G&O Book Store

2016-2017 Spring

Ozan KÜÇÜK

Contents

Page

Subject

Online bookstore web application that represent G&O book store (http://gobookstore.ozankck.com).

Problem Definition

Book stores have their own system and web sites like hepsiburada attracts big interest. Because users want to reach different suppliers in a short time.

Purpose

We develop this web application for sell books. We created a supplier system so there will be multi supplier for each books, Orders system to gather different orders and pay in one time.

Business Purpose

In general book stores present their own sale system. We gather the suppliers to one platform so it will be more useful users. Users will be abte to see different saller and different price for each book. Benefit of this each customer can prefere which supplier appropriate for own.

Software Development Life Cycle

- **Requirement Gathering and Analysis**

In the market area each book store have their shopping system and customers visit and check each web site one by one. This process takes to much time and customers want to spend less time while looking for a book. Customers' expectations are not meet in many system.

- **Design**

We created a web site that meet customers' needs. The system will be show in below.

- **Implementation / Coding**

Firstly we created a database model and according to that we start to develop project based on database first method. We used technologies shown and described in below.

- **Testing**

We tried every function in system minimum twice to be sure that there is no visible error come out to harm our application.Each function tested successfully and did not encounter with any problem in normal routines.

- **Deployment**

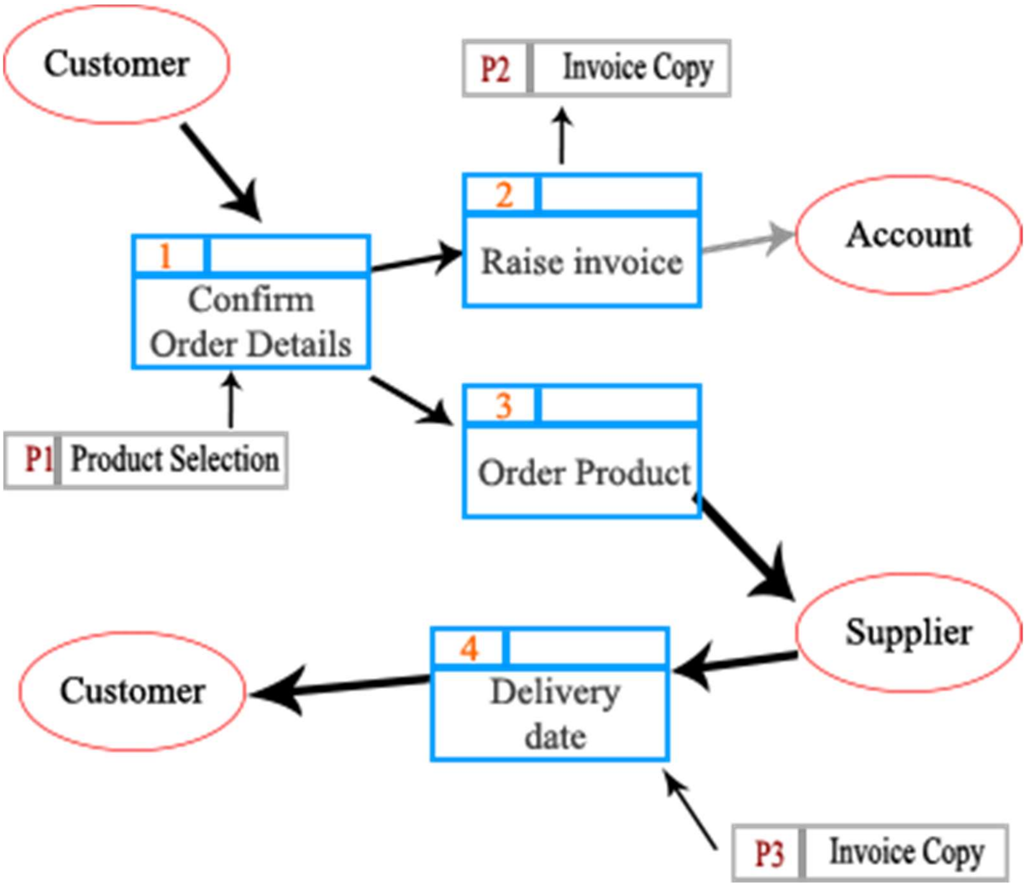
We opened a host for the application and all the functions can be use by different users.
You can access to web site on : gobookstore.ozankck.com

- **Maintenance**

Project published to live during the time if any error occurs we able to solve the errors.

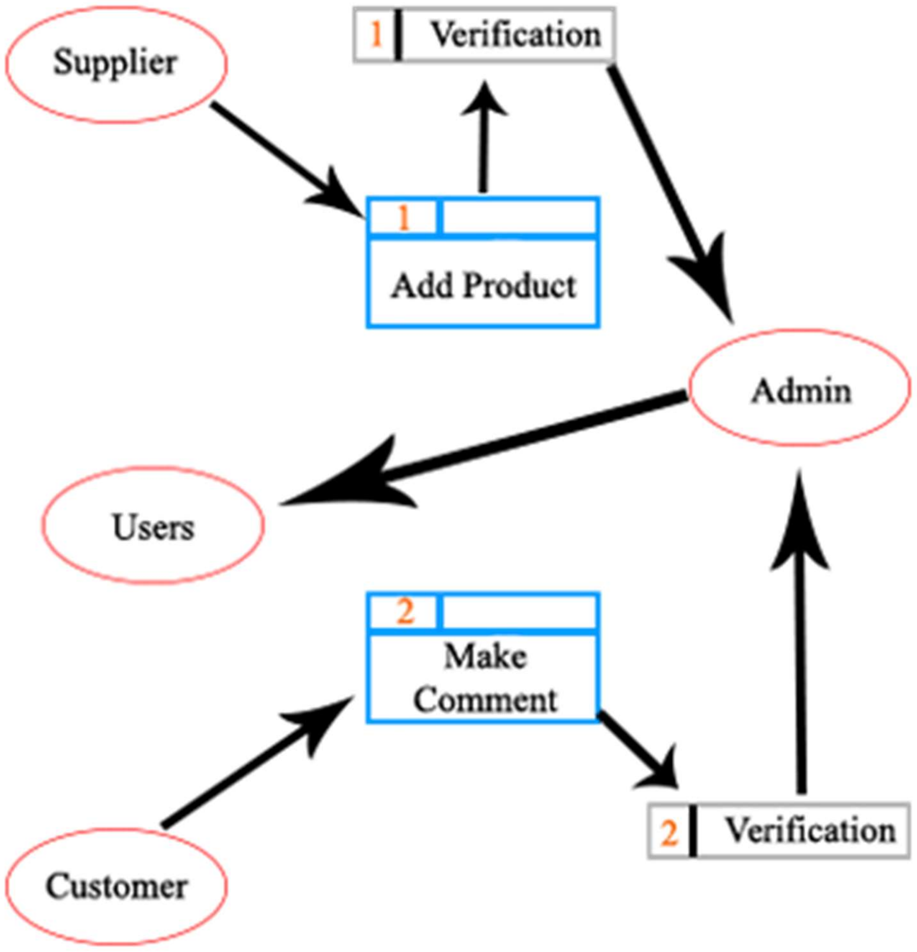
UML Diagrams

Data Flow Diagram



That diagram shows the process steps that if customer want to buy a book from web site.

