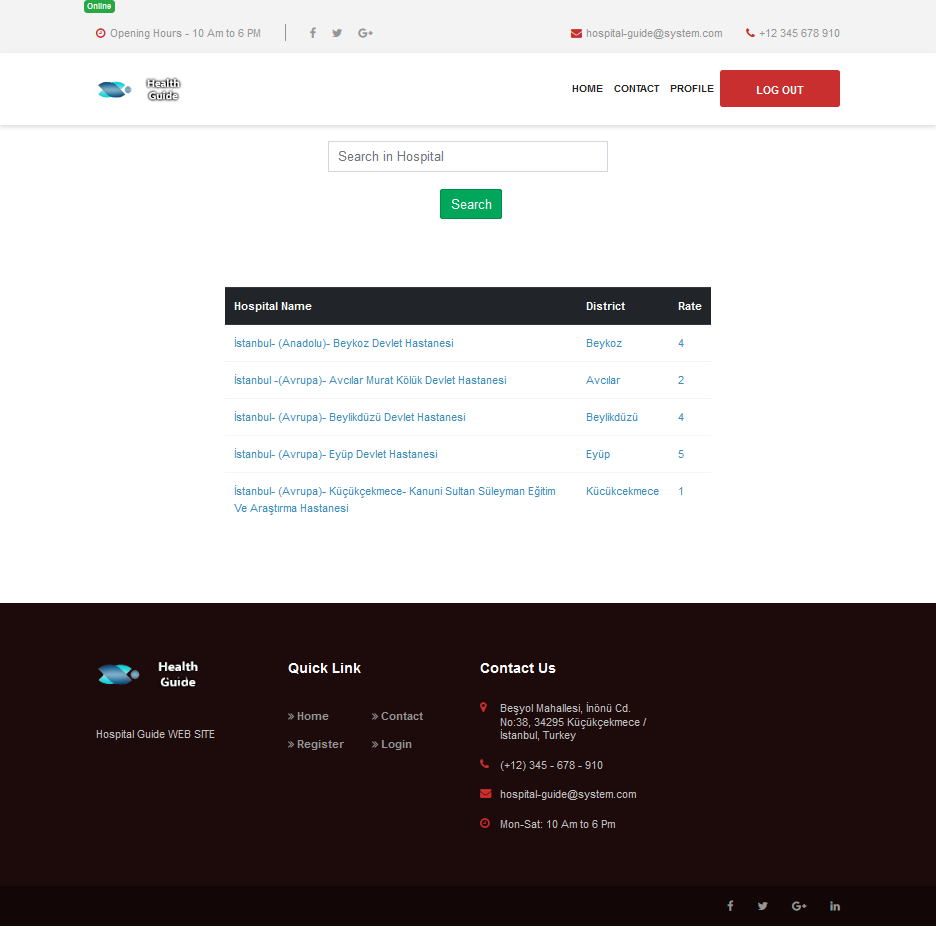


Figure - Hospitals Page

## Hospital Details

After select hospital from hospitalpage. Hospitaldetailpage load detailed information of related hospital and user can see. Full Location, Phone number, Users comments and rates for the hospital. They can make comment and rate if they wan to do or just click make an appointment button and create an appointment.

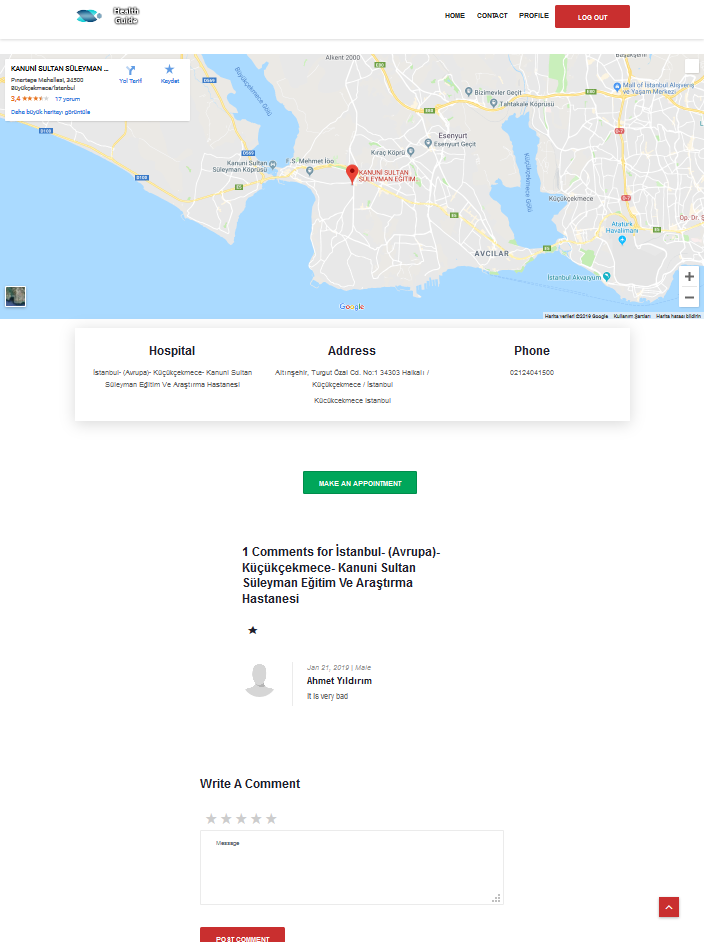


Figure - Hospital Detail Page

## Select Appointment

After user click to book an appointment application loads appointmentpage and users select their clinic, doctor, day and time informations from related fields.

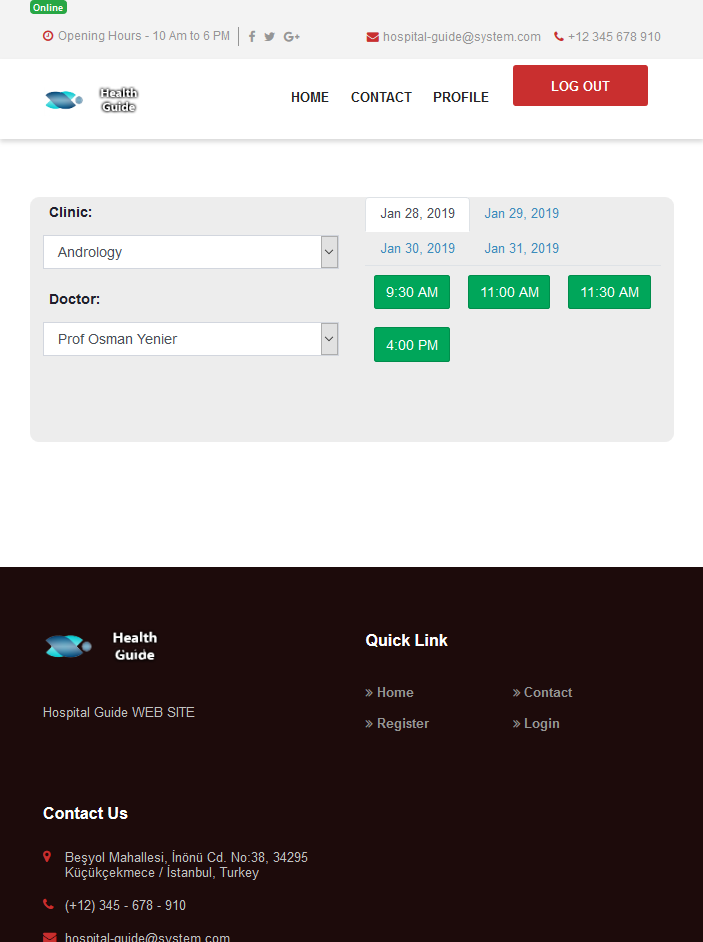


Figure - Create Appointment Page

## Appointment Confirmation

After select the appointment time application generates a pop-up screen and users see all the information of the appointmet. If users confirm the appointment appointmet saves to system. If user cancel the appointment application return to appointmentpage and user select new date and time on related doctor.

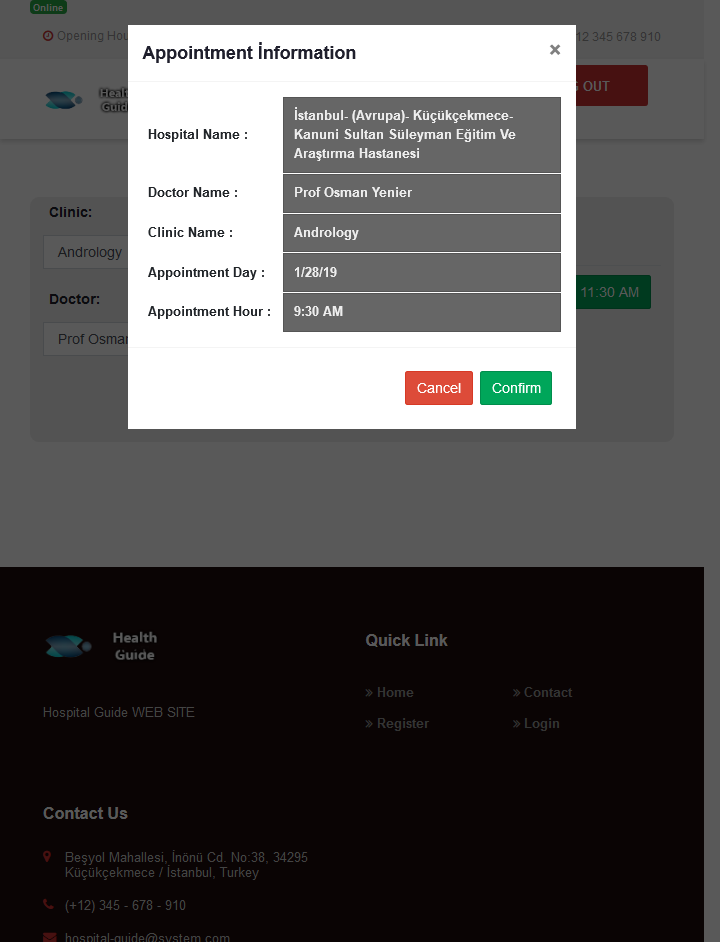


Figure - Confirm Appointment

## My Appointment

My appointmet page is accessable from profile tab. Users see their appointments on that field. If they want to cancel the appointment they can deletee the appointment.

