SOFTWARE DESIGN DESCRIPTION

**MatchMaker**

**Department of Computer Engineering**

**Faculty of Engineering**

**Istanbul Okan University**

**Göktürk Ramazan DOĞAN**

**170212013**

**CONTENTS**

Table of Contents 2

1.INTRODUCTION 3

1.1. Title 3

1.2. Purpose of This Report 3

1.3. Scope of The Report 3

1.4. Abbreviations and synonyms used 3

2.DATA DESIGN 4

2.1. Data Objects and Resultant Data Structures 4

3.MODULE STRUCTURE 10

3.1. Module Map 10

3.2. LOGIN 11

3.3. IDENTIFY 12

3.4. ADDPLAYERS: 13

3.5. ADDFACILITIES: 14

3.6. ADDMATCHES 15

3.7. EDITPLAYERS 16

3.8. EDITFACILITIES 17

3.9. ADDTEAM 18

3.10. VOTE 19

3.11. DELETE 20

3.12. ERROR 21

4.INTERFACE DESIGN: 22

5.RECOMMENDATIONS 36

# **1.INTRODUCTION**

## **1.1. Project Name:**

MatchMaker.

## 

## **1.2. Purpose of This Report:**

This report contains information about the software design features and developability of the MatchMaker project.

## 

## **1.3. View of The Report:**

Data structure, module structure, administration panel admin interface and user interface designs are presented in this report. It must be used together with the requirements specification report.

## 

## **1.4. Abbreviations and synonyms used:**

**Admin:** Administrator.

**Info:** Information.

**Int:** User Interface.

**UserID:** User identification.

# 

# **2. DATA DESIGN**

## **1.1. Data Objects and Resultant Data Structures**

The entity-relationship diagram of the MatchMaker is depicted in the below figure. The Matchmaker system has two data objects. One of them is the administrator data object and one is the login data object. The login data object contains the username and password information. If the entered data matches the data in the database, the login operation is successful. If it does not match, the login will fail.