Customer select a product then add to cart when product added to cart the invoice will be raise and customer will see reflection of that to account. When customer checkout the book and complete the buying process supplier get the delivery request and do the shipping process and Customer will get the book on delivery date.

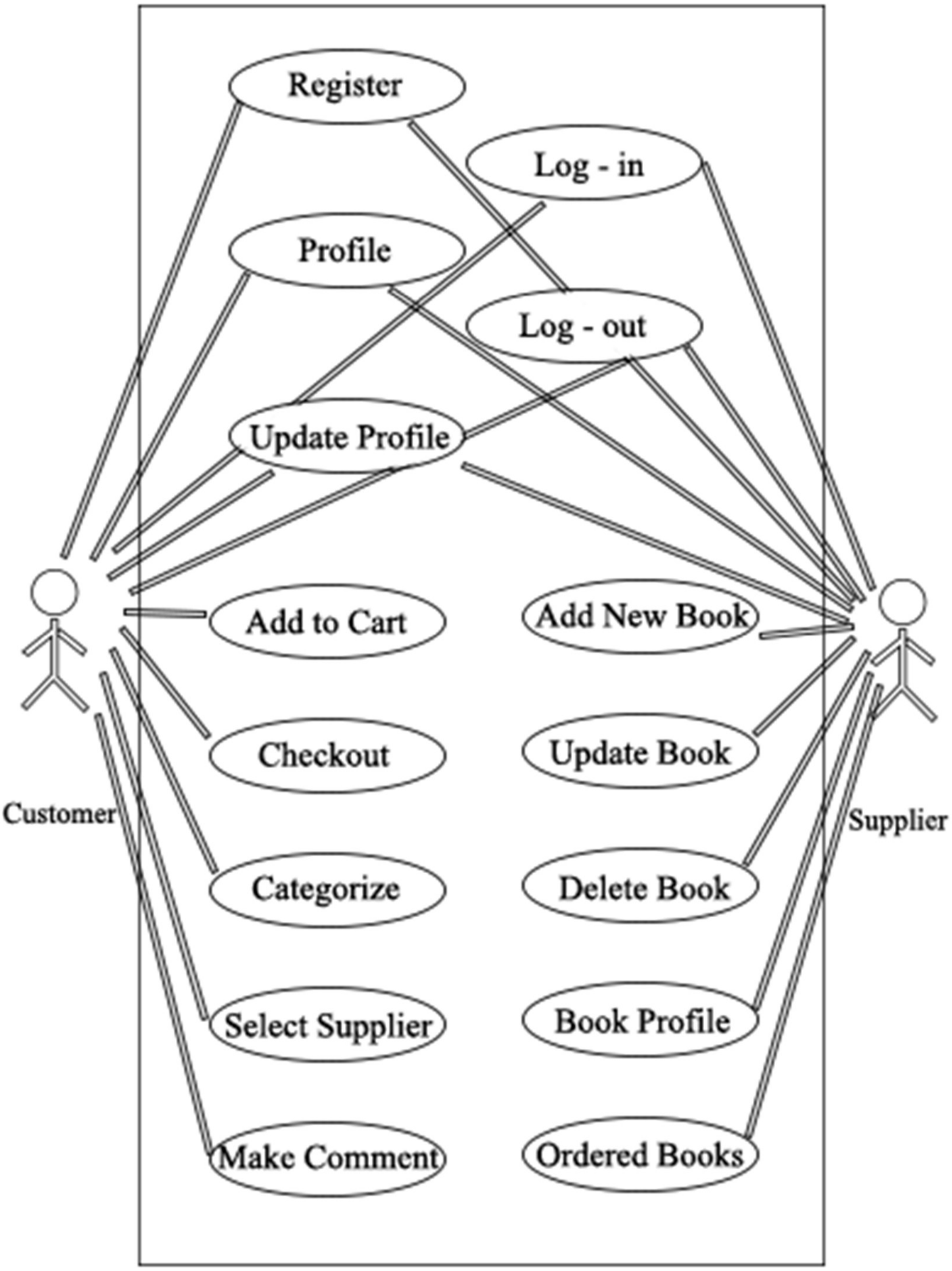


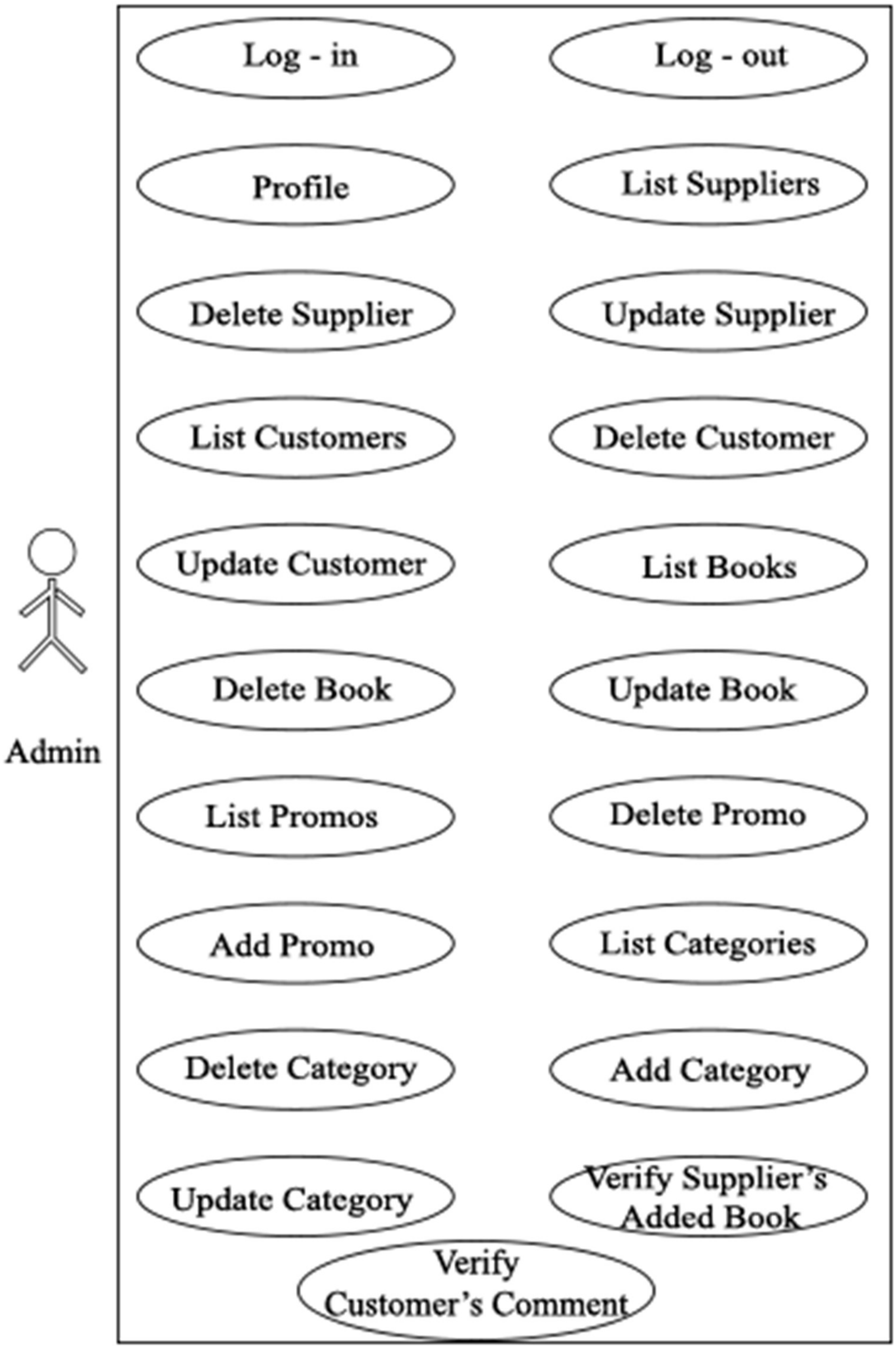
That diagram shows the process steps of verification.