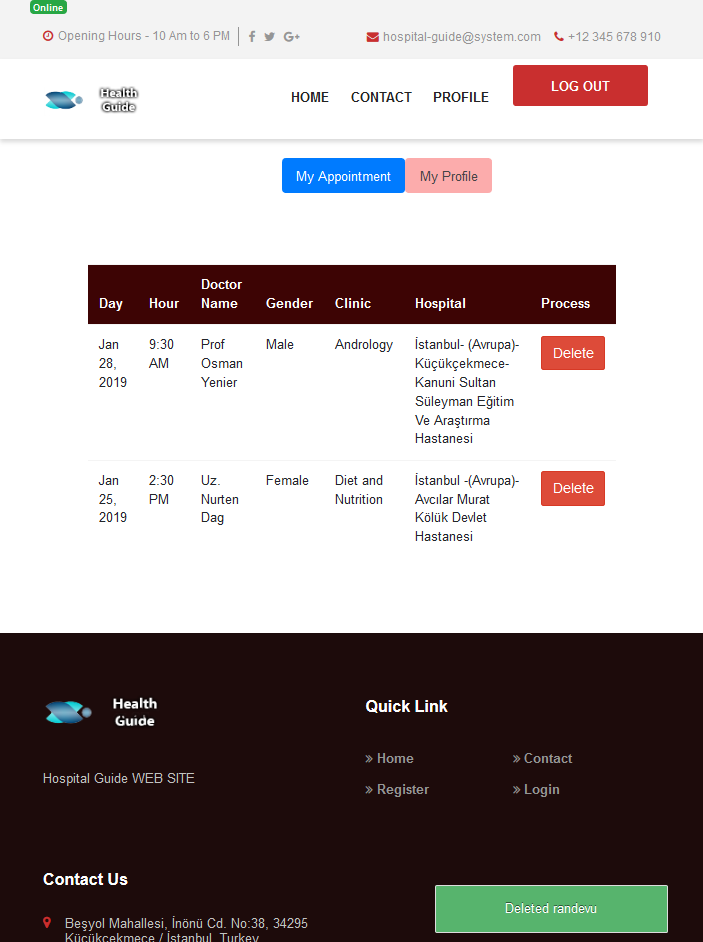


Figure - My Appointments Page

## Home Page Countries Search

Search field in country components can use for search a country.



Figure - Contry Search Field

## Cities Search

Search field in city components can use for search a city.

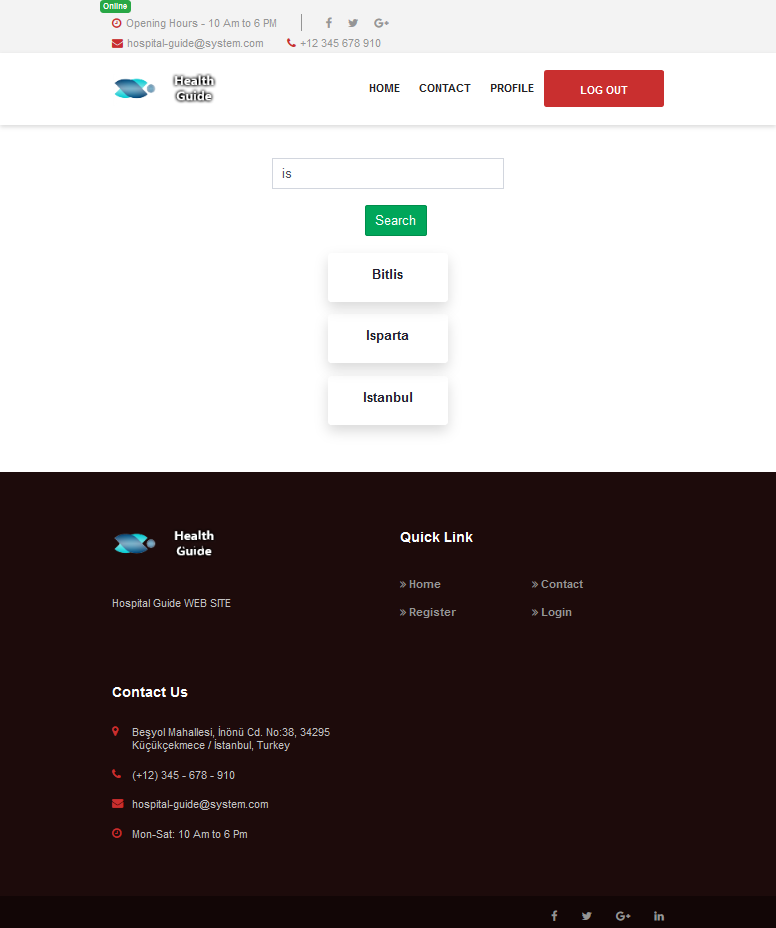


Figure - Search City Field

## Hospitals Serach

Search field in hospital components can use for search a hospital.

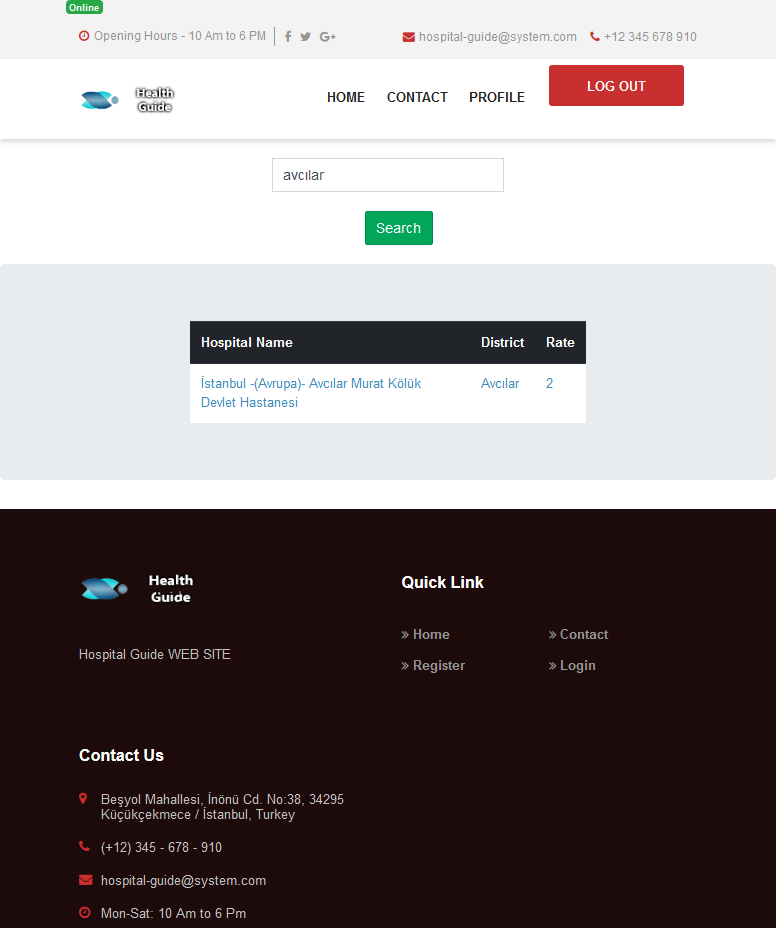


Figure - Search Hospital Field

# ADMİN

## Login

Admin login process done by that adminloginpage.

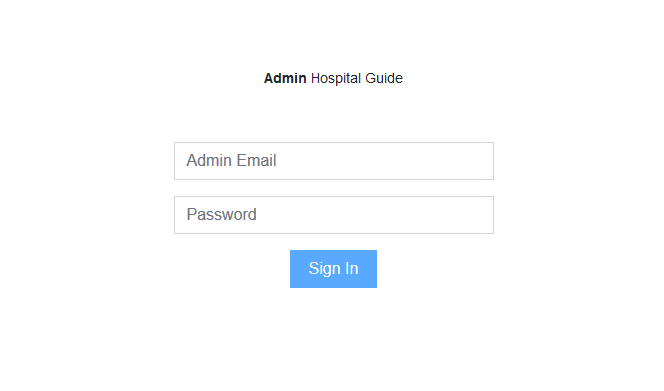


Figure - Admin Login

## Admin Dashboard

When admin successfuly login to system redirect to that daashboard. Admin able to see data counts on related sections.

