FEYZİYE SCHOOLS FOUNDATION
IŞIK UNIVERSITY

BiletBiz

System Design Document

Mehmet Keremhan Yılmaz

Büşra Nur Tekin

Bekir Malik Özkuran

Uğur Kaya Küçükkaptan

Prepared for
SOFT3101 Software Engineering

IŞIK UNIVERSITY
COMPUTER
SCIENCE AND
ENGINEERING

Table of Contents

1.Introduction

Biletbiz is a ticket selling website, coded using PHP , HTML and CSS.We are providing an online ticket selling system which will provide an easy way to purchase tickets for the user for the events they want to attend. Users will be able to receive their tickets easily and safely.

1.1. Purpose of the System

The main purpose of the system is to be a safe platform that can be easily used by users. The tickets will be on sale for events in the website.We have chosen an easy interface so that everyone can use the site very easily.In short, the website provides users the ability to view events and search the events they are interested in attending, and buy tickets for their desired events. Overall it is a simplified and easy to use system to use.

1.2.Design Goals

The design objectives determine the desired characteristics of the online ticket sales system and set criteria related to the system.

Performance:

- The system should respond to the user in less than 10 seconds when the user wants to perform any action.
- The system should register data to the MySql database very quickly.
- The processes like login authentication processes should not take long time to complete.

Supportability:

- The system must support many users at the same time.
- The system should support the English language.
- The database of the system should be able to support 15% of growth without losing any performance.

Usability:

- The user should be able to search for the events they want really easily.
- The ticket purchasing process should be easy to complete.
- The user should be able to use the system without any training.

Availability:

- The system is web-based, users can access the site through the server.They can also provide navigation on the site without the need for users to download something.

Security:

- The e-mail, password and payment information of the people who log in to the site will be confidential and will be stored safely.
- Users' password is not displayable for everyone except the owner.

Learnability:

- There should be an easy interface so every user should be able to use this site with ease.
- The system should be easy to navigate and understand such that a user should be able to successfully complete their learning phase within 10 minutes on their first time.

Feedback:

- Everyone who uses their website will receive feedback after their actions, so that the user will have received feedback on the situation in an unfavorable situation.

1.3. Definitions, Acronyms, and Abbreviations

The abbreviations used throughout the report as well as some definitions are given below.

Event An entertainment planning that is for the purpose of entertainment that happens or takes place.

Buy Function to purchase tickets.

Publish Admins making events publicly available for the user on the website.

Consumer An individual that is using the system.

Admin User type that arrange events and communicate with the operators.

Visitor A user who is not logged in to system

Registered User A user who io logged in to the system.

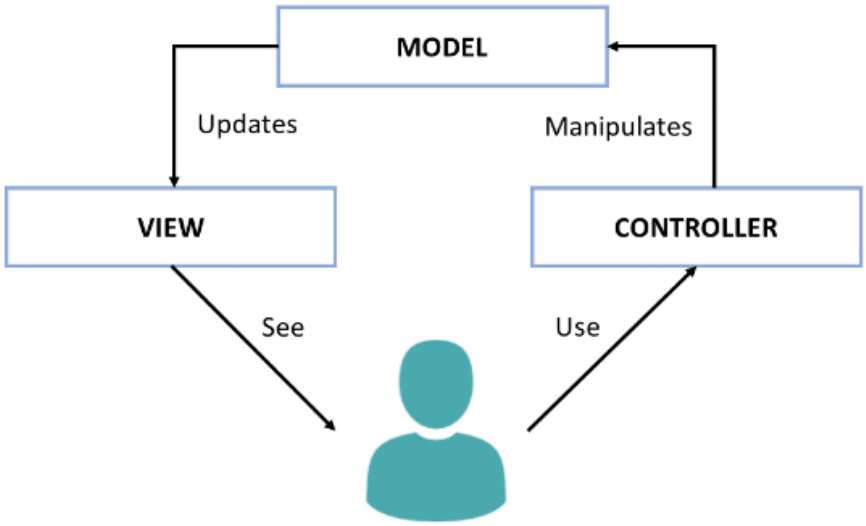
Search Different criteria functions to find an event

MVC Model-View-Controller Structure

1.4. References

2.Current Software Architecture

There are many ticket selling sites and currently, the most used website that offers the function of purchasing tickets for events in our country is www.biletix.com.
The main function of this website is to purchase tickets for a variety of events.There is a very wide range of users for whom this website hosts more than one type of activity, and our website application is an application that will be displayed via a web-based server.We chose the MVC architecture for our project.ForvView subsystem,there are four actors in the system which can interact with interface: an administrator, a registered user, a visitor, and a company that will communicate with registered users.Model subsystem consists of user management, company management, event management, ticket management and admin management. And lastly the model subsystem consists of user storage,receipt storage and event storage.