Firstly supplier add a book to system. This create a request to admin for verify the book if admin verify the book every user of web site can see the book.

Secondly customer make a comment to a book and that create a request to admin for verify the comment admin checks the detail of comment and if admin verify the comment every user of web site can see that comment in related book.

Use Case Diagram





**** Admin connect to all the operations.*

Use case explanations

Use case name: Register
Actor: Customer, Supplier
Summary: User of the web site can become a customer or supplier. So user want to register to system.

Use case name: Log - in
Actor: Customer, Supplier
Summary: After users create an account they are able to log in to system.

Use case name: Log - out
Actor: Customer, Supplier
Summary: While customer or supplier logged in to system they are able to log out.

Use case name: Profile
Actor: Customer, Supplier
Summary: While customer or supplier logged in to system they are able to see thir personel informations saved to system.

Use case name: Update Profile
Actor: Customer, Supplier
Summary: While customer or supplier logged in to system they are able to see thir personel informations saved to system. And they can update that informations if they want to change

Use case name: Add to Cart
Actor: Customer
Summary: Customers are able to add a book to cart so they can make payment and buy book

Use case name: Checkout
Actor: Customer
Summary: After books added to cart customers do checkout to complete buying process

Use case name: Categorize
Actor: Customer
Summary: Customer can categorize the books according to category type.

Use case name: Select supplier
Actor: Customer
Summary: When customers want to buy a book they can select a different supplier in book detail section. There are multiple suppliers for each book and customers can choose any supplier that sell related book.

Use case name: Make Comment
Actor: Customer
Summary: Customer can make comment to book and see other customers comments too.

Use case name: Add New Book
Actor: Supplier
Summary: Suppliers are able to add new book to system. System provide multiple suppliers to customers.

Use case name: Update Book
Actor: Supplier
Summary: Once suppliers add a book to system they are able to update that book's informations such as price, category etc.

Use case name: Delete Book
Actor: Supplier
Summary: Supplier can delete book from system.

Use case name: Book Profile
Actor: Supplier
Summary: Suppliers are able to see their all books added to system in a tab and see their informations too.

Use case name: Ordered Book
Actor: Supplier
Summary: Supplier see the books which are added to cart by which customer in order to start shipping process.

Use case name: Log - in
Actor: Admin
Summary: Admin can login to system in admin sections. Other users cant see that section.

Use case name: Log - out
Actor: Admin
Summary: Admin can log out from system.

Use case name: Profile
Actor: Admin
Summary: Admin see the informations of the account

Use case name: List Suppliers
Actor: Admin
Summary: Admin can see all the suppliers that registered to system.

Use case name: Delete Supplier
Actor: Admin
Summary: Admin can delete supplier from system directly. That supplier account remove from system permanently.

Use case name: Update Supplier
Actor: Admin
Summary: Admin can update any information about supplier.

Use case name: List Customers
Actor: Admin
Summary: Admin can see all the customers that registered to system.

Use case name: Delete Customer
Actor: Admin
Summary: Admin can delete customer from system directly. That customer's account remove from system permanently.

Use case name: Update Customer
Actor: Admin
Summary: Admin can update any information about customer.

Use case name: List Books
Actor: Admin
Summary: Admin can see all the books that added to system by any supplier.

Use case name: Delete Book
Actor: Admin
Summary: Admin can directly delete a book from system.

Use case name: Update Book
Actor: Admin
Summary: Admin can update any information about book

Use case name: List Promos
Actor: Admin
Summary: Admin see the all promos in that section.

Use case name: Delete Promo
Actor: Admin
Summary: Admin can delete a promo from system and informations about it will be gone.

Use case name: Add Promo
Actor: Admin
Summary: Admin can create a promo and make configration of the promo.

Use case name: List Categories
Actor: Admin
Summary: Admin see all categories added to system.

Use case name: Delete Category
Actor: Admin
Summary: Admin can delete a category from system but there must not be any releated book with that category

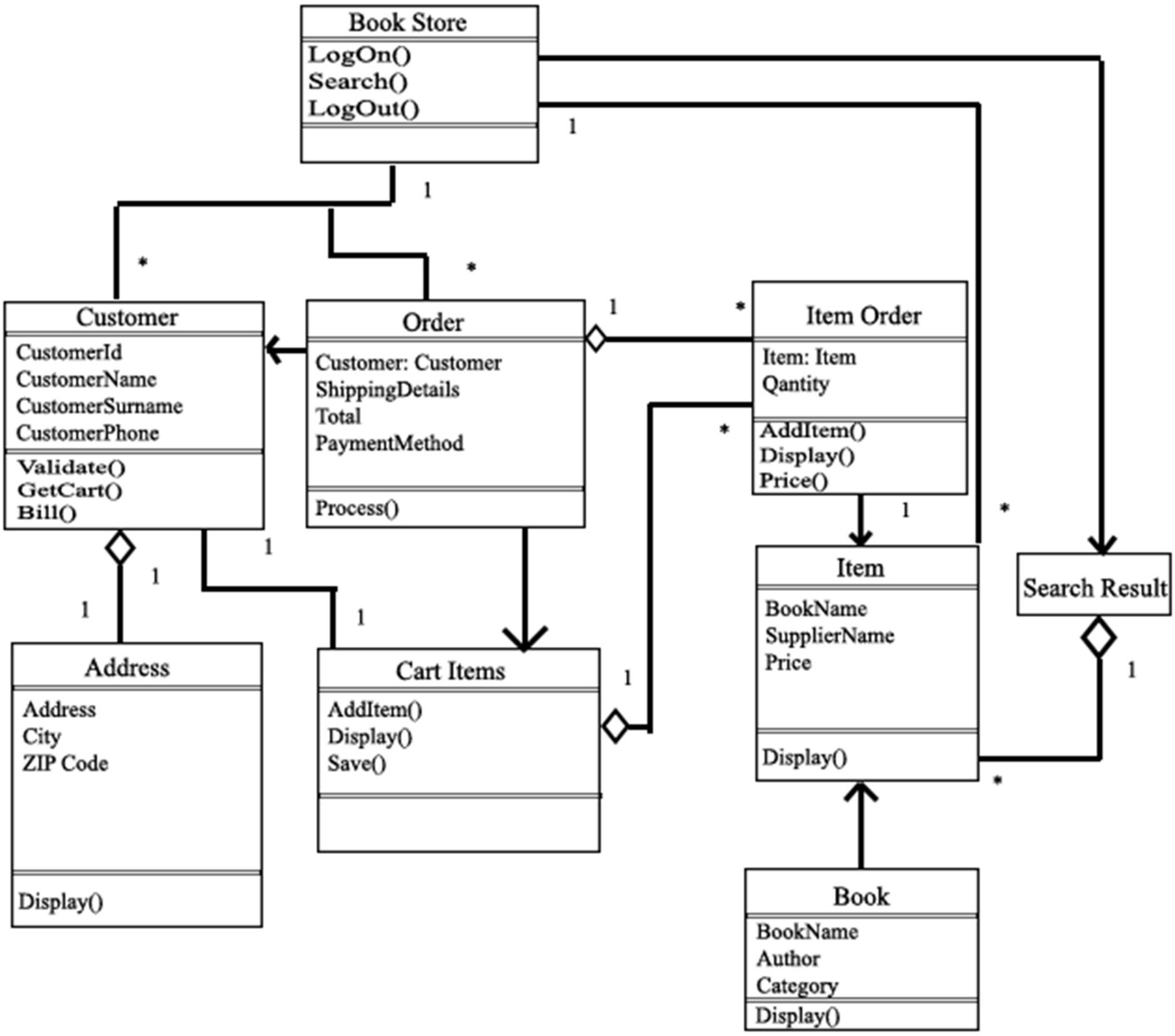
Use case name: Add Category
Actor: Admin
Summary: Admin can add a new category to system.