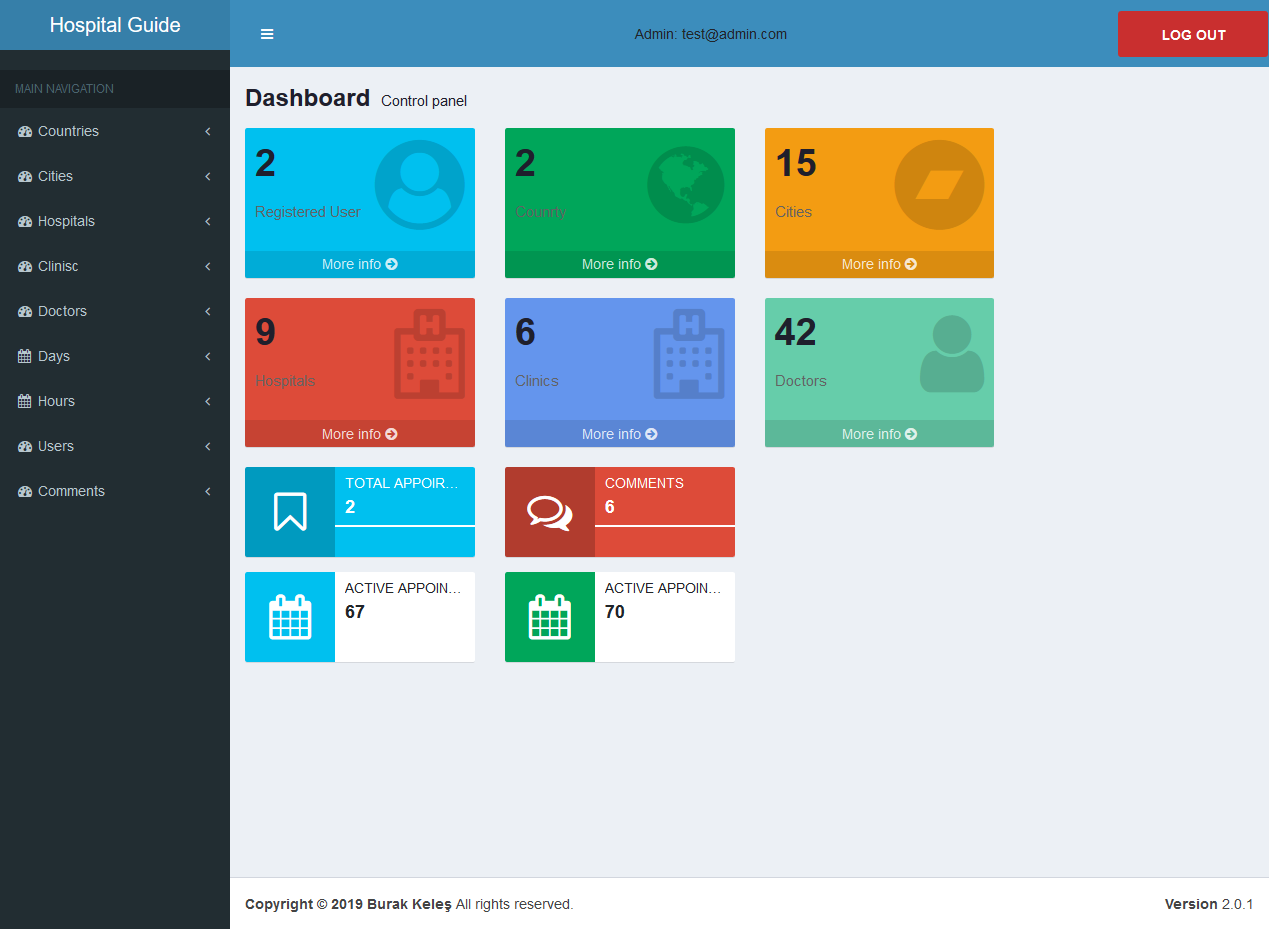


Figure - Admin Home Page

## Add Country

Admin selects the add country process from main navigation and website load related page.

By filling all the fields admin add a country to system.

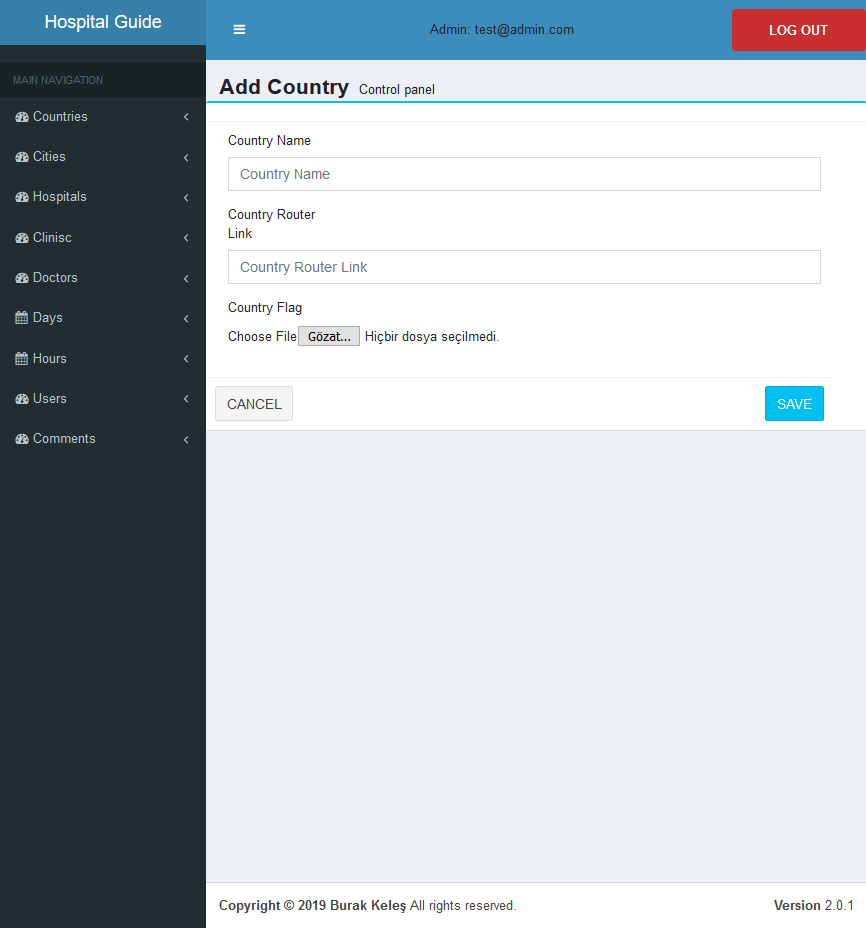


Figure - Add Country Page

## Counrty List

Added countries are listed in countrylist section. Admin can update country, delete country or list the cities of related country.

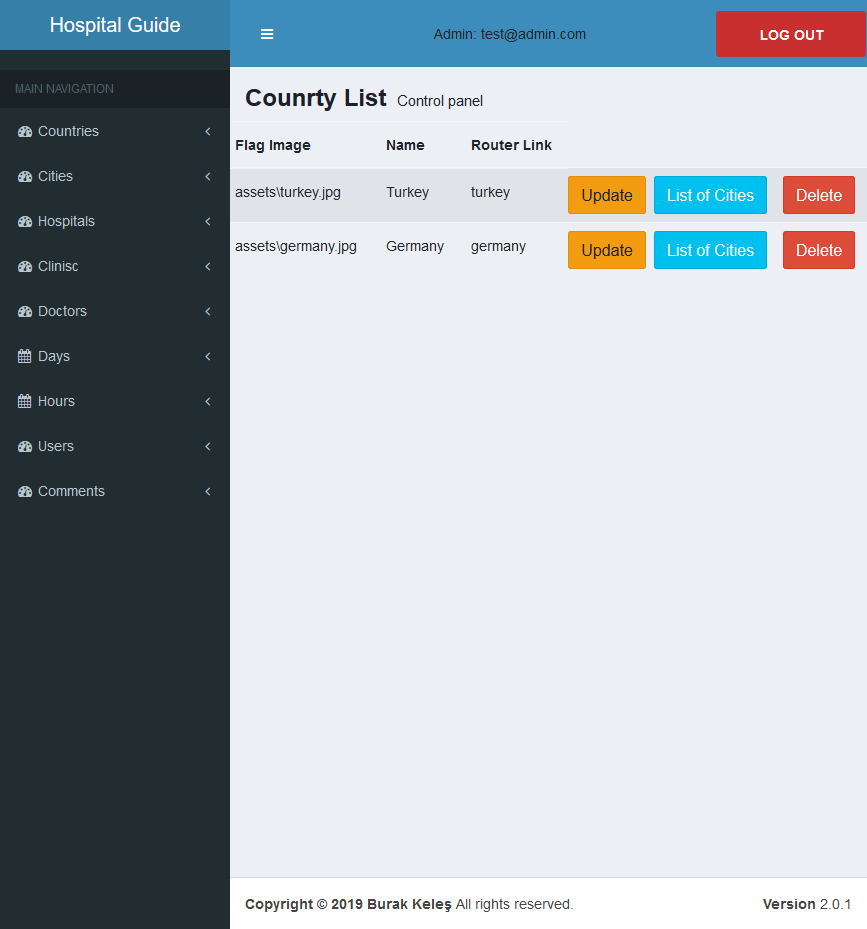


Figure - Country Page List

## Update Country

This is the page for the update information of country.

