<Online Ticket System>

System Design

<1.0>

<11.12.2016>

HASAN MURAT AKPOLAT

ERTAN GÖKYILDIRIM

EMRE YILMAZ

MELİS KILIÇ

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc433996772)

[1.1. Purpose of the System 1](#_Toc433996773)

[1.2. Design Goals 1](#_Toc433996774)

[1.3. Definitions, Acronyms, and Abbreviations 1](#_Toc433996775)

[1.4. References 2](#_Toc433996776)

[2. Current Software Architecture 2](#_Toc433996777)

[3. Proposed Software Architecture 2](#_Toc433996778)

[3.1. Overview 2](#_Toc433996779)

[3.2. System Decomposition 3](#_Toc433996780)

[3.3. Hardware Software Mapping 4](#_Toc433996781)

[3.4. Persistent Data Management 4](#_Toc433996782)

[3.5. Access Control and Security 5](#_Toc433996783)

[3.6. Global Software Control 5](#_Toc433996784)

[3.7. Boundary Conditions 5](#_Toc433996785)

[4. Subsystem Services 6](#_Toc433996786)

[5. References 7](#_Toc433996787)

SYSTEM DESIGN DOCUMENT[]

# Introduction

## Purpose of the System

Online Ticket Systemprovides an effective and online way to users to purchase ticket. Users do not need to go to ticket distributor by using this system. They can buy any ticket by using Online Ticket System. Moreover, Online Ticket System has included some services to clients, and admins. Clients can offer new events to customers. Clients add events ,time period of events and event type to these events. Admin add new places, regions and event types to the system for clients. Finally, purpose of the system is to provide efficient way to customers to purchase ticket, and purpose of the system is to provide efficient way to clients to add new events and purpose of the system is to provide efficient way to admin to manage system.

## Design Goals

Online Ticket System is anweb-based ticket platform for users and clients. Therefore, it should provide some constraints, functional and non-functional requirements. For example, on users’ side, Online Ticket System design should provide that customers can see all events in any region, customers can select any date and then they can search all events by using this date. In addition, our important goal is to purchase ticket to users; this is main and important goal of our design. Also, our design should provide that clients can add new events to the system and they can add event type of an event, price and time period to these events by using Online Ticket System. Moreover, the other goals of our design Online TicketSystem should besafety, should accept upgradesand should be implemented on Netbeansplatform,In addition, “Php”should be used as the programming language. For Database the system should use “MySQL”. User Interface should be web-based.

## Definitions, Acronyms, and Abbreviations

RAD: Requirement Analysis Design

SDD: System Design Document

HTTP: The Hypertext Transfer Protocol

UI: User Interface

PHP:Hypertext Preprocessor

MYSQL:My Structured Query Language

## References

[**http://blog.slickedit.com/2007/05/how-to-write-an-effective-design-document/**](http://blog.slickedit.com/2007/05/how-to-write-an-effective-design-document/)

[**https://en.wikipedia.org/wiki/Software\_design\_description**](https://en.wikipedia.org/wiki/Software_design_description)

# Current Software Architecture

Nowadays, Web-based and mobile ticket applications are very popular. In Turkey, “biletix.com” is one of the most popular and largest online ticket system. With this kind of ticket systems, the user selects the region that where s/he lives and selects the best event for their purpose or their budget.

In our Online Ticket System, Users can search for events for free with signing in. Also Users selectan event to buy a ticket; the system makes sure that the user is registered to the system. If not, the user fills the needed forms to register the system. After the registration, user can but his/her tickets easily and s/he can make the payment with the system. Also, users can check their own past/future events with their informations. A Client is a member too, who can register their events with their information. Clients can also change the information and the pricing of the event. Also The other functions of the client are add time period of event,add category to event and add seats of the event. In addition, admin can approve or reject the client registrations and also Admin can add new place, region, and event type into the system.In addition Admin can modify the event information and cancelation of events. Also Admin can update the places which are found into database.