Use case name: Update Category
Actor: Admin
Summary: Admin can update category name or description of any category.

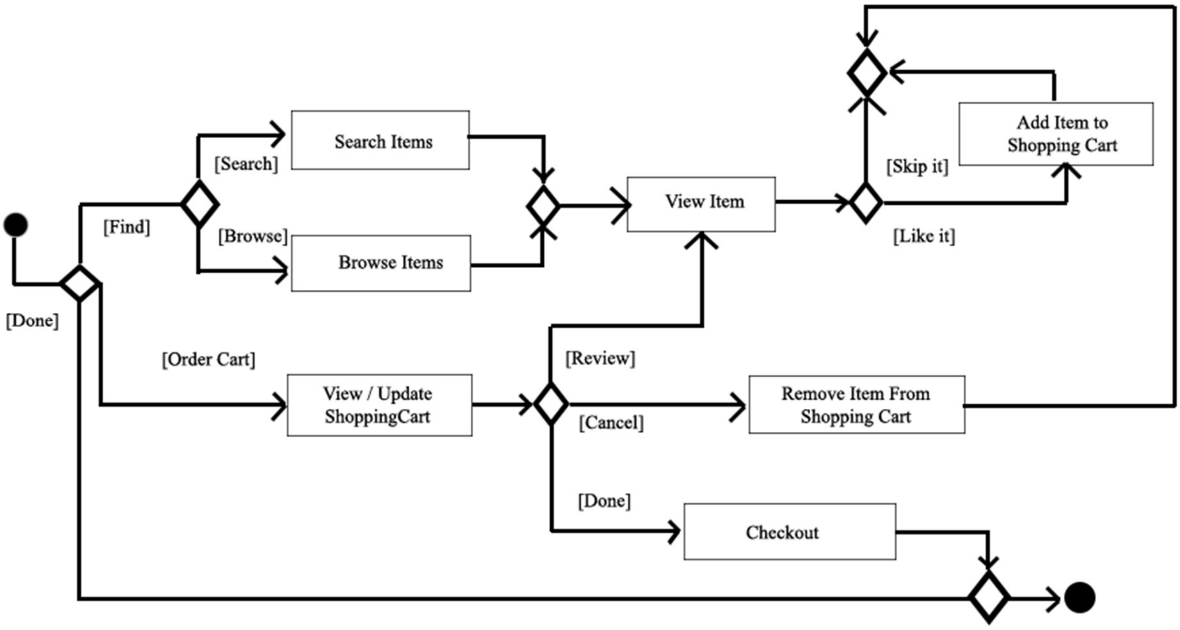
Use case name: Verify Supplier's Added Book
Actor: Admin
Summary: After a book added to system by any supplier admin must verify that book to customers can see that book on book section.

Use case name: Verify Customer's Comment
Actor: Admin
Summary: After any comment to any book admin must verify that comment to comment must be seen by other customers

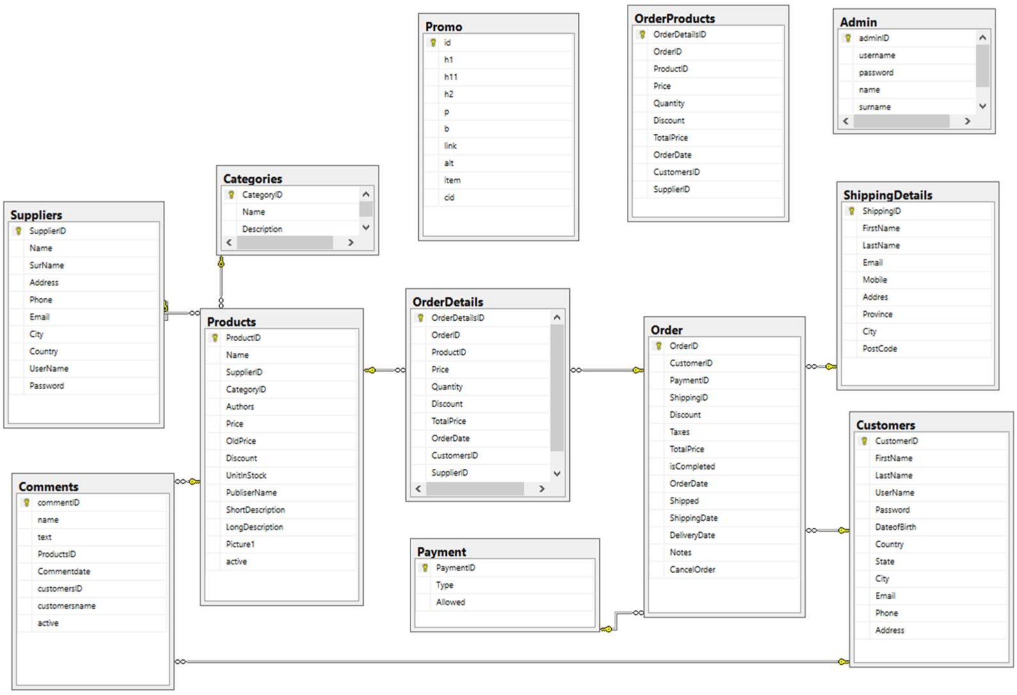
Class Diagram



State Diagram



Data Diagram



Implementation

MVC :

In object-oriented programming development, model-view-controller (MVC) is the name of a methodology or design pattern for successfully and efficiently relating the user interface to underlying data models. The MVC pattern is widely used in program development with programming languages such as Java, Smalltalk, C, C# and C++.