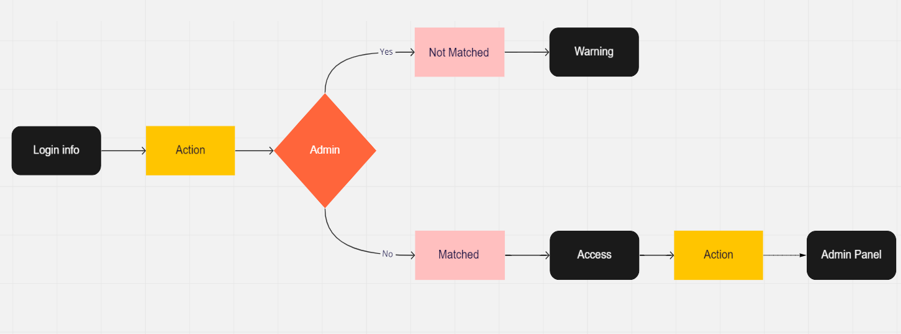


Figure 1 Login Event MatchMaker

Based on the E-R model, following tables are created. The fields of tables, the type nad primary key. All of them has created by admin manually from the database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | Type | **Primary Key / Foreign Key** | Required |
| **ayarlar** |  |  |  |
| ID | Int(11) | Primary Key |  |
| baslik | Varchar(160) |  | √ |
| anahtar | Varchar(255) |  | √ |
| aciklama | Varchar(255) |  | √ |
| telefon | Varchar(50) |  | √ |
| mail | Varchar(100) |  | √ |
| adres | Varchar(255) |  | √ |
| fax | Varchar(50) |  | √ |
| url | Varchar(120) |  | √ |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | Type | **Primary Key / Foreign Key** | Required |
| **kullanicilar** |  |  |  |
| ID | Int(9) | Primary key |  |
| adsoyad | Varchar(20) |  | √ |
| kullanici | Varchar(20) |  | √ |
| sifre | Varchar(20) |  | √ |
| mail | Varchar(30) |  | √ |
| tarih | Date |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | Type | **Primary Key / Foreign Key** | Required |
| **maclar** |  |  |  |
| ID | Int(11) | Primary key |  |
| tarih | Varchar(55) |  | √ |
| saat | Varchar(55) |  | √ |
| tesis | Varchar(120) |  | √ |
| takimakaleci | Varchar(500) |  |  |
| takimadefans1 | Varchar(500) |  |  |
| takimadefans2 | Varchar(500) |  |  |
| takimadefans3 | Varchar(500) |  |  |
| takimaortasaha1 | Varchar(500) |  |  |
| takimaortasaha2 | Varchar(500) |  |  |
| takimaforvet | Varchar(500) |  |  |
| takimbkaleci | Varchar(500) |  |  |
| takimbdefans1 | Varchar(500) |  |  |
| takimbdefans2 | Varchar(500) |  |  |
| takimbdefans3 | Varchar(500) |  |  |
| takimbortasaha1 | Varchar(500) |  |  |
| takimbortasaha2 | Varchar(500) |  |  |
| takimbforvet | Varchar(500) |  |  |
| takimakalecipuan | Int(11) |  |  |
| takimadefans1puan | Int(11) |  |  |
| takimadefans2puan | Int(11) |  |  |
| takimadefans3puan | Int(11) |  |  |
| takimaortasaha1puan | Int(11) |  |  |
| takimaortasaha2puan | Int(11) |  |  |
| takimaforvetpuan | Int(11) |  |  |
| takimbkalecipuan | Int(11) |  |  |
| takimbdefans1puan | Int(11) |  |  |
| takimbdefans2puan | Int(11) |  |  |
| takimbdefans3puan | Int(11) |  |  |
| takimbortasaha1puan | Int(11) |  |  |
| takimbortasaha2puan | Int(11) |  |  |
| takimbforvetpuan | Int(11) |  |  |
| takimaskor | Int(55) |  |  |
| takimbskor | Int(55) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | Type | **Primary Key / Foreign Key** | Required |
| **mevki** |  |  |  |
| ID | Int(11) | Primary key |  |
| mevki | Varchar(120) |  | √ |
| url | Varchar(500) |  | √ |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | Type | **Primary Key / Foreign Key** | Required |
| **mevki** |  |  |  |
| ID | Int(11) | Primary key |  |
| isim | Varchar(255) |  | √ |
| soyisim | Varchar(255) |  | √ |
| mevki | Varchar(255) |  | √ |
| telefonnumarasi | Varchar(255) |  | √ |
| ayak | Varchar(255) |  | √ |
| resim | Varchar(500) |  | √ |
| dogumtarihi | Varchar(255) |  | √ |
| mevkiid | Int(11) |  | √ |

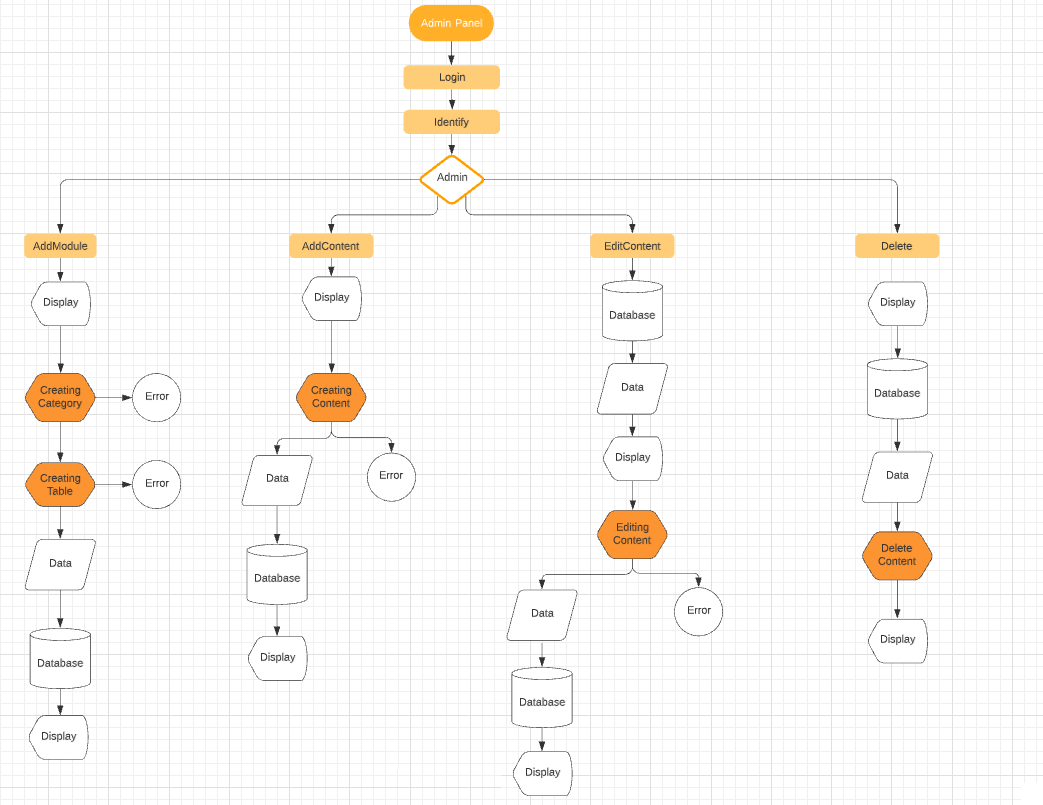
|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | Type | **Primary Key / Foreign Key** | Required |
| **puan** |  |  |  |
| ID | Int(11) | Primary key |  |
| mevki | Varchar(120) |  | √ |
| url | Varchar(500) |  | √ |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | Type | **Primary Key / Foreign Key** | Required |
| **tesisler** |  |  |  |
| ID | Int(11) | Primary key |  |
| tesisadi | Varchar(255) |  | √ |
| sahasayisi | Varchar(255) |  | √ |
| telefon | Varchar(30) |  | √ |
| sahatipi | Varchar(20) |  | √ |
| resim | Varchar(255) |  | √ |
| adres | Varchar(255) |  | √ |
| konum | Varchar(500) |  | √ |