Current system is being made by using PHP and HTML. Also, for database creation, MySQL Workbench is used. In all current online ticket systems, flow of events works in this way. However, details about the system can be vary from one to another such as which architecture the system use or is the system supported by any hardware equipment etc.

# Proposed Software Architecture

The main purpose of the system is to ensure simplicity of operations. Users should make a reservation easily without any problems.

The system inferface is designed as quite simple, target-driven and easy to use to fulfill our main purpose. Our proposed model of Online Ticket System is MVC. Model subsystems maintain domain knowledge, View subsystems display it to the user, and Controller subsystems manage the sequence of interactions with the user. We used MVC model because, in our system, entity objects and data will be in Model (java classes-entities), and Controller (Faces Config in our project) can be called bridge. Controller provides communication and interaction between Model and View. View (Bean in our project) can be call interface between users and our system, thus it has boundary objects.

## Overview

In our design, the system has a lot of subsystems. These are: Region Interface, User Interface, Seat Interface, Category Interface, EventType Interface, Event Interface, Place Interface, Login Interface, Profile Information are subsystems for the Presentation Layer. We use “MySQL”. for data access layer.

**User Interface:** provides services for users (user, client and admin) for common interfaces, It has Login Form, Password Change Form and etc.

**Region Interface**: provides services for users to display all regions and to select this region from box. And also provides services for admin to add new region into system.

**Category Interface:** provides services for users to display all categories and to select this category.

**Seat Interface:** provides services for users to display all seats and to select this seat. And also provides services for client to add new seat into any category.

**EventType Interface:** provides services for users to display all types and to select this type (music, sport). And also provides services for admin to add new type to the system. And it provides services for client to add event type into any event.

**Event Interface:** provides services for users to select event whatever they want. And also it provides services for client to add new event into the system.

**Place Interface:** provides services for admin to add new place into the system. And also provides services for client to add place into the any event.

**Login Subsystem:** provides services for users (user, client and admin) to login.

**Update Profile Subsystem:** provides services for user, client, and admin to update their personal information. In addition, the subsystem also provides the service of changing password for users, admin and client.

## System Decomposition

In Online Ticket system decomposition, the subsystems are: User Interface, Region Interface, Category Interface, Seat Interface, EventType Interface, Event Interface, Place Interface, Login Interface, Profile Information and Data Access Subsystem. User Interface contains Login Form, Profile Update Form, Change Password Form All Events List Form and etc. Region Interface contains All Regions List Form. Category Interface contains All Category List Form. In that there are events. A event contains a lot of categories. Seat Interface contains All Seat List Form. EventType Interface contains All Type List Form. Event Interface contains All Event List Form. Place Interface contains All Place List Form. Login Subsystem provides services for registered people (user, client, and admin) to login, it contains Login Control object. Profile Info Subsystem provides services for users (user, client) to update their information and to change password, it contains Update Profile Control object to update and Change Password Control object to change password. Data Access Subsystem; contains all our persistent objects.