The model-view-controller pattern proposes three main components or objects to be used in software development:

- A *Model* , which represents the underlying, logical structure of data in a software application and the high-level class associated with it. This object model does not contain any information about the user interface.
- A *View* , which is a collection of classes representing the elements in the user interface (all of the things the user can see and respond to on the screen, such as buttons, display boxes, and so forth)
- A *Controller* , which represents the classes connecting the model and the view, and is used to communicate between classes in the model and view.

Example:
ProductsController

```
public ActionResult Index(int? page)
  {
        using (bookStoreDBEntities data = new bookStoreDBEntities())
        {
                      int Page1 = page ?? 1;

                      var urunler = data.Products.Where(b => b.active
              ==true). OrderByDescending(m =>
              m.UnitInStock).ToPagedList<Products>(Page1, 12);

              return View(urunler);
        }
  }
```

Here we take information about books from database and then informations send to html page with in 12 product in each page.We define start page as Page 1.
TopageList is in PagedList.Mvc library. This library helps us to build pages. Usage of it is provided by Nuget

Connection of database is done with using. This technic provide that after use the function connection will close automatically.

HTML Page

The model which comes from controller used with loop for books. Because of we used ToPagedList we include content of that library to html section. Also we should define the model for usage. In here we defined Products model and used it.

```
@using PagedList;
@using PagedList.Mvc;
@model IPagedList<BookStore.Models.Products>
```

```
@{
    ViewBag.Title = "Books Page";
}
<!--features_items-->

<h2 class="title text-center">Features Items</h2>
                        foreach (var item in Model)
                        {
<div class="col-sm-3">
    <div class="product-image-wrapper">
        <div class="single-products">
            <div class="productinfo text-center">
                <a href="@Url.Action("Details","Products", new { id = @item.ProductID,
            bname = Model.Name })"> <img src="@item.Picture1" alt="" /> </a>
                <h2>@item.Price <span>₺</span></h2>
                <h5>@item.Name</h5>

                         @if(item.UnitInStock > 0)
                         {
                <p>@item.Authors</p>
                <a href=" @(Session["username"] != null ? Url.Action("AddToCart",              "Products",
new { id = @item.ProductID , qty =1 , user =                                          Session["UserName"]})  :
Url.Action("Index", "Accounts")) " class="btn               "><i class="fa fa-shopping-cart"></i>Add to
cart</a>
               }
               else
               {
            <a class=" add-to-cart"><i class="fa fa-shopping-cart"></i>Not Fount Stoks</a>
               }
            </div>
        </div>
    </div>
</div>
 }
</div><!--features_items-->

<!—Sayfalar arası gecıs yapma  1-2-3-4-..son-->
<div class=" pager">@Html.PagedListPager(Model, page1 => Url.Action("Index", "Products", new {
page = page1 }), new PagedListRenderOptions { LinkToFirstPageFormat = "<< İlk",
LinkToPreviousPageFormat = "< Önceki", LinkToNextPageFormat = "Sonraki >",
LinkToLastPageFormat = "Son >>" })
                </div>
```

At the bottom we used @Html.PagedListPager comes from PagetList.mvc to separate pages. This provede us if there are 100 product in list separate pages that include 12 product in each page.

ORM :

Object-relational mapping (ORM) is a mechanism that makes it possible to address, access and manipulate objects without having to consider how those objects relate to their data sources.

ORM tools:

1. **ORM frameworks for Java:**

- Hibernate
- JPA
- OpenJPA
- Toplink
- EclipseLink
- Apache Cayenne
- MyBattis

2. **ORM frameworks for .Net:**

- Entity Framework
- Nhibernate
- .Net Persistence
- BBADataObjects-
- DataObjects.NET
- DotNorm
- FastObjects.NET
- Norm
- OJB.NET

3. **ORM frameworks for PHP:**

- Propel
- Doctrine
- PHP-Activerecord
- PdoMap
- RedBean

We use Entity Fremework ORM.

What is Entity Framework?

The Microsoft ADO.NET Entity Framework is an Object/Relational Mapping (ORM) framework that enables developers to work with relational data as domain-specific objects, eliminating the need for most of the data access plumbing code that developers usually need to write. Using the Entity Framework, developers issue queries using LINQ, then retrieve and manipulate data as strongly typed objects. The Entity Framework's ORM implementation provides services like change tracking, identity resolution, lazy loading, and query translation so that developers can focus on their application-specific business logic rather than the data access fundamentals.

Example 1:

```
public ActionResult ListValidComment(int id, string Ladmin)
    {
        using (bookStoreDBEntities db = new bookStoreDBEntities())
        {
            Comments comm = db.Comments.Find(id);
            comm.active = true;
            db.Comments.AddOrUpdate(comm);
            db.SaveChanges();
            return RedirectToAction("ListComment", "AdminSite");
                    }
    }
```

Here first get the id and according to that id we get the comment in Comments table.
We fill the active column as true value.
AddOrUpdate update all the information on table.
using System.Data.Entity.Migrations; includes AddOrUpdate fuction.
After that we save the changes to database.

Example 2

```
public ActionResult Search(string product, int? page)
        {
            using (bookStoreDBEntities data = new bookStoreDBEntities())
            {
                        List<Products> productss;
                if (!string.IsNullOrEmpty(product))
                {
                    productss = data.Products.Where(b => b.active == true).Where(x =>
            x.Name.Contains(product)).ToList();
                }
                else
                {
              productss = data.Products.Where(b => b.active == true).ToList();
                        }

                return View("Index", productss.ToPagedList(page ?? 1, 8));
            }
        }
```

Here we create a new list inside of product table and select the products which has true value on Active column .

With the contains we seach items according to the it’s name.

Contains prodeve making search on columns.For example in name column we want to search hob and if there are Books Named as “Hobbit” and “The Hobbit” it takes both.

Modal Box :

A modal box is a popup window that is displayed on top of the current page. Created a Modal Box with CSS and JavaScript.