3. Proposed Software Architecture

The proposed software architecture in Biletbiz is applying the MVC model to the website system by using PHP. All functions such as Login,Register,Logout or Create Ticket Listings will be services for View Layer. They will be redirected towards the Service Management Systems as Controller Layer to work on services and data will be held in Model Layer.

3.1 Overview

We have used the MVC architecture in our system.There are 3 main subsystems included in this architecture. These systems are the Model Subsystem, the View Subsystem and the Controller Subsystem.The View Subsystems describe how the data is presented visually to the users, the model subsystems include data of websites only. The next section goes through the

subsystem decomposition in further depth. Following the subsystem decomposition, the system's hardware and software mapping is shown in its own part. The "Persistent Data Management" section outlines the sorts of data that will be stored by the system. Accessibility and security details for the data are noted in the "Access Control and Security" section.

3.2 System Decomposition

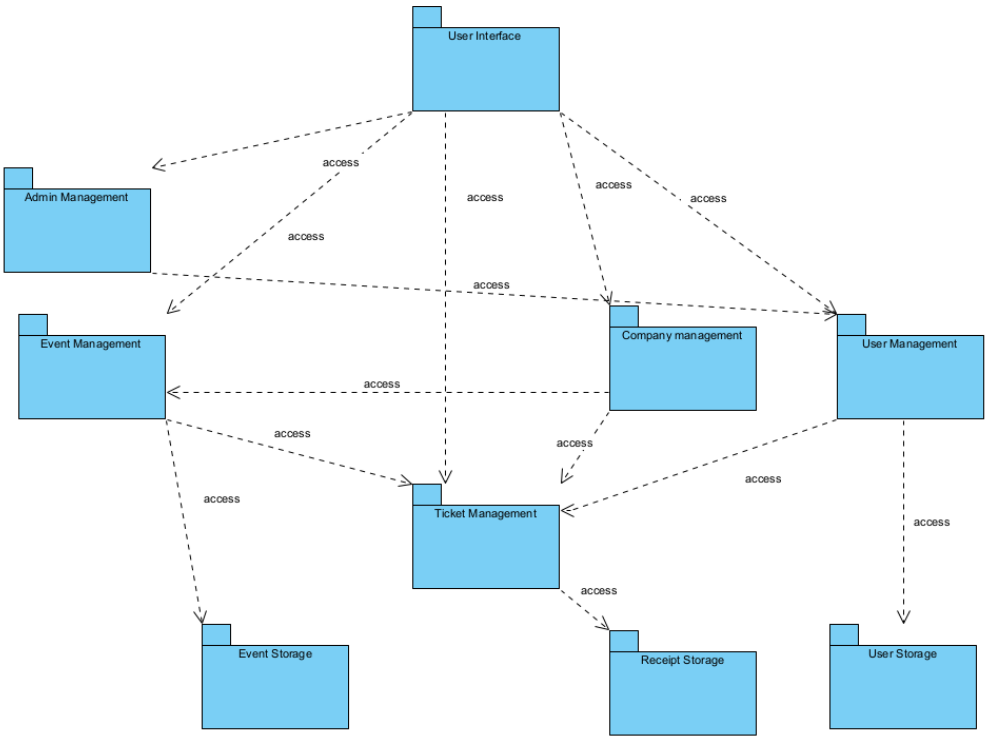


Figure 1: Subsystem Decomposition

The Model-View-Controller (MVC) architectural design is used to break down the Online Ticket System. MVC was a good fit for our system because it allows for a speedier development cycle and allows for several modifications without changing the entire model. Model, View, and Controller are the three layers of the system.

The data and where it is stored are governed by the model level. It's broken down into three distinct parts.

• Receiving event information is the responsibility of Event Storage. The name of the event, the name of the artist, the description of the event, the date of the event, the stage of the event, and other details are all included in the material.

• Receiving ticket information is the responsibility of Ticket Storage. The event and user details are both included in the ticket information.

• User information is received by the user directory. Name, surname, birthday, and other details about the user are included.

The view level is in charge of displaying the output via an interface to the user. It shows the data provided by the Controller level or the Model level.

3.3 Hardware Software Mapping

System runs on a web browser coded using PHP, and uses MySQL as Database. System is designed for every browser so it does not have any specifications.Users reach service management which contains management for user,company,event,ticket and admin by interacting with the web browser. Biletbiz servers store data using MySql database.