# **3. MODULE STRUCTURE(FOR ADMIN PANEL)**

## 

## **3.1. Module Map:**



## **3.2. Login:**

Function : Program entry, gets name and password of the user and passes to Identify module .

Called by : Main program entry, called firstly after running the AdminPanel.

Calls : Identify module.

Inputs : Name and password information from user.

Outputs : Name and password information from user passes to Identify module.

Algorithm :

Display Login Page (see Int.1.) shown in part 5-interface design.

Get name and password of user as strings from login form at the page of Login.

Calls Identify module.

## 

## **3.3. Identify:**

Function : Gets name and password of the user (see Int.1.) shown in part 5-interface design. Checks their validity from Admin table from database.

Called by : Login Module.

Calls : Error Module, Admin Panel Homepage.

Inputs : Username and password information.

Outputs : If login is success Refreshes the page and goes to Main page of Admin Panel (see Int. 2.). If it is not, displays (see Int. 6-1-1. or Int. 6-1-2.) shown in part 5-interface design.

Algorithm :

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## **3.4. Addplayers:**

Function : This module saves the player information entered by the administrator on the relevant form page (see Int. 3-2.) in the 'oyuncular' table. shown in part 5-interface design.

Called by : Identify module.

Calls : Error Module.

Inputs : Information of the player to be created.

Outputs : If the add operation is successful displays (see Int. 6-2-1.) and switches to players-list (see Int. 3-1.). But if the add operation is not successful displays (see Int. 6-3.) shown in part 5-interface design.

Algorithm :

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## **3.5. Addfacilities:**

Function : This module by taking the information of the facility to be entered from the form on the page (see Int. 4-2.) by admin. records in the corresponding columns in the 'tesisler' table.

Called by : Identify module.

Calls : Editfacilities Module, Delete Module, Error Module.

Inputs : Information of the faicility to be created.

Outputs : If the save operation is successful display (see Int. 6-2-1.) and switches to facility-list page (see Int. 4-1.) in part-5. But if the save is not successful displays (see Int. 6-3.) in part-5.

Algorithm : Display Addfacility page (see Int. 4-2.) in part-5.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## **3.6. Addmatches:**

Function : This module saves the match data received by the admin from the form inputs on the match add page to the 'maclar' table.

Called by : Identify module.

Calls : Editmatch Module, Delete Match Module, Error Module, Addteam Module, Vote Module.

Inputs : Information of the match to be created.

Outputs : If the save operation is successful display (see Int. 6-2-1.) and switches to match-list page (see Int. 5-1.) in part-5. But if the save is not successful displays (see Int. 6-3.) in part-5.

Algorithm : Display Addmatches page (see Int. 5-2.) in part-5.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## **3.7. Editplayers:**

Function : It allows editing the data of the players added to the related table with the Addplayers algorithm.

Called by : Identify Module, Addplayers Module.

Calls : Error Module, Delete Module.

Inputs : Replacing the data of the relevant player with the data to be updated.

Outputs : It displays (see Int. 6-2-1.) or (see Int. 6-3.) in part-5.

Algorithm : After the editing process is finished displays (see Int. 6-2-1.) if the proccess is success. But if it is not displays (see Int. 6-3.) in part-5.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## **3.8. Editfacilities:**

Function : It allows editing the data of the facility added to the related table with the Addfacilities algorithm.

Called by : Identify Module, Addfacilities Module.

Calls : Error Module, Delete Module.

Inputs : Replacing the data of the relevant facility with the data to be updated.

Outputs : It displays (see Int. 6-2-1.) or (see Int. 6-3.) in part-5.