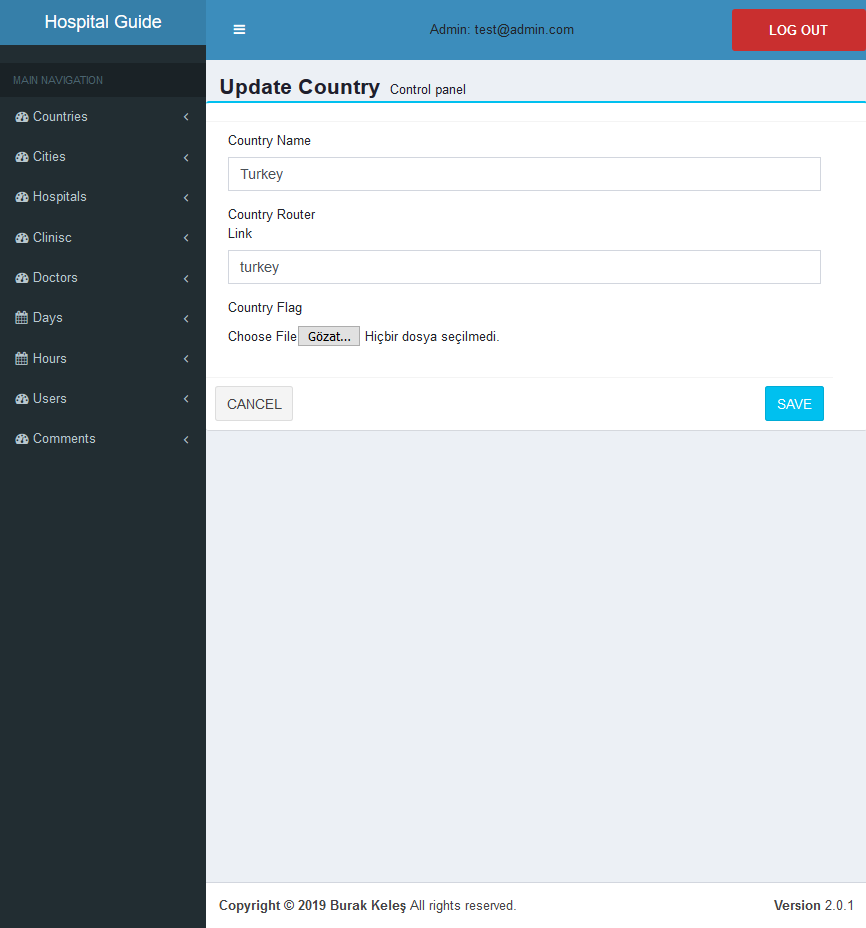


Figure - Update Country Page

## Add City

Admin can add city to a country by select add city process by main navigation and application loads related component. Admin enters information to related fields and save it.

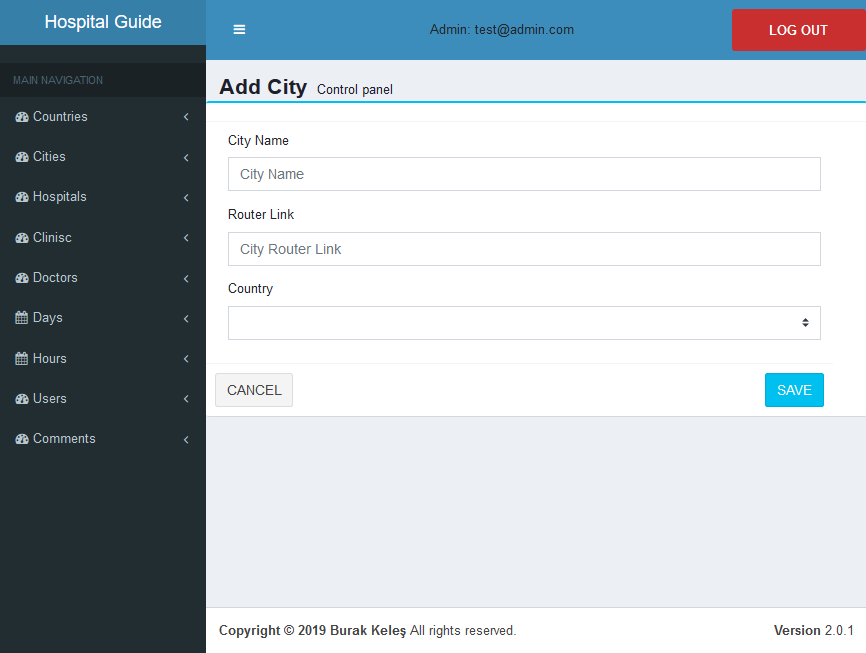


Figure - Add City Page

## City List

Added cities are listed in citylist section. Admin can update city, delete city or list the hospitals of related city.

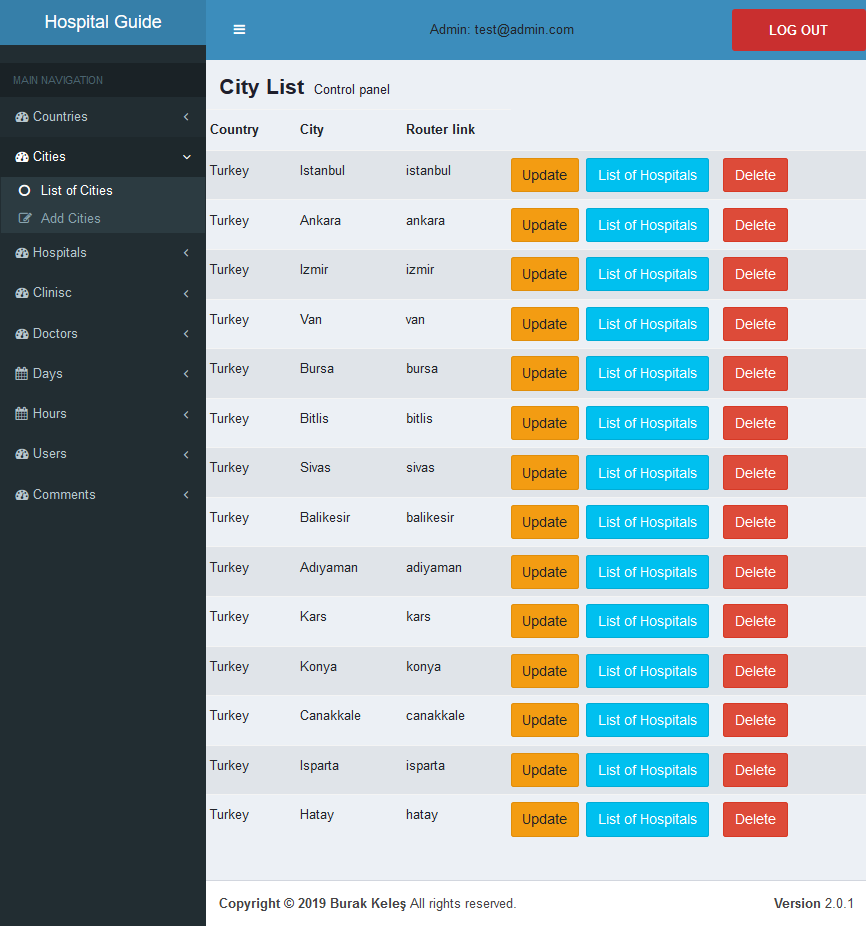


Figure - City List Page

## Update City

This is the page for the update information of city.

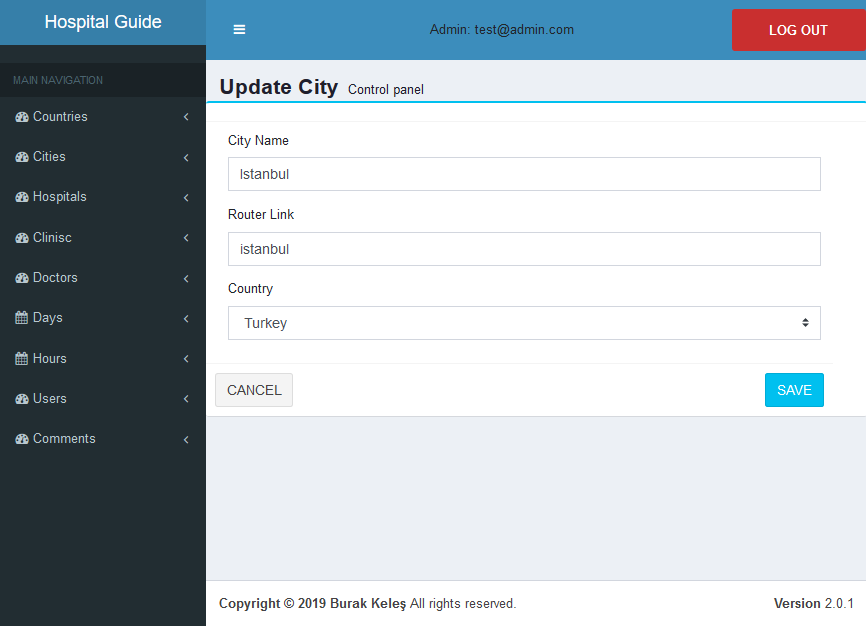


Figure - Update City

## Add Hospital

Admin can add hospital to a city. First select add hospital process from main navigation and then fills the fields.