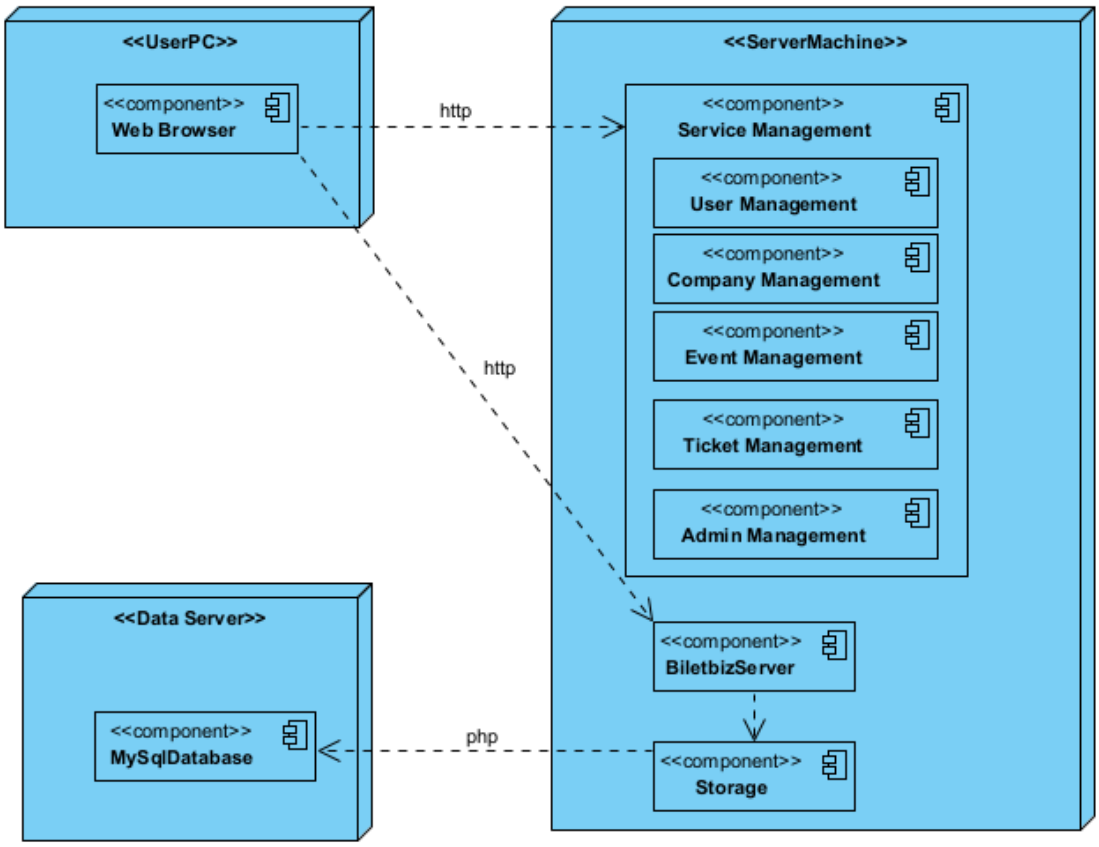


Figure 2: Hardware Software Mapping

3.4 Persistent Data Management

The Online Ticket System is a web application that allows people to buy tickets online. As a result, it keeps track of user accounts. The system uses MySQL to store the users' date of joining, email address, first name, last name, user type, password, and username. In order to carry out its primary functions, the system additionally stores events and phases. As a result, the system saves stage, date, name, price, rules, and quota for events, as well as address, location, and quota for stages.

3.5 Access Control and Security

In the system of Biletbiz, there are four kinds of actors: User(Registered User), Admins, Company and Visitors (Unregistered User). The information about users, companies and admins are stored in the system. To gain security, crucial information such as users' passwords must be encrypted. Biletbiz uses MySQL database to hold the data.MySQL access control involves two stages when you run a client program that connects to the server: For stage one, the server accepts or rejects the connection based on your identity and whether you can verify your identity by supplying the correct password for the system, and for second stage assuming that you can connect, the server checks each statement you issue to determine whether you have enough authority and privileges to perform it.

Actor	User	Administration	Ticket
Registered User	• Login() • Logout() • ForgotPassword() • ChangeUsername() • SearchEvents()		• purchaseTicket() • refundTicket() • changeUsername() • viewMyTicket() • enterPayment() • viewReceipt()
Admin	• Login() • Logout() • ForgotPassword()	• ApproveCompany() • DisapproveCompany() • BanUser() • removeCompany()	
Company	• CompanyLogin() • CompanyLogout()		• CreateTicketListing() • EditTicketListing() • ShowParticipants()
Visitor	• Register() • RegisterCompany() • SearchEvents()		

Figure 3: Access Matrix

3.6. Global Software Control

Procedure-driven control is the one that will work best with our system. When operations need data from either the website or a user type, they wait for the user (visitor, admin, company, registered user) to deliver the required information or action. Because our project is web-based, the web server and database server, for example, await requests from the web browser. Finally, each user type can access features such as how they want to view events, their tickets, or account information.