Algorithm : After the editing process is finished displays (see Int. 6-2-1.) if the proccess is success. But if it is not displays (see Int. 6-3.) in part-5.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## **3.9. Addteam:**

Function : This module gives the authority to add the players as teams to the positions where they will be assigned by the manager and adds this data to the columns of the relevant match in the 'maclar' table.

Called by : Addmatches Module.

Calls : Error Module, Delete Module, Vote Module.

Inputs : Selection of players from the relevant select inputs by the administrator

Outputs : It displays (see Int. 6-2-1.) or (see Int. 6-3.) in part-5.

Algorithm : After the adding process is finished displays (see Int. 6-2-1.) if the proccess is success. But if it is not displays (see Int. 6-3.) in part-5.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## **3.10. Vote:**

Function : This module records the score entered by the administrator and the performance information of the players out of 5 in the relevant columns in the 'maclar' table.

Called by : Addmatches Module, Addteam Module.

Calls : Error Module, Delete Module.

Inputs : The score produced by the teams entered by the manager and the performance score shown by the players.

Outputs : It displays (see Int. 6-2-1.) or (see Int. 6-3.) in part-5.

Algorithm : After the voting process is finished displays (see Int. 6-2-1.) if the proccess is success. But if it is not displays (see Int. 6-3.) in part-5.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## 

## **3.11. Delete:**

Function : Deletes all data of the related content and itself from the related table.

Called by : Addplayers Module, Addfacilities Module, Addmatches Module.

Calls : None

Input : Admin click.

Outputs : None

Algorithm :

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## **3.12. Error:**

Function : Displays a warning when algorithm return false value.

Called by : Identify module, Addplayers Module, Addfacilities Module, Addmatches Module, Editplayers Module, Editfacilities Module, Addteam Module, Vote Module.

Calls : none

Input : Admin click.

Outputs : Error Messages which created by admin.

Algorithm : (see Int.6-1-1.),(see Int.6-1-2.),(see Int.6-1-3.).

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

# **5.INTERFACE DESIGN**

**ADMIN PANEL**

Int. 1. Login Page.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 2. Default HomePage.

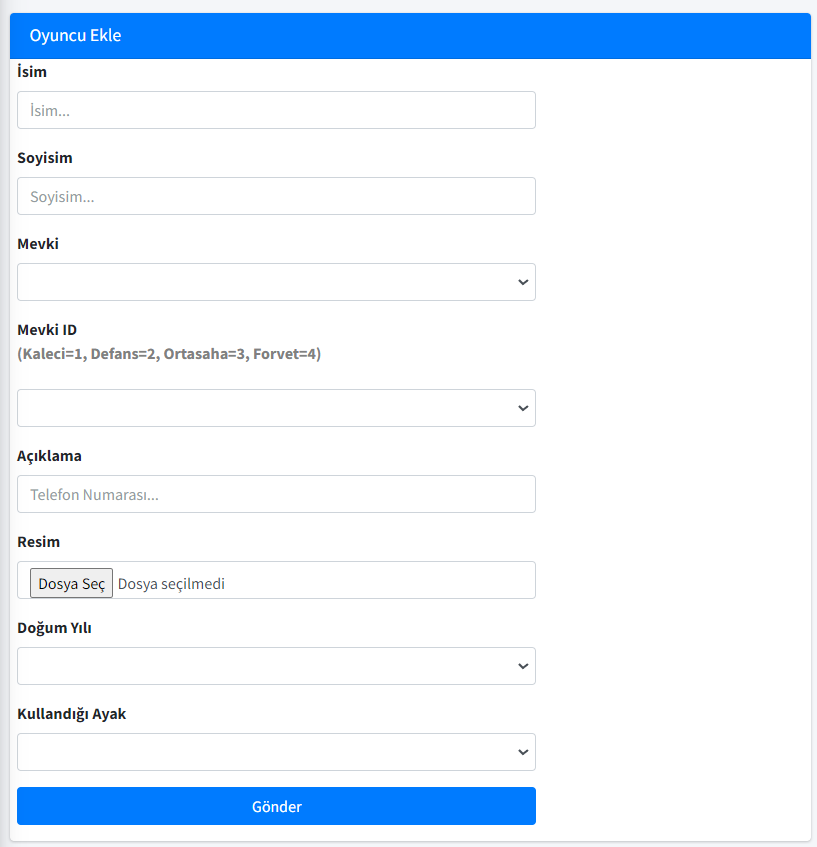


Int. 3-1. PlayersList Page.