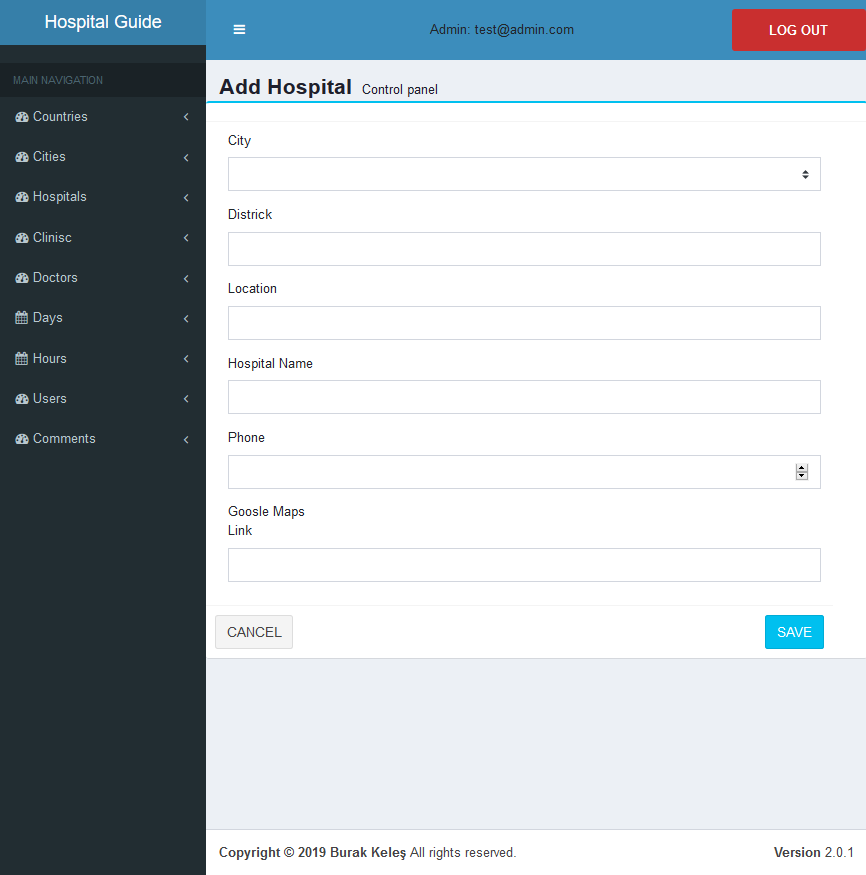


Figure - Add Hospital Page

## Hospitals List

Added hospitals are listed in citylist section. Admin can update hospital, delete hospital or list the clinics of related hospital.

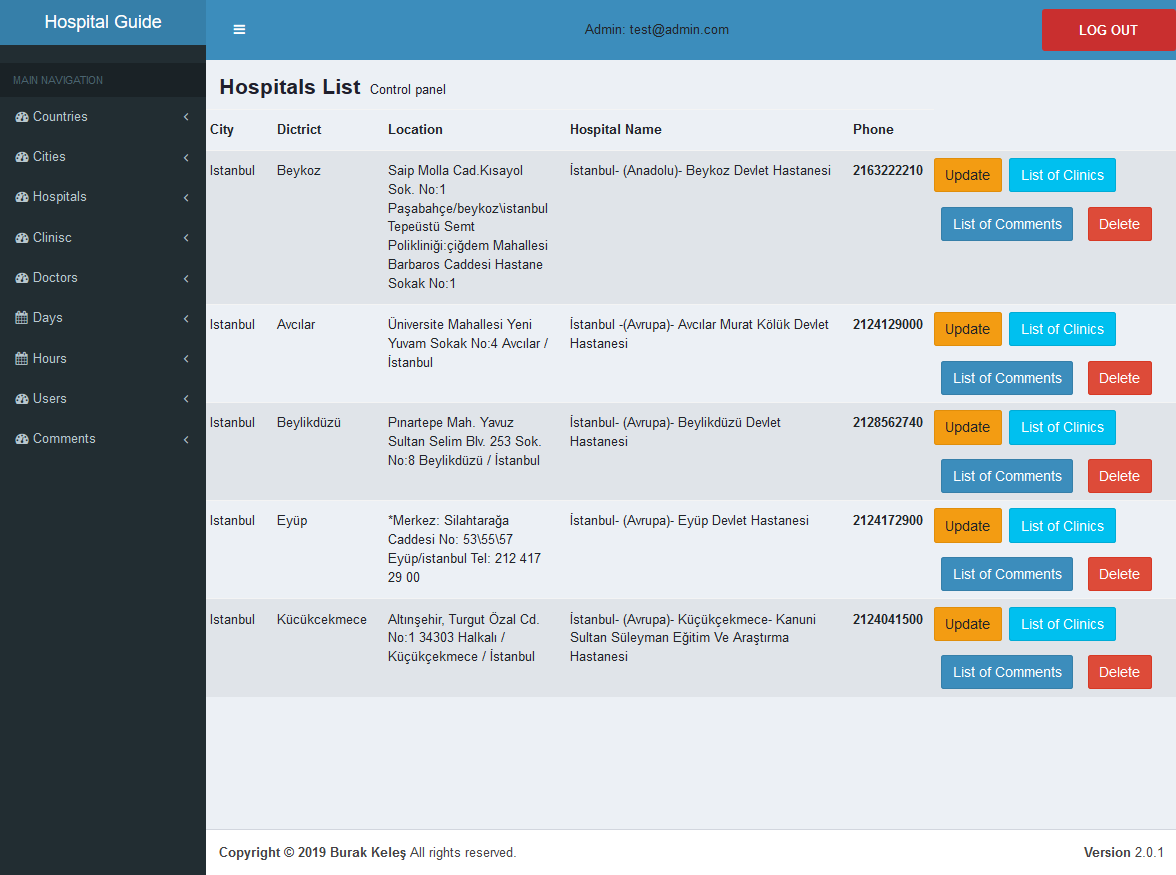


Figure - Hospital List Page

## Update Hospital

This is the page for the update information of hospital.

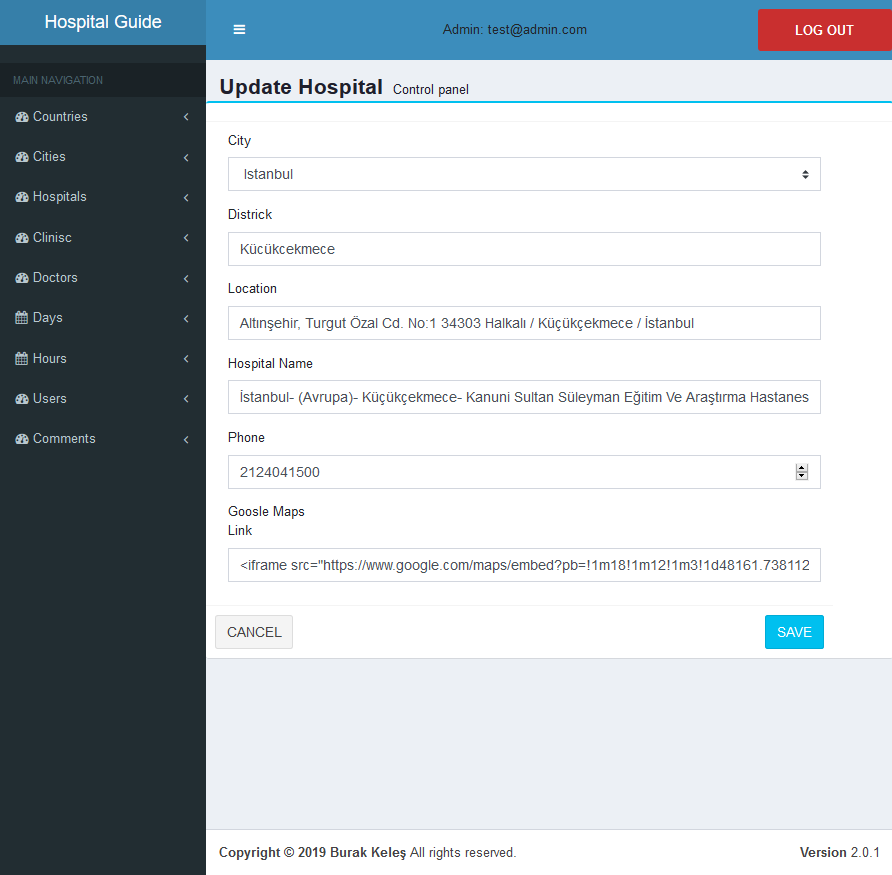


Figure - Update Hospital Page

## Add Clinic

Admin can add a clinic system. First select add clinic process from main navigation and then fills the related fields. After save the clinic clinic can be use for add doctor.

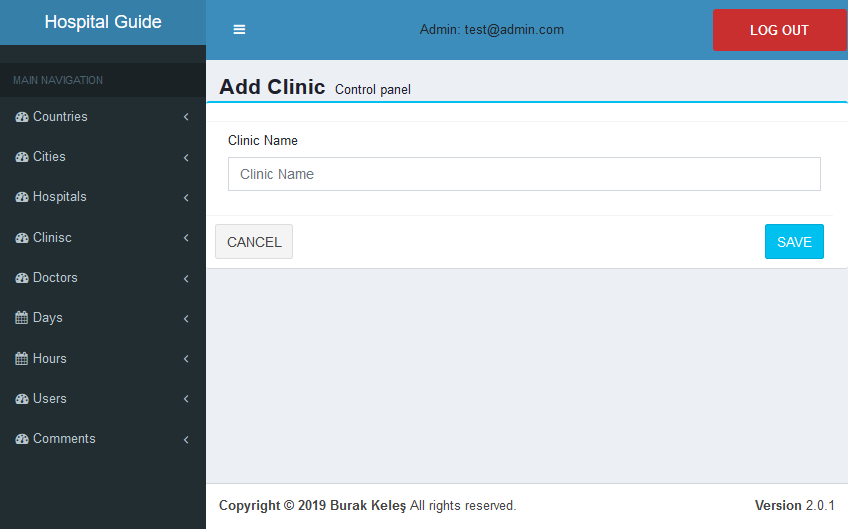


Figure - Add Clinic Page

## Clinics List

Added clinics are listed in cliniclist section. Admin can update clinic, delete clinic or list the doctors of related clinic.