Step 1) Add HTML:

In homepage layout we write html codes.This codes are not visible when someone enter to page. When someone click the button we read the button id with Javascript and call the pop-up screen.

```
<div class="modal-content">
  <div class="modal-header">
    <h2>Modal Header</h2>
  </div>
  <div class="modal-body">
    <p>Some text in the Modal Body</p>
    <p>Some other text...</p>
  </div>
  <div class="modal-footer">
    <h3>Modal Footer</h3>
  </div>
</div>

<button id="myBtn">Open Modal</button>
```

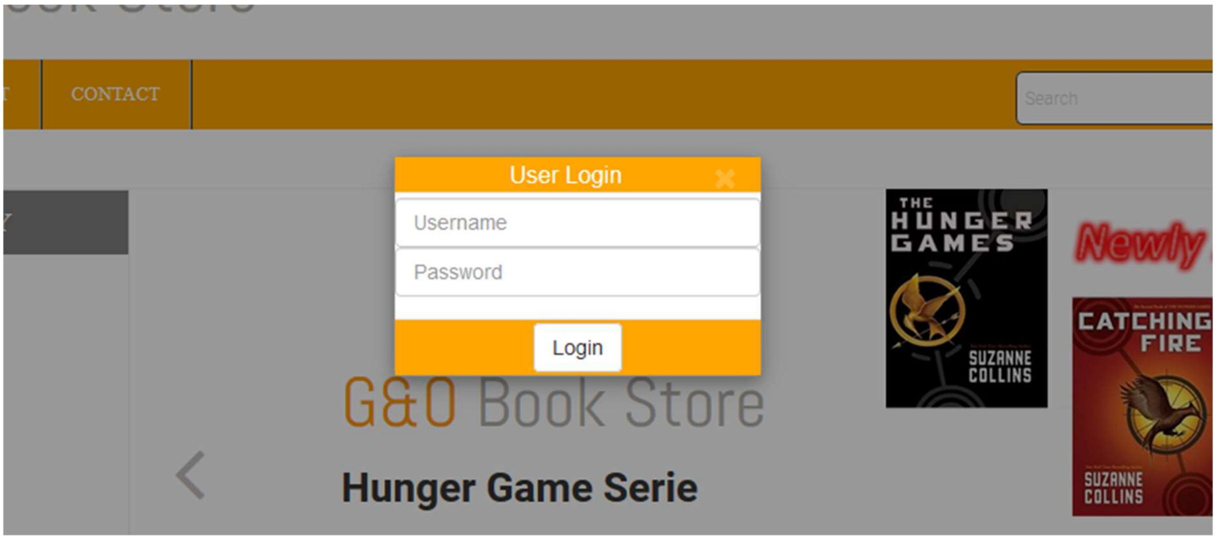
Button for run the Modal box.

Step 2) Add CSS:

```
/* Modal Header */
.modal-header {
    padding: 2px 16px;
    background-color: #5cb85c;
    color: white;}
/* Modal Body */
.modal-body {padding: 2px 16px;}
/* Modal Footer */
.modal-footer {
    padding: 2px 16px;
    background-color: #5cb85c;
    color: white;}
/* Modal Content */
.modal-content {
    position: relative;
    background-color: #fefefe;
    margin: auto;
    padding: 0;
    border: 1px solid #888;
    width: 30%;}
```

Step 3) Add JavaScript:

```
// Get the modal
var modal = document.getElementById('myModal');
// Get the button that opens the modal
var btn = document.getElementById("myBtn");
// When the user clicks on the button, open the modal
btn.onclick = function() {
   modal.style.display = "block";}
```



Bootstrap :

Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for common user interface components like Typography, Forms, Buttons, Tables, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel and many other as well as optional JavaScript extensions.

Example:

```
<div class="container">
  <div class="row">
    <div class="col-sm-8">

        <ul class="nav nav-pills">
          <li><a href="#"><i class="fa fa-phone"></i>96</a></li>
          <li><a href="#"><i class="fa fa-envelope"></i>
                                        info@gobookstore.com</a></li>
        </ul>
    </div>
    <div class="col-sm-4">
                @Html.Actionlink(“home PAge”, “Home”)
    </div>
  </div>
</div>
```

Container:

Bootstrap requires a containing element to wrap site contents and house our grid system. Note that, due to padding and more, neither container is nestable.

Row:

Must be placed within a .container (fixed-width) for proper alignment and padding.

Col-(*)-(*):

Bootstrap's grid system allows up to 12 columns across the page. If you do not want to use all 12 column individually, you can group the columns together to create wider columns

Application Insights :

Application Insights is an extensible Application Performance Management (APM) service for web developers on multiple platforms. Use it to monitor your live web application. It will automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and J2EE, hosted on-premises or in the cloud. It integrates with your devOps process, and has connection points to a variety of development tools.

Example:

Fısrty import application insights to the Project

```
using BookStore.Models;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Helpers;
using System.Web.Mvc;
using Microsoft.ApplicationInsights;


namespace BookStore.Controllers
{
    public class HomeController : Controller
    {
        private TelemetryClient telemetry = new TelemetryClient ();

        public ActionResult Index()
        {
            using (bookStoreDBEntities data = new bookStoreDBEntities())
            {
                ViewBag.cata = data.Categories.ToList().OrderBy(x => x.Name);

                ViewBag.book =data.Products.Where(b=>b.active == true).OrderByDescending(u =>
u.ProductID).Take(8).ToList();
                ViewBag.pro = data.Promo.ToList();

                ViewBag.rec2 = data.Products.Where(b => b.active == true).OrderByDescending(b =>
b.SupplierID).Take(4).ToList();
                ViewBag.rec = data.Products.Where(b => b.active == true).OrderByDescending(b =>
b.Price).Take(4).ToList();

                telemetry.TrackPageView("HomePage");
                return View();
            }
        } } }
```

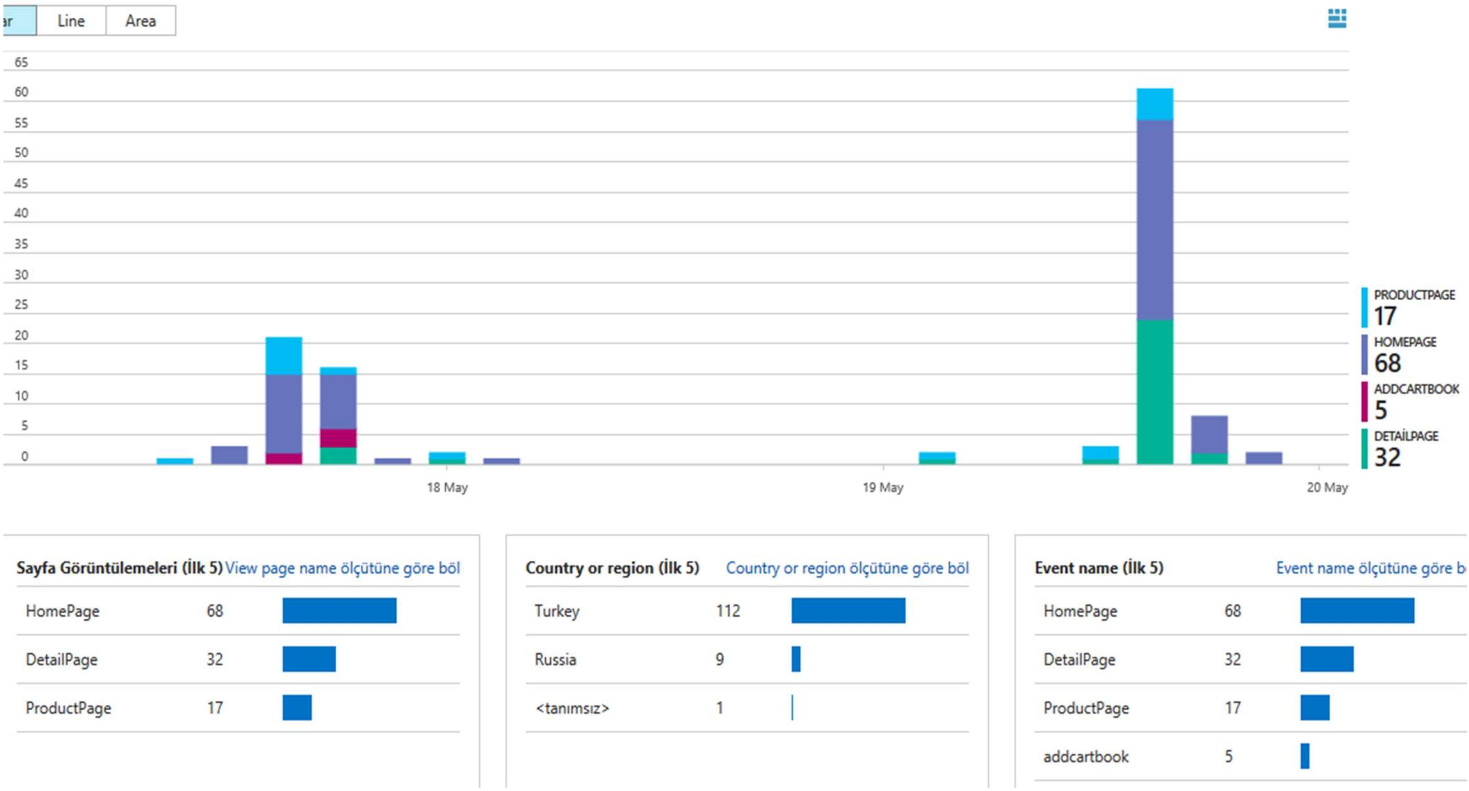
To use the Application Insightst add library to controllers:

```
using Microsoft.ApplicationInsights;
```