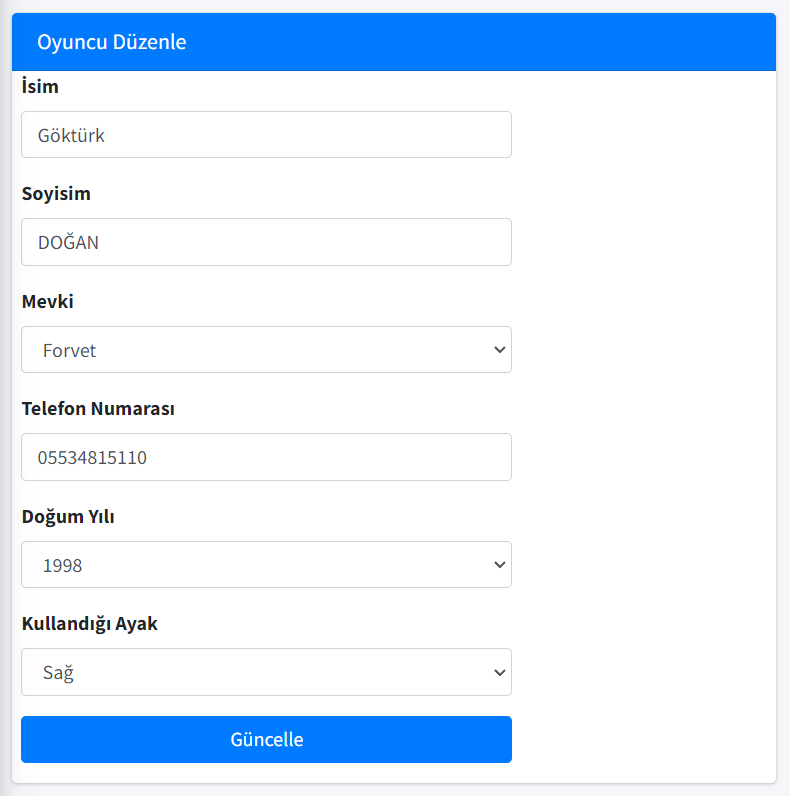
metin, iç mekan, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 3-2. Addplayers Page.



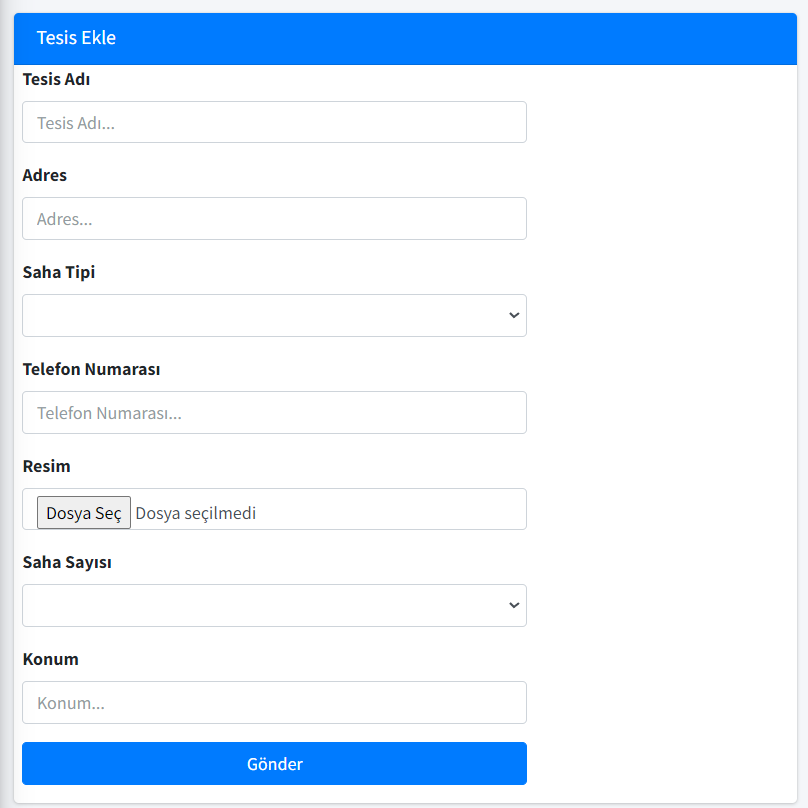
Int. 3-3. Player-Edit Page.