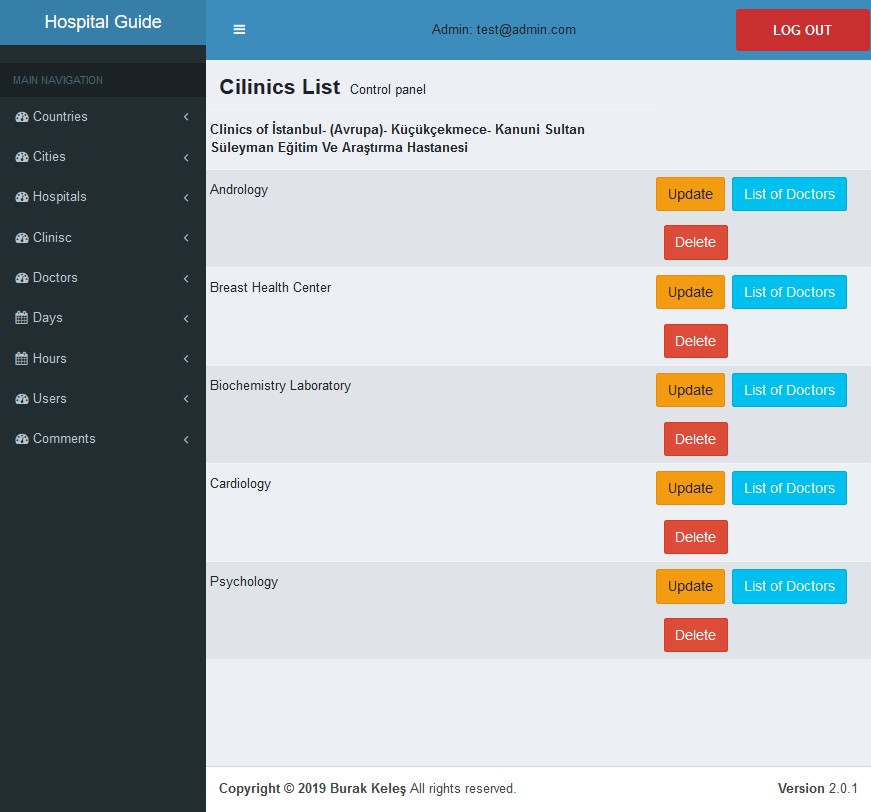


Figure - Clinic List Page

## Update Clinic

This is the page for the update information of clinic.

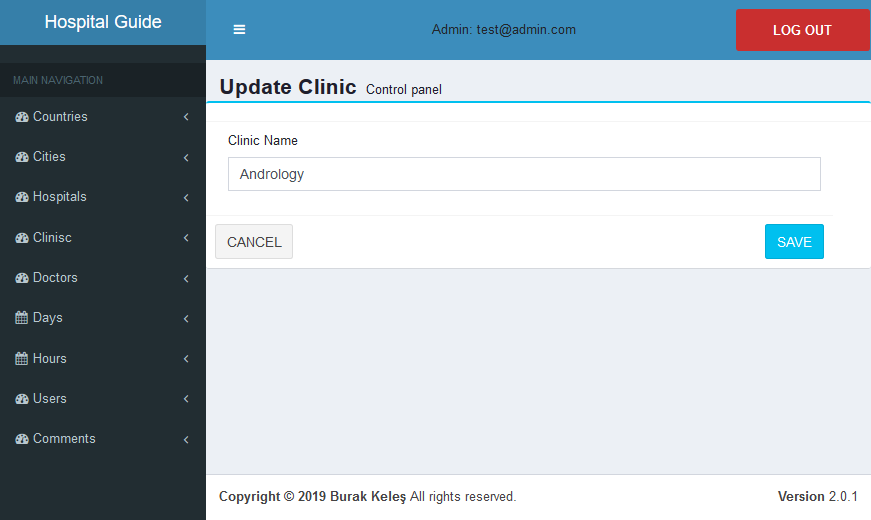


Figure - Update Clinic Page

## Add Doctor

Admin add doctor to system on adddoctorpage. Process started by select the add doctor from main navigation. After admin save the doctor. Schedule of doctor can be create.

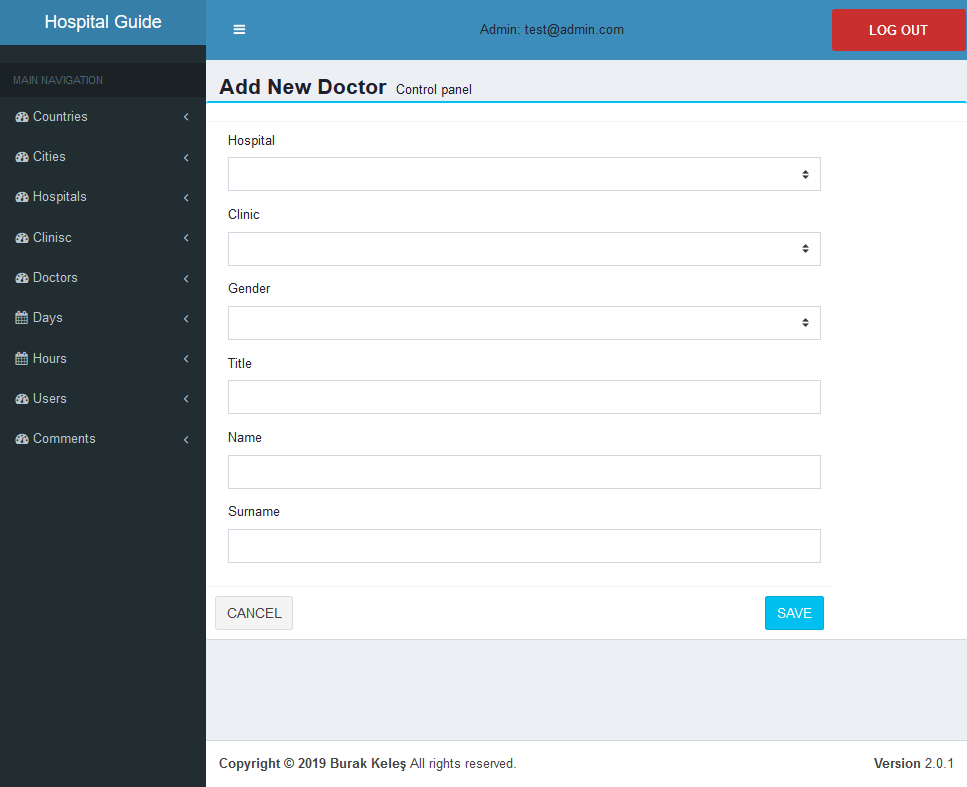


Figure - Add Doctor Page

## Doctors List

Added doctors are listed in doctorlist section. Admin can update doctor, delete doctor or list the days of related doctor.

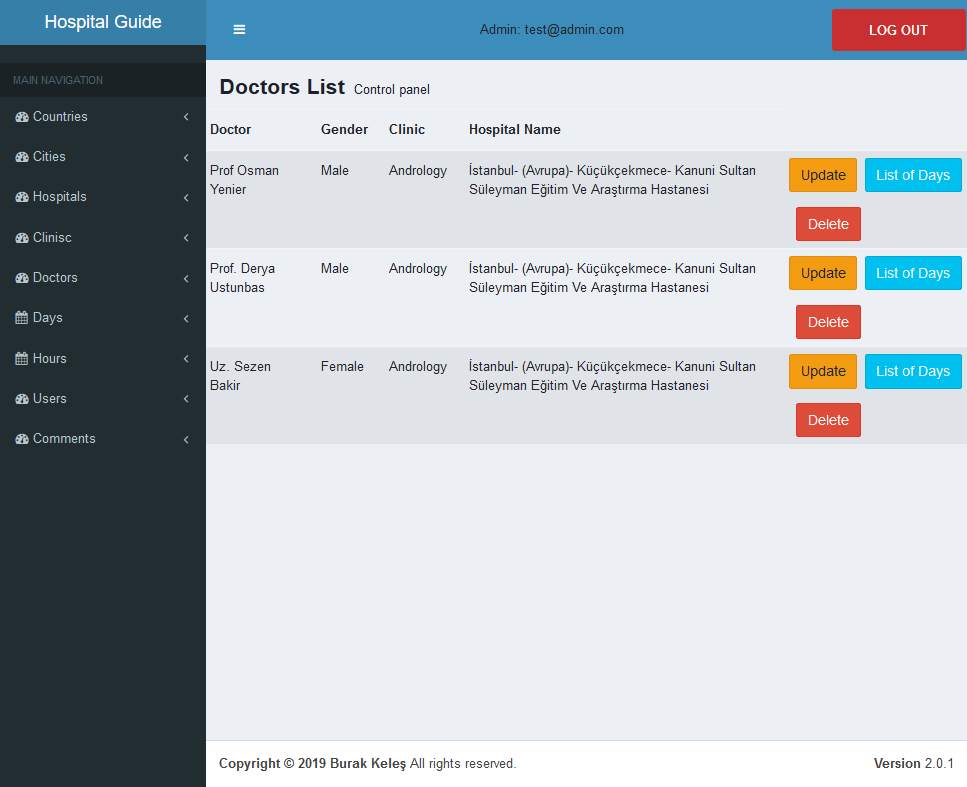


Figure - Doctor List Page

## Update Doctor

This is the page for the update information of doctor.