Once we import it to project inside of web config some adjustments are being made to connect Application Insightst automatically. And we start connection just like database connection.

```
private TelemetryClient telemetry = new TelemetryClient();
```

In here we want to see that, which page in which time from where displayed. And for that telemetry.TrackPageView is used. “Homepage” is the value of page to show which page is it in analys page. In below you can see the results of last 3 days informations.

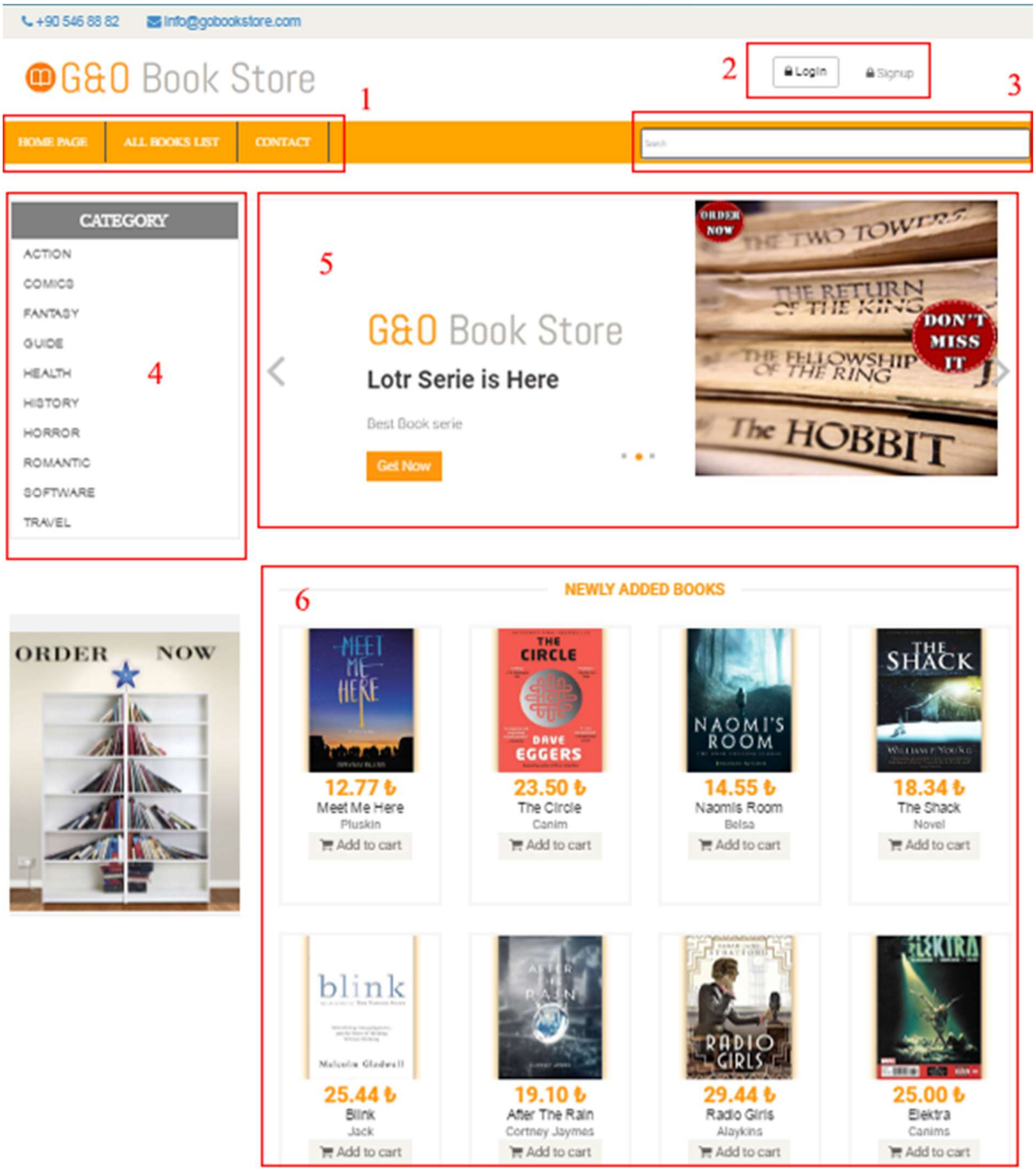


Sayfa Görüntülemeleri (İlk 5)	View page name ölçütüne göre böl
HomePage	68
DetailPage	32
ProductPage	17