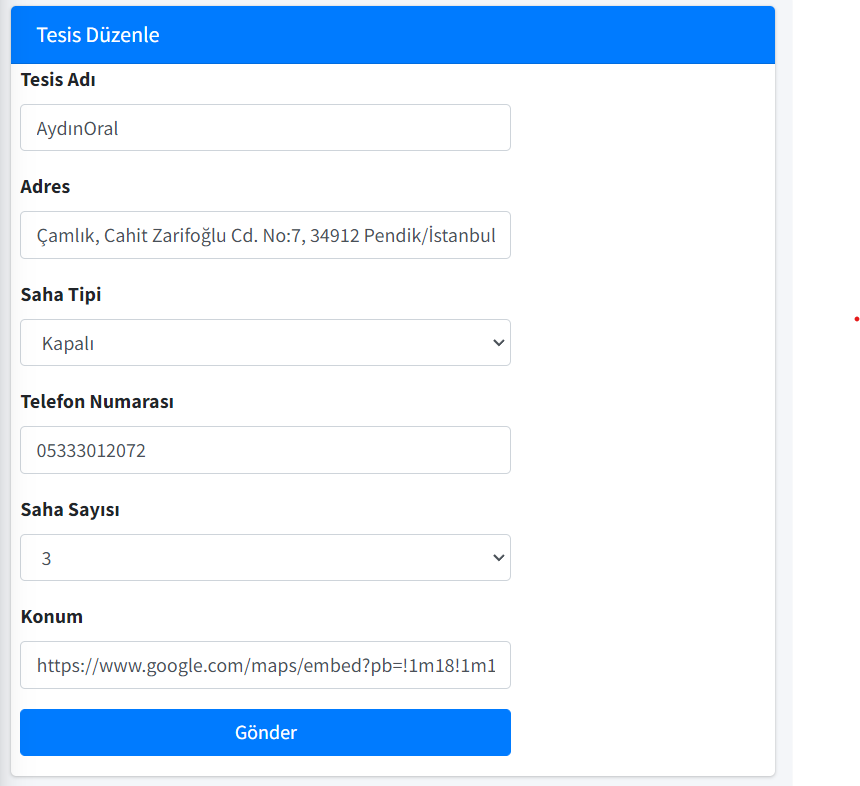
Int. 4-1. Facilities-List Page.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 4-2. Facilityadd Page.

Int. 4-3. Facility-edit Page.

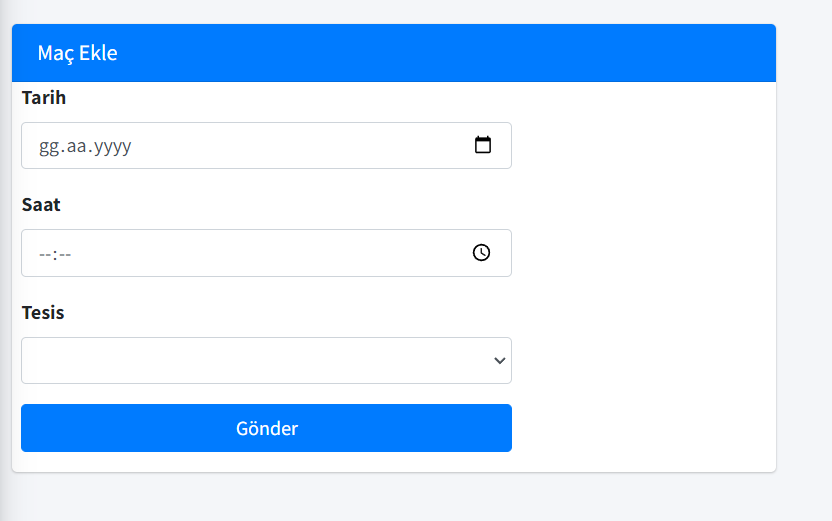


Int. 5-1. Matches-list Page.

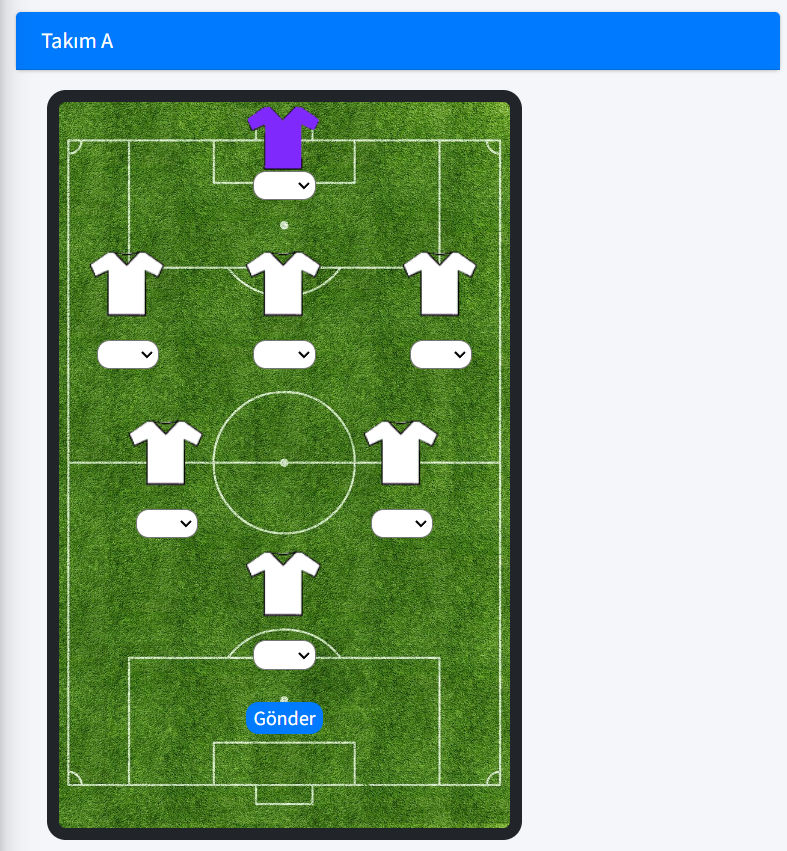
metin, ekran görüntüsü, iç mekan içeren bir resim

