<ONLINE TICKET SYSTEM>

Object Design

<1.1>

<02.01.2017>

<HASAN MURAT AKPOLAT

EMRE YILMAZ

ERTAN GÖKYILDIRIM

MELİS KILIÇ>

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc436772639)

[1.1. Object Design Trade-offs 1](#_Toc436772640)

[1.2. Interface Documentation Guidelines 2](#_Toc436772641)

[1.3. Definitions, Acronyms, and Abbreviations 3](#_Toc436772642)

[1.4. References 4](#_Toc436772643)

[2. Packages 4](#_Toc436772644)

[3. Class Interfaces 5](#_Toc436772645)

# Introduction

This Object Design Document defines the design of an online ticket system.The goal of this project is provide a reliable service for people and its main purpose simplify buying an online ticket.Also it provides help for clients for use online ticket system for selling their events in online.

The Online Ticket system has trade off which leads to some costs like as: development, understand ability and security.

## Object Design Trade-offs

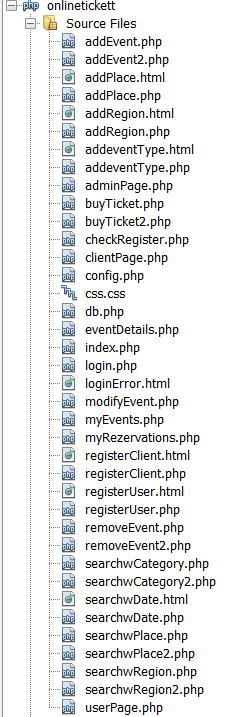
Object Design trade-offs leads to some costs like as: development,understand ability and security.

Online Ticket System users, clients and admin use functions of the system for using it.That causes development cost because all that functions takes time for design and implement.

When developers implement the code of the system it needs to be understand ability because when they need to change something missing or wrong they can’t lose time for found where that excuse.They need to use commend lines for avoid that time lose.

In Online Ticket System all users needs to be registered.Unregistered users cannot use the system.Also clients when they registered they need to wait admin approval.All of guests need an username and password.This situation requires an security cost.

## Interface Documentation Guidelines



|  |  |  |
| --- | --- | --- |
| Fully qualified class name | Unqualified name | Remarks |
| addEvent | PHP File | Client’s adding event class |
| addPlace | PHP File | Admin’s adding place class |
| addRegion | PHP File | Admin’s adding region class |
| addeventType | PHP File | Admin’s adding type class |
| adminPage | PHP File | Admin Panel |
| buyTicket | PHP File | User’s buying ticket class |
| checkRegister | PHP File | Checking registers on system |
| config | PHP File | Db connection class |
| clientPage | PHP File | Client Panel |
| db | PHP File | Db info class |
| eventDetails | PHP File | Last screen before buying a ticket |
| index | PHP File | Form of login screen |
| login | PHP File | Accessing on the system class |
| modifyEvent | PHP File | Update events on system class |
| myEvents | PHP File | Client’s own events class |
| myRezervations | PHP File | Bought tickets class for users |
| registerClient | PHP File | Register for Clients class |
| registerUser | PHP File | Register for Users class |
| removeEvent | PHP File | Removing an event class |
| searchwCategory | PHP File | Search with Category class |
| searchwPlace | PHP File | Search with Place class |
| searchwRegion | PHP File | Search with Region class |
| searchwDate | PHP File | Search with Date class |
| userPage | PHP File | User Panel |

|  |  |  |
| --- | --- | --- |
| Fully qualified class name | Unqualified name | Remarks |
| addPlace | HTML File | Adding Place form |
| addRegion | HTML File | Adding Region form |
| addeventType | HTML File | Adding event type form |
| registerClient | HTML File | Client register form |
| registerUser | HTML File | User register form |
| loginError | HTML File | Login error screen |
| searchwDate | HTML File | Search with Date form |

## Definitions, Acronyms, and Abbreviations

- Database: Database is a collection of information that is organized so that it can easily be

Accessed

-MYSQL: My Structured Query Language

- ID : Identification

- Login: to get access to an operating system or application, usually in a remote computer

- ODD : Object Design Document

- UI : User Interface

- PHP: Hypertext Preprocessor

- HTML : HyperText Markup Interface

-CSS: Cascading Style Sheets

-Session: Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

- Member: a person who use online ticket system

- Exception: Represents errors that occur during application execution.

- Error: The condition of having incorrect or false knowledge.

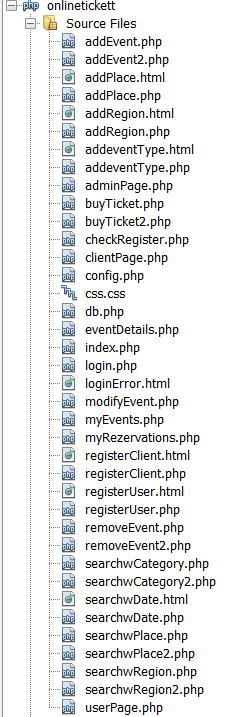
## References

<http://blog.slickedit.com/2007/05/how-to-write-an-effective-design-document/>

<http://ee.hawaii.edu/~tep/EE467/BDLecs/html/odd.htm>

# Packages

We used one main package on Online Ticket System.