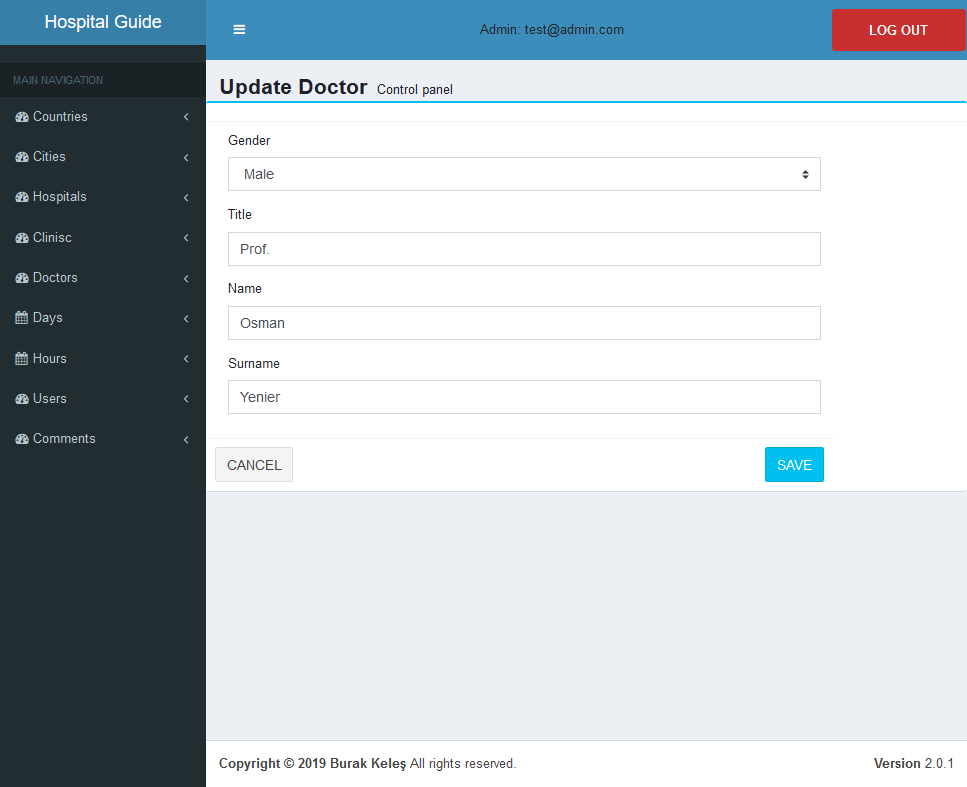


Figure - Update Doctor Page

## Add Day

Admin creates doctors Schedule day by select add day from main navigation. After page load selected day is assigned to doctor.

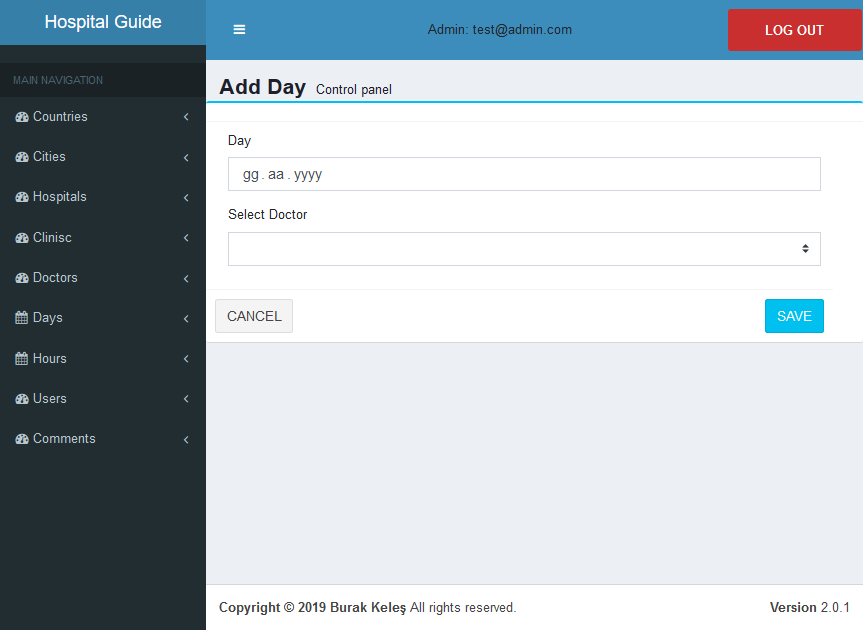


Figure - Add Day Page

## Days List

Added days are listed in daylist section. Admin can update day, delete day or list the hours of related day.

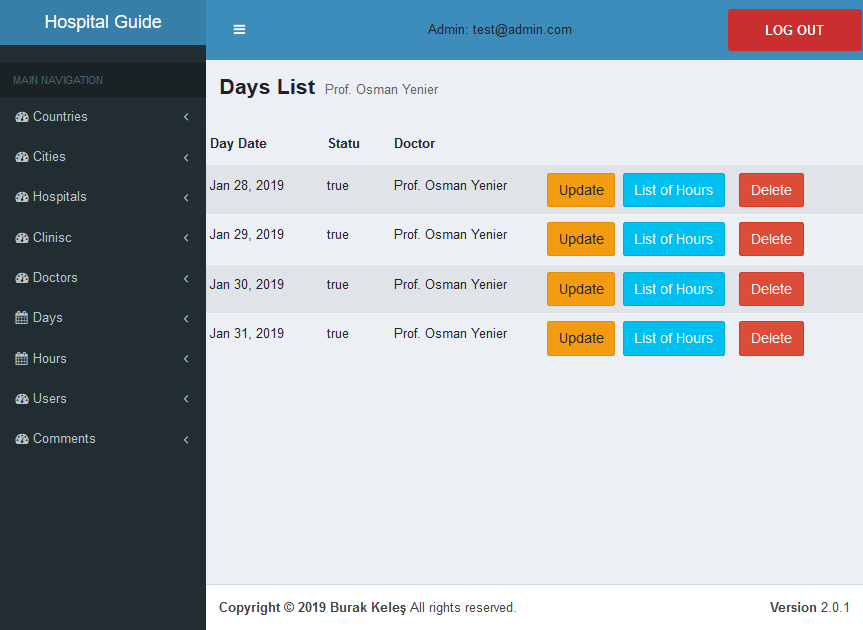


Figure - Day List Page

## Update Day

This is the page for the update information of day.

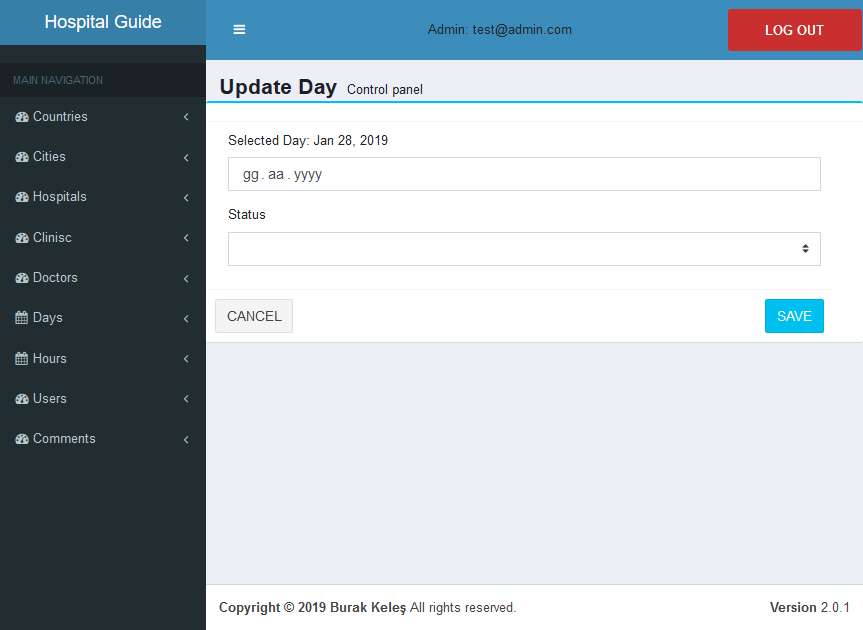


Figure - Update Day Page

## Add Hour

Admin can add hour to a doctors work Schedule. Add hour process selected from main navigation and addhourpage loaded. Admin Assign a work hour to a doctor with day.

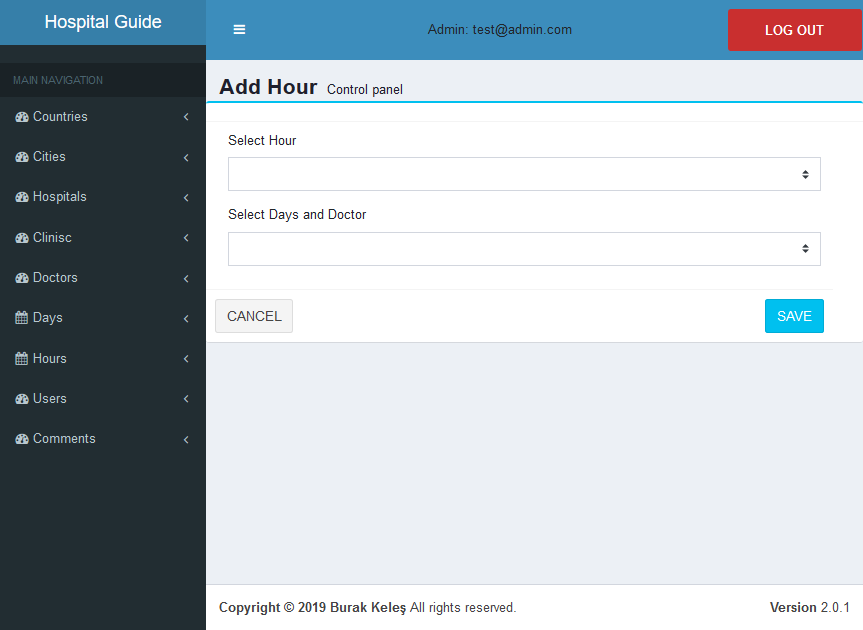


Figure - Add Hour Page

## Hours List

Added hours are listed in hourlist section. Admin can update hour or delete hour.