Açıklama otomatik olarak oluşturuldu

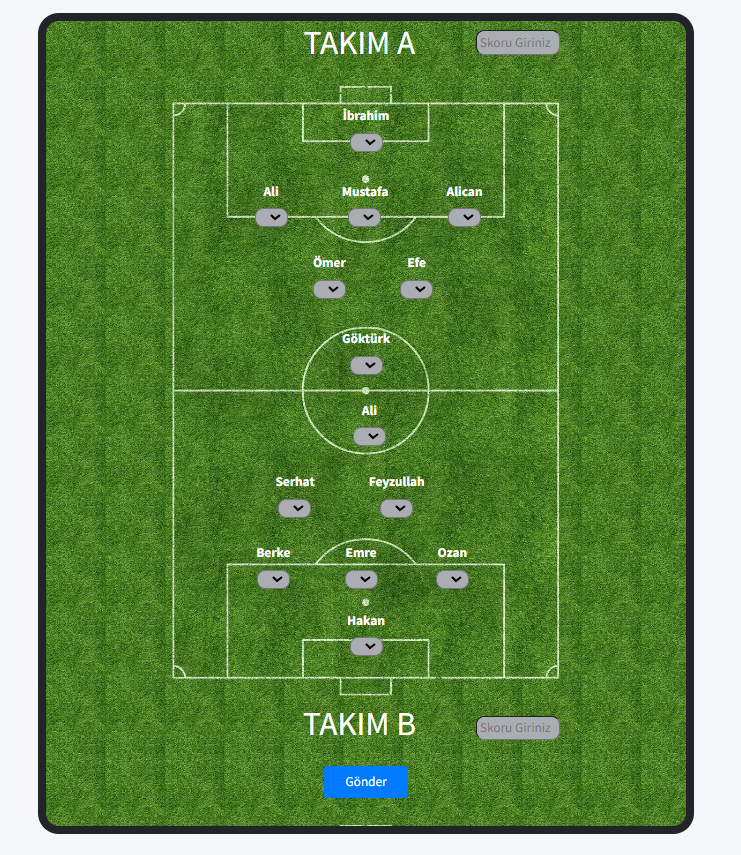
Int. 5-2. Addmatch Page.



Int. 5-3. Addteam Page.



Int. 5-4. Vote Page.



Int. 6-1-1. Error.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 6-1-2. Error.

metin içeren bir resim

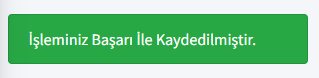
Açıklama otomatik olarak oluşturuldu

Int. 6-1-3. Error.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 6-2-1. Success.