## Hardware Software Mapping

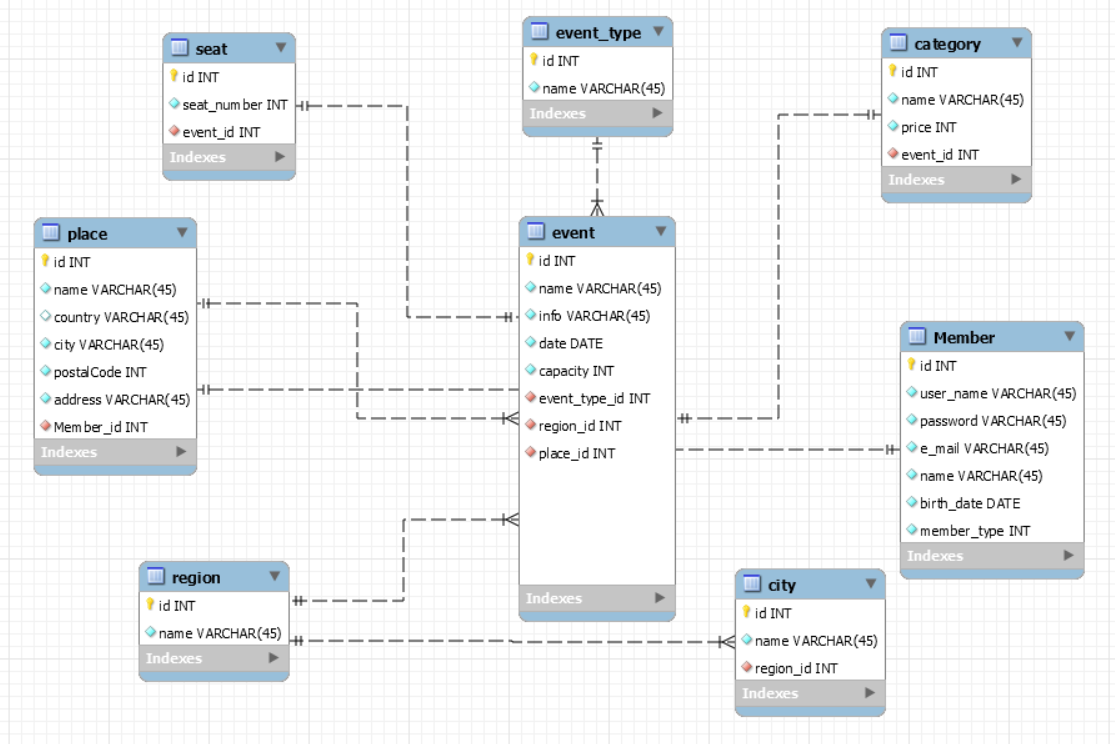
To make it easier for us to work, we decided to use nodes in Online Ticket project. We created three nodes. These are; Business layer node, data access layer node and presentation layer node. We make sure that the data sent in the presentation layer is understandable by the other computer. Presentation Layer will run codes; contains Html and CSS and then it offers web browser like as Mozilla Firefox and Google chrome. Data Access layer will run on PDO in Mysql Workbench. We used a lot of systems while creating sub-builds and because of deployment our system on different nodes. This nodes need to communicate with each other. We communicate with MVC (Model-View-Controller). The system will be overloaded during registration and many users will try to access the tickets. Data access layer node and business layer node will receive ticket requests from users.

## Persistent Data Management

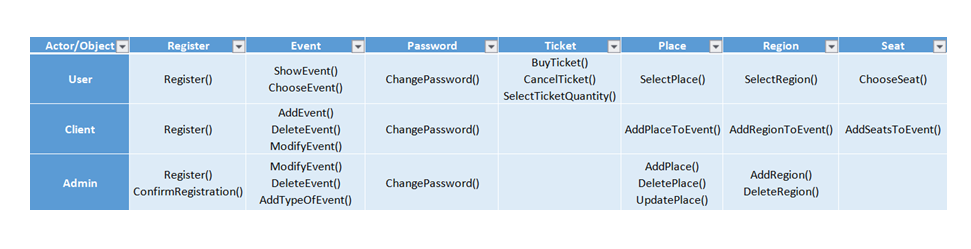
InOnline Ticket system, we decide to store persistent data in a database. The persistent datais; Users of the system (User, Client, Admin), Region data thatdisplays regions’ names. Place data that displays place’s features like city, post code, address. There is relation between place and client. We store client in member data. For each client there are some places. Event Type data shows event type name like as music, sport,and so on. Event table shows event type, date, capacity, event name, region\_id, place\_id and information of event. There is relation between region and event. For each region there are some events.There is relation between event type and event. For each event type there are some events.There is relation between place and event. For each place there are some events. Category data displays some categories like as X categories, Y categories.There is relation between category and event. For each event there are some categories. Seat data displays seat number. Thereis relation between seat and event.For each event there are some seats.