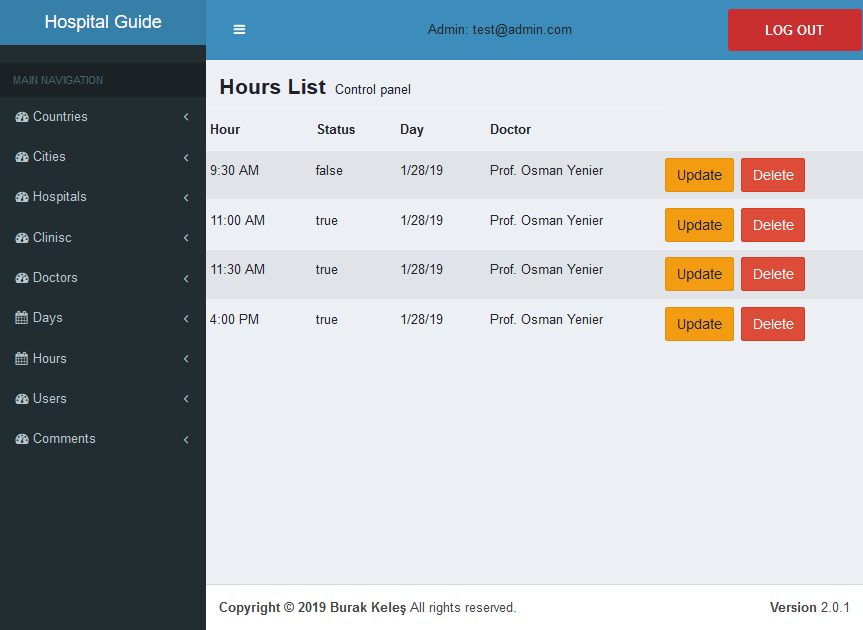


Figure - Hour List Page

## Update Hour

This is the page for the update information of hour.

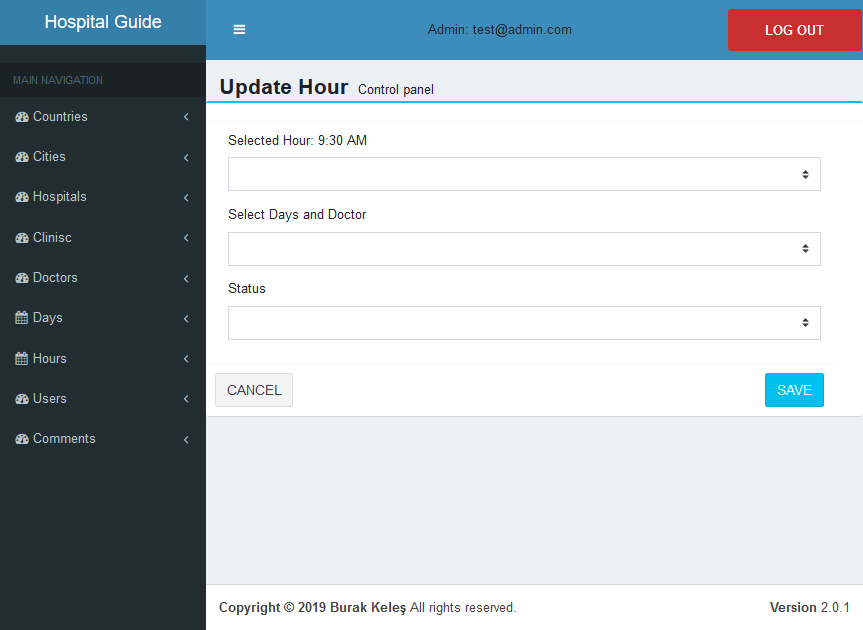


Figure - Update Hour Page

## Users List

Admin see the userlist by select users from main navigation. In this page admin can delete a user.

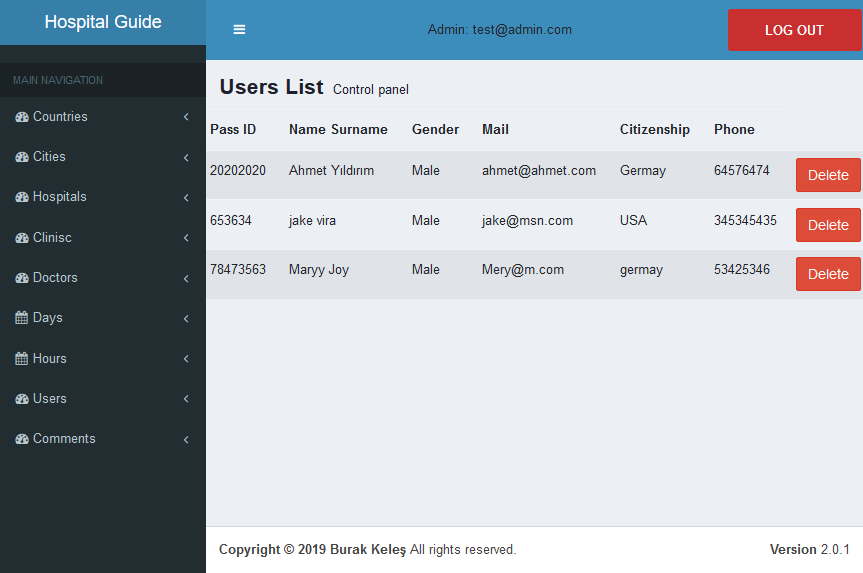


Figure - Users List Page

## Comments List

Admin can list comments by select commnets from main navigation. On that page admin can delete a comment that made for a hospital.

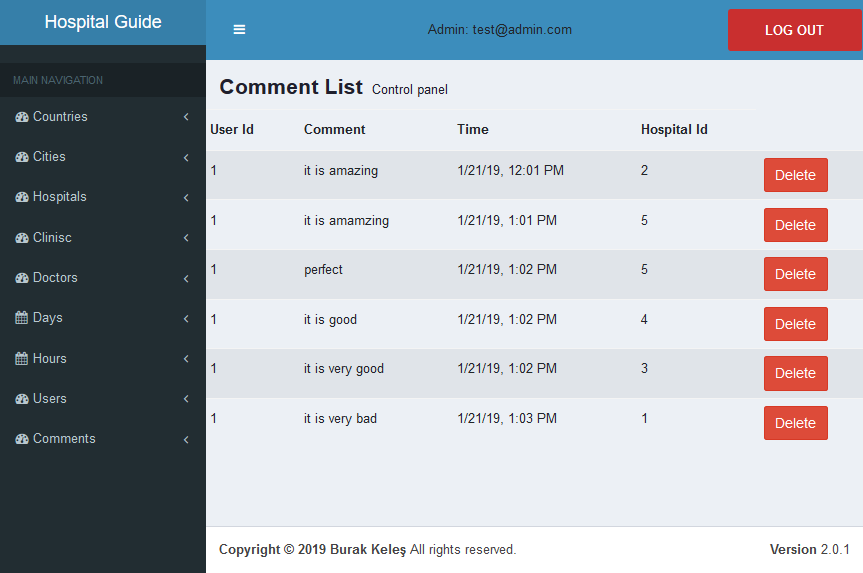


Figure - Comments List Page

**CHAPTER 6:**

**TESTING**

# TESTING

I tested ever webpage one by one. If any error occured after correction a new test applied. All the operation of user tested by condition test. If any user tries to do unusual thing application response according to it and does not crush. I applied lots of test after all the user operations finished. For admin operation all the removed from database and inserted and updated by admin functions programmability. Every condition has been tested sevarel times to see the results correctly. Using a test class is effective but i tried to learn everything on that project and minimize my own faults was very important to keep this progress up.

Software maintenance is the inevitable part of the development process and one that could give developers the most trouble. We’ve all been there, whether we left our code in someone else’s care, or we’ve inherited some legacy code.

This doesn’t necessarily need to be a bad thing and there are ways to improve our code and make it more maintainable. Unit testing plays a very important role in making software more maintainable.

Our intention in this post is to make an intro to unit testing of ASP.NET Core Web API application

## About Unit Testing in General

What is unit testing in the first place? It begins by defining what a „unit“ is and although this is not strictly defined, unit represents a unit of work – usually a single method in our code. We test these units individually, making sure that each of them is doing exactly that what it is written for. Nothing more, nothing less. What is important to understand is that we are not testing the behavior of the dependencies of that method. That is what the integration tests are for.

# REFERENCES

[1] https://en.wikipedia.org/wiki/ASP.NET\_MVC

[2] https://en.wikipedia.org/wiki/ASP.NET\_Core

[3] https://en.wikipedia.org/wiki/AngularJS

[4] https://stackoverflow.com/.../entity-framework-join-3-tables

[5] https://stackoverflow.com/questions/20101705/how-to-save-select-sql-query-results-in-an-array-in-c-sharp-asp-net

[6] https://stackoverflow.com/questions/20475223/create-table-in-database-using-entity-framework

[7] https://stackoverflow.com/questions/35763730/difference-between-constructor-and-ngoninit

[8] https://stackoverflow.com/questions/34024888/how-to-implement-rooting-angularjs-with-customize-url

[9] https://www.techiediaries.com/angular-components/

[10] https://stackoverflow.com/questions/36527605/how-to-style-child-components-from-parent-components-css-file/36527774

[11] https://getbootstrap.com/docs/3.3/getting-started/

[12] https://stackoverflow.com/questions/35265976/include-bootstrap-in-my-html-page

[13] https://stackoverflow.com/questions/648992/session-timeout-in-asp-net