3.7. Boundary Conditions

3.7.1. Initialization

Because this system is a web-based application, it does not require any special installation. As a result, the user accesses the application using a web browser. To access the application, you'll need an internet connection. There are multiple types of users, one of them does not need to be logged in to the system but is prohibited from certain features, while the others must be logged in to use the system.

3.7.2 Termination

Logging out of the application is how you get rid of it. This advancement is linked to the session management subsystem. When we logout, one of the subsystems will be terminated, and the rest of the subsystems will be alerted of this termination. They'll communicate with other subsystems to keep processes running smoothly, and the database will be updated on a regular basis via the web server.

3.7.3 Failure

When a user fails to login by providing incorrect password or username information, or when a user fails to sign up by providing incomplete information, the system displays error messages. Due to internet connection issues, the web server may not reply. In order to avoid the busy wait, the timeout will be enforced if the sent request does not receive a response.

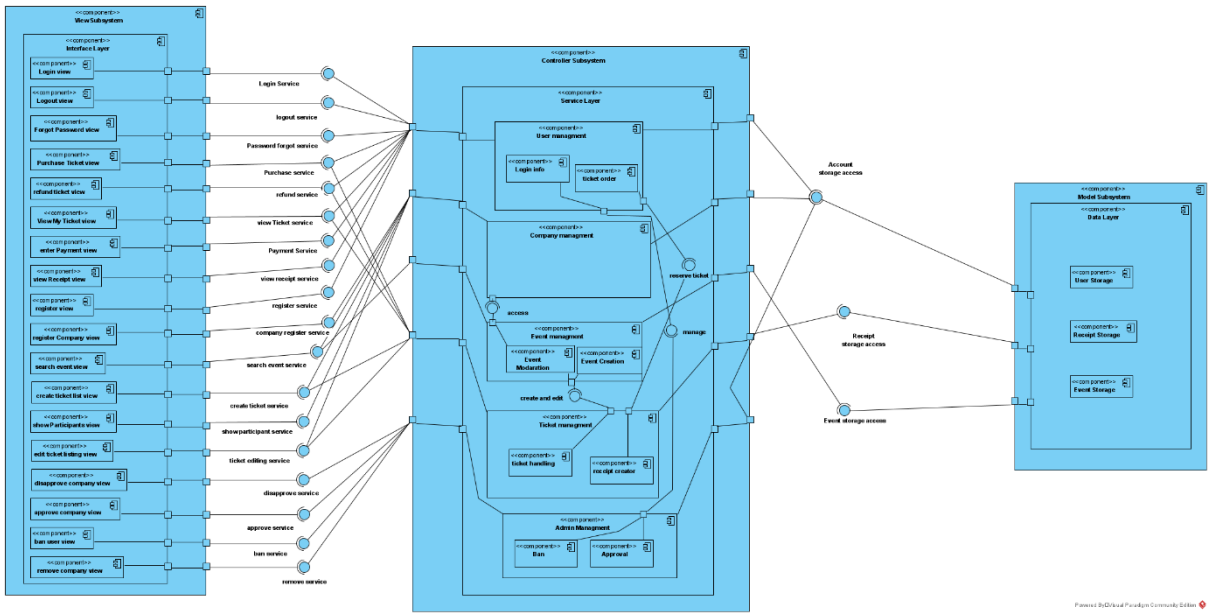
4.Subsystem Services

View Subsystem's responsibility is to show reactions coming from Controller Subsystem by the user trigger. Besides this, it is responsible for presenting the model's data to the user. View Subsystem has Login View ,Logout View, Forgot Password View, Purchase Ticket View, refund ticket View, View My ticket View, enter payment View, view receipt View, register View, register company View, search event View, create ticket List View, show participants View, edit Ticket Listing View, disapprove company view, approve company View, ban user view and remove company view components.

When a request comes to the Controller components, after the execution, the Controller object updates the data in the Model Subsystem components. Model subsystem stores information about users, companies, tickets, events and receipts.

The Controller Subsystem acts as a bridge down between the View Subsystem and the Model Subsystem. It processes at the server side of the system and provides main functionality of the system. In the system, there are five components in the Controller Subsystem: User Management, Company Management, Event Management, Ticket Management and Admin Management. User Management manages as an authentication tool for the Login, Logout, Forgot Password,Purchase Tickets, refund tickets, view my Ticket, enter payment, view receipt and register services. Company Management manages the services of register companies, search events, create ticket listing show participants and edit ticket listings. Ticket Management also helps manage previously mentioned services purchase, refund and view ticket services as well as create ticket listing and edit ticket listing services. Event management takes the service of search events. Lastly Admin management takes services from disapprove company, approve company, ban and remove company.

User management, Admin Management and Company Management stores the data in User Storage as Account storage. Tickets store their data to the receipt store to hold and lastly Event Management holds the info at Event Storage at Data Layer.



5. References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.

2. Visual Paradigm

3. Biletix.com