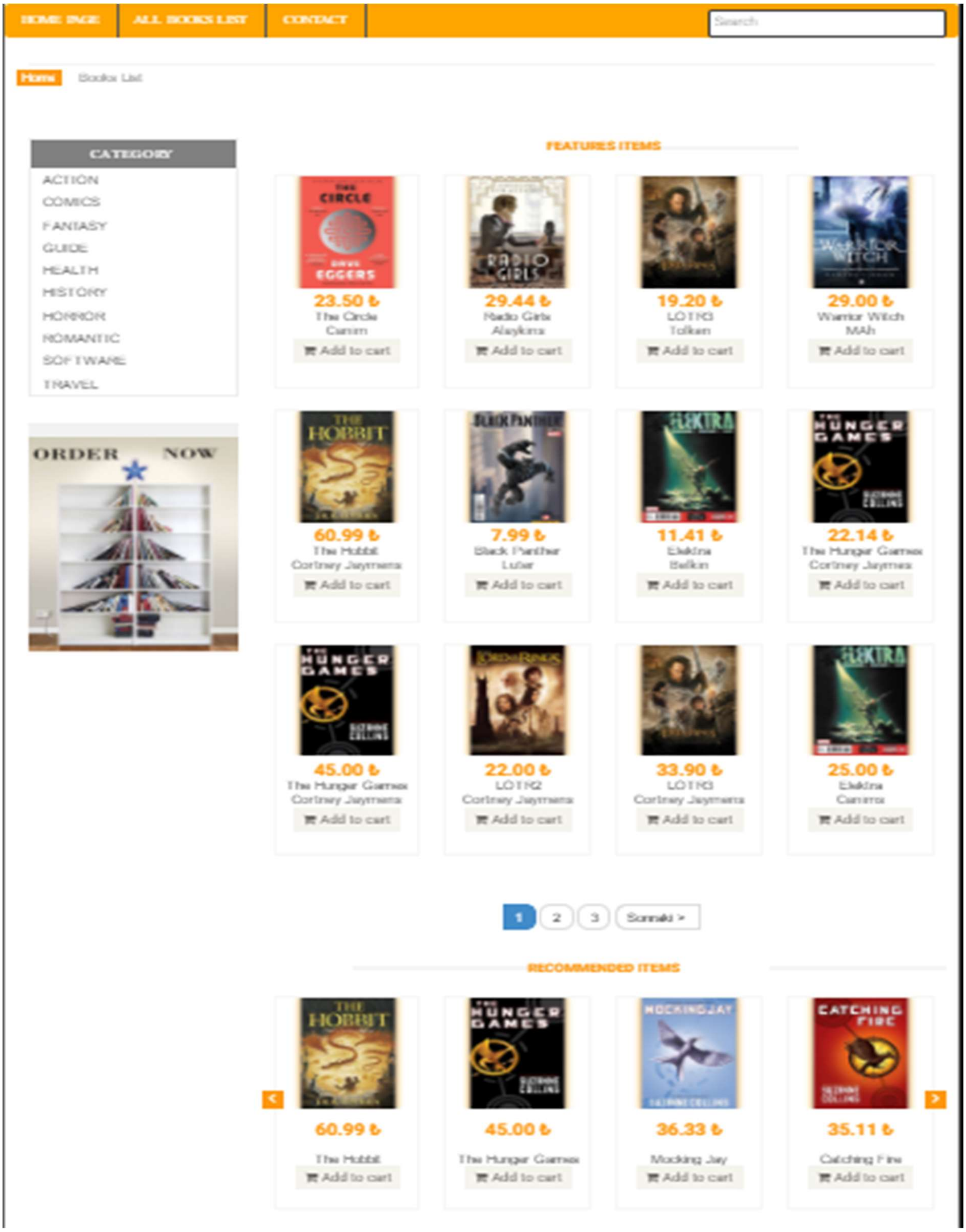
Country or region (İlk 5)	Country or region ölçütüne göre böl
Turkey	112
Russia	9
<tanımsız>	1

Event name (İlk 5)	Event name ölçütüne göre b
HomePage	68
DetailPage	32
ProductPage	17
addcartbook	5

User Interfaces with explanations



Sec1 : The pages you can see; Homepage is showing on the picture and you can display promos and books together in homepage. All Books List is a page that you can see all the books on that page seperated to pages. Contact is a page for users can get information about G&O bookstore or users can send a mail to us. And also users see the social media links.
Sec2 : Log in is perform a pop-up screen to users so they can log in. Sign up redirect you to account create page so you can become a member.
Sec3 : Provide you to a search field you can search a book according to characters on it's name.
Sec4 : Category section in here you can see books which belong only to that category type.
Sec5 : Promo field is provide some advert links to releated books.
Sec6 : Newly added books provide you last added books. You can see last 8 book added to system.



23.50 ₺
The Circle
Camim
Add to cart

29.44 ₺
Radio Girls
Alaykina
Add to cart

19.20 ₺
LOTR3
Tolkien
Add to cart

29.00 ₺
Warrior Witch
MAh
Add to cart

60.99 ₺
The Hobbit
Cortney Jaymera
Add to cart

7.99 ₺
Black Panther
Lufer
Add to cart

11.41 ₺
Elektra
Belkin
Add to cart

22.14 ₺
The Hunger Games
Cortney Jaymera
Add to cart

45.00 ₺
The Hunger Games
Cortney Jaymera
Add to cart

22.00 ₺
LOTR2
Cortney Jaymera
Add to cart

33.90 ₺
LOTR3
Cortney Jaymera
Add to cart

25.00 ₺
Elektra
Camira
Add to cart

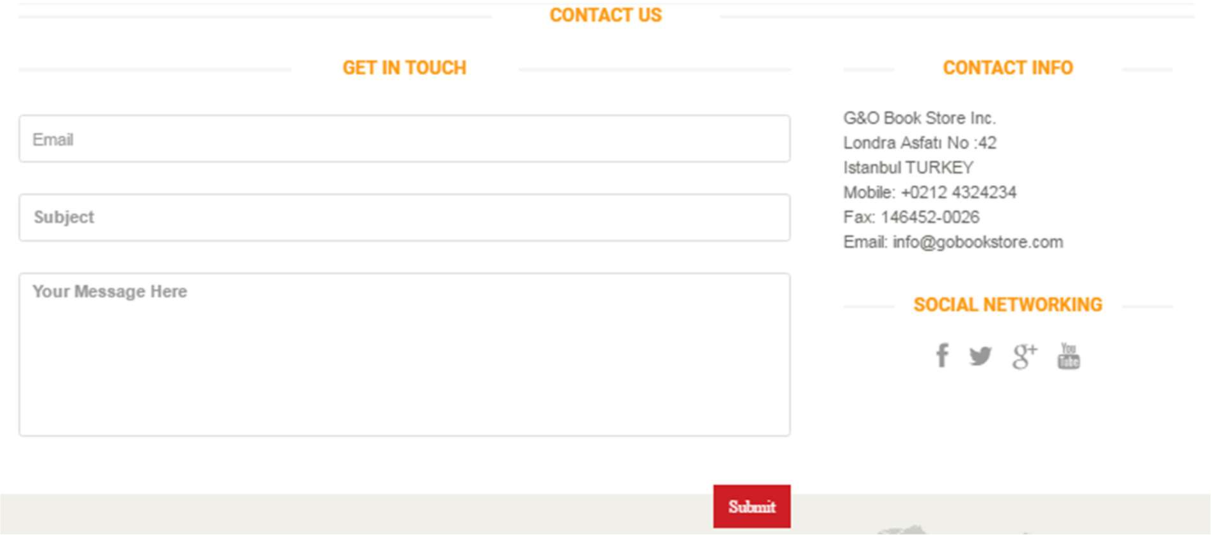
60.99 ₺
The Hobbit
Add to cart

45.00 ₺
The Hunger Games
Add to cart

36.33 ₺
Mocking Jay
Add to cart

35.11 ₺
Catching Fire
Add to cart

This is All Books List page. You can see books and pages. And at the bottom you see the recommended books which 4 books have highest price and last registered suppliers books.



CONTACT US