In Online Ticketdespite its advantages for the Online Ticket system, we decided to use **relational database** for the data management infrastructure. Some advantages: Security, flexibility, easyuse and so on.

Here Online Ticket’s ER diagram:



## Access Control and Security



## Global Software Control

Online Ticket users always trust. So the online ticketing system should be secure. Online Ticket is safety. And also Online Ticket provides thread safety. That means, Online Ticket system is multi-threaded program. The online ticket should be served at the same time without the problem of many people (users, admin, client etc.). Also some functions of Online Ticket should be simultaneous. In addition to, the thread introduces many system problems during debugging and testing. Such as, in our design, when users buy a one-time ticket, the tickets need to be simultaneous, because many users can try to get the same tickets at the same time. This would be a violation of our Online Ticket system.

## Boundary Conditions

Online Ticket System is initialized by the system admin invoking the ‘Initialize System’ use case. Once the initialization of the system is completed, the place, and region are added into system by the system admin. In addition, the system admin initialize the server by invoking the ‘StartServer’ use case. After that, when the system is brought from non-initialized state to steady-state, the system is opened and free for the system users (Users, Client, and Admin) to login. While the users is online and perform their tasks there might be some errors occurring during the transaction processes of the tasks.

**S**tartup: Enter the website and Login.

ShutDown. Logout and close the website

-Signing in:

\*Username and password do not match

\*Username or password is wrong

\*Username does not exist.

-Signing up :

\*E-mail address does not valid

\*Mandatory fields do not filled

-Select Payment

\*During online operation, a credit card does not valid.

# Subsystem Services

In our design, the system has a lot of subsystems. These are: User Interface, Region Interface, Category Interface, Seat Interface, EventType Interface, Event Interface, Place Interface, Login Interface,

Decoupling our application from the datasource means that the components used to implement the business logic and flow of control are not tied to the specific datasource. By utilising a pattern that adheres to this paradigm, we can easily switch out the datasource for a different kind of database or a different kind of storage medium without requiring code changes throughout the rest of our application.

Typically we achieve this by adding layers of abstraction to the data retrieval and storage process. In a web application that follows an MVC-based architecture, all of the code responsible for managing domain entities will be stored in Model classes.

Also we used the PDO(PHP Data Objects) in our project. PDO\_MYSQL is a driver that implements the PHP Data Objects (PDO) interface to enable access from PHP to MySQL 3.x, 4.x and 5.x databases.

**User Interface:** provides services for users (user, admin and client) for common interfaces, It has Login Form, Password Change Form, and etc.

**Region Interface**:provides services for users to display all regions and to select this region from box. And also provides services for admin to add new region into system.

**Category Interface:**provides services for users to display all categories and to select this category. And also provides services for client to add new category into any event.

**Seat Interface:**provides services for users to display all seats and to select this seat. And also provides services for client to add new seat into any category.

**EventType Interface:** provides services for users to display all types and to select this type (music, sport). And also provides services for admin to add new type to the system. And it provides services for client to add event type into any event.

**Event Interface:**provides services for users to select event whatever they want. And also it provides services for client to add new event into the system.

**Place Interface:** provides services for admin to add new place into the system. And also provides services for admin to add place into the any event.

**Login Subsystem:** provides services for users (user, client, and admin) to login.

**Profile Info Subsystem:** provides services for user, client, and admin to update their personal information. In addition, the subsystem also provides the service of changing password for user, client and admin.

**Data Access Subsystem:** contains all our persistent objects, this part could be called Model of MVC.

# References

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on []).

1. Bruegge B. &DutoitA.H.. (2010). *Object-Oriented Software EngineeringUsing UML, Patterns, and Java*, Prentice Hall, 3rded.