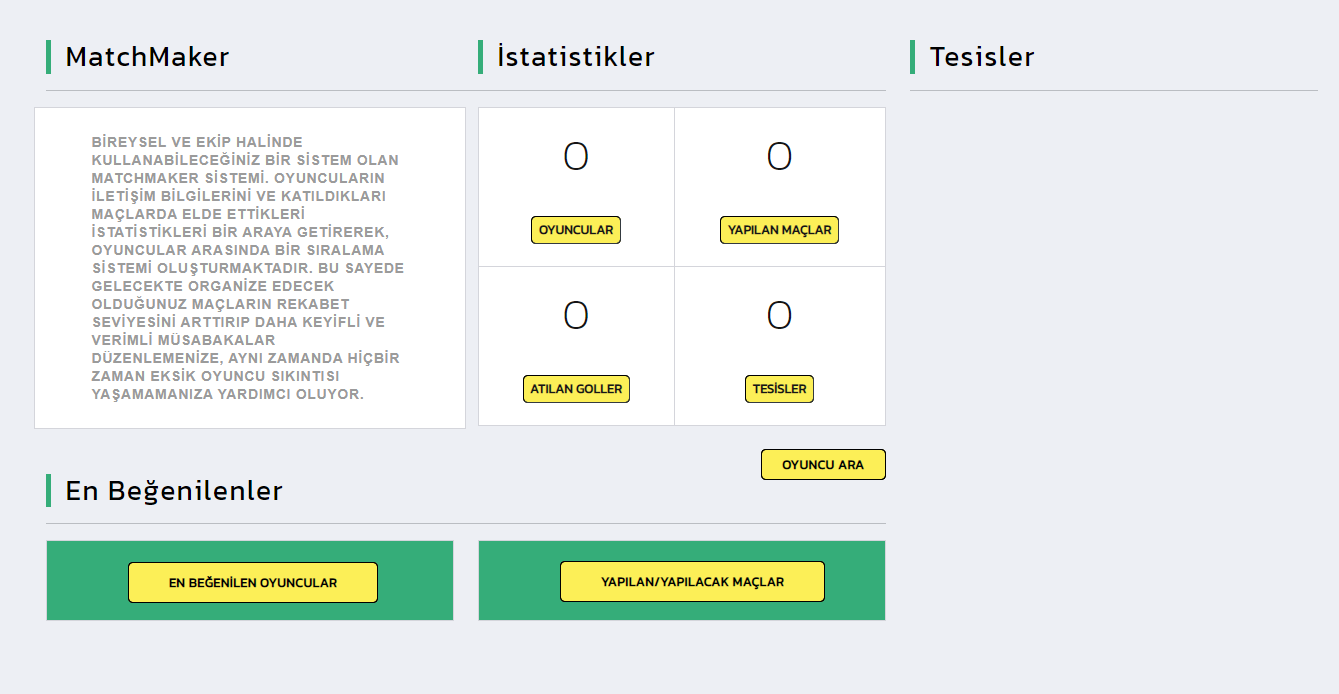
Int. 7. Menu.

metin, elektronik eşyalar, ekran görüntüsü içeren bir resim

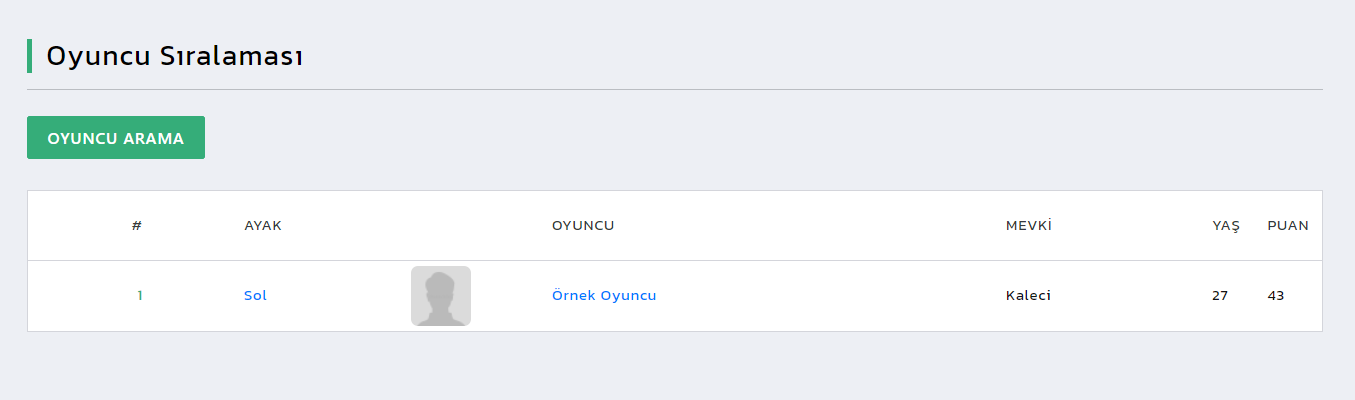
Açıklama otomatik olarak oluşturuldu

**USER INTERFACE**

Int. 8. HomePage.



Int. 9. Players Page.



Int. 9-1. Player Detail Page.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 10-1 Facilities Page.

metin, ekran görüntüsü, iç mekan içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 10-2 Facility-Detail Page.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 11-1 Matches Page.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 11-2 Match-Detail Page.



Int. 12 Menu.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Int. 13 Footer.

metin, ekran görüntüsü, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu

# **6. RECOMMENDATIONS**

The project requires a PhpMyadmin database structure to be formed manually for the software.

Use of Php prgoramming language may be beneficial for the ease of implementation.