# Class Interfaces

In our online ticket system, there are no methods.We used html forms and backround php classes. These forms and classes connected with Redirect() functions and window.location() alerts. All functions occur in main php classes with sql queries.

2 php classes for accessing the system shown below;

**Index.php**

**/\*<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Online Ticket System</title>**

**<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">**

**<link rel="stylesheet" type="text/css" href="css.css" />**

**</head>**

**<body id="top">**

**<div class="wrapper row1">**

**<div id="header" class="clear">**

**<h1><a href="index.php">Online Ticket System</a></h1>**

**<p>Işık Üniversitesi</p>**

**<br> </br>**

**</div>**

**</div>**

**<div class="row5" id="footer">**

**<ul>**

**<br>**

**<a href="index.php"> <font size="3"> Home</font> </a> &emsp;**

**<a href="registerClient.html"><font size="3">Client Register</font></a> &emsp;**

**<a href="registerUser.html"> <font size="3">User Register</font></a>&emsp;**

**</br>**

**</ul>**

**</div>**

**<?php**

**?>**

**<div class="wrapper row4 ">**

**<div id="footer footbox" class="clear" >**

**<br> </br><br> </br>**

**<form method = "post" action = "login.php">**

**<table>**

**<tr>**

**<td>**

**<label><font size="3" color="#000000">Username :</font></label>**

**</td>**

**<td>**

**<input type="text" size="49" name="user\_namee" >**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<label><font size="3" color="#000000">Password : </font></label>**

**</td>**

**<td>**

**<input type="password" size="50" name="passwordd" >**

**</td>**

**</tr>**

**<tr>**

**<td></td>**

**<td>**

**<input type = "submit" name = "Login" value = "Login" style="width: 155px;"/>**

**</td>**

**</tr>**

**</table>**

**</form>**

**<br></br><br> </br><br></br><br>**

**<br> </br><br> </br><br> </br>**

**</div>**

**</div>**

**<div class="wrapper ">**

**<div id="footer" class="clear">**

**<h2><br />Isik University</h2>**

**<address>**

**Sile Kampusu/Mesrutiyet Koyu<br />**

**Sile/Istanbul<br />**

**34980**

**</address>**

**<ul>**

**<li><strong>Tel:</strong> 0 216 712 14 60</li>**

**<li><strong>Fax:</strong> 0 216 710 28 73</li>**

**<li class="last"><strong>Email:</strong> <a href="mailto:info@isikun.edu.tr">info@isikun.edu.tr</a></li>**

**</ul>**

**</div>**

**</div>**

**</body>**

**</html>**

**LOGIN.PHP**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">**

**<title>Apply Page</title>**

**</head>**

**<body>**

**<?php**

**function Redirect($url, $permanent = false)**

**{**

**if (headers\_sent() === false)**

**{**

**header('Location: ' . $url, true, ($permanent === true) ? 301 : 302);**

**}**

**exit();**

**}**

**include 'config.php';**

**try {**

**$baglanti=new PDO("mysql:host=localhost;dbname=onlineticket","root","");**

**//echo "Mysql Bağlantısı Başarıyla Sağlandı. <br />";**

**$baglanti->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION); // <== add this line**

**$user\_name=$\_POST['user\_namee'];**

**$password=$\_POST['passwordd'];**

**$sql = "SELECT member\_type FROM member WHERE user\_name = '$user\_name' and password='$password' ";**

**session\_start();**

**$result = $baglanti->prepare($sql);**

**$result->execute();**

**$number\_of\_rows = $result->fetchColumn(); //bu blokta username password kontrol ediyoruz**

**$sql2 = "SELECT status FROM member WHERE user\_name = '$user\_name' ";**

**$result = $baglanti->prepare($sql2);**

**$result->execute();**

**$mid = $result->fetchColumn();**

**if ($number\_of\_rows!= NULL) {**

**if($number\_of\_rows == "0") {**

**$\_SESSION["user\_name"] = $user\_name;**

**Redirect('adminPage.php', false);**

**}**

**else if ( $number\_of\_rows == "1" && $mid == "Accepted" ) {**

**$\_SESSION["user\_name"] = $user\_name;**

**Redirect('clientPage.php', false);**

**}**

**else if ($number\_of\_rows == "2" ) {**

**echo "güzel";**

**$\_SESSION["user\_name"] = $user\_name;**

**Redirect('userPage.php', false);**

**}**

**else if($mid == "Waiting"){**

**echo "<script>alert('Onay bekliyorsunuz'); window.location = './index.php';</script>";**

**}**

**}**

**else{**

**Redirect('loginError.html', false);**

**}**

**}**

**catch(PDOException $e)**

**{**

**echo $e->getMessage();**

**}**

**?>**

**</body>**

**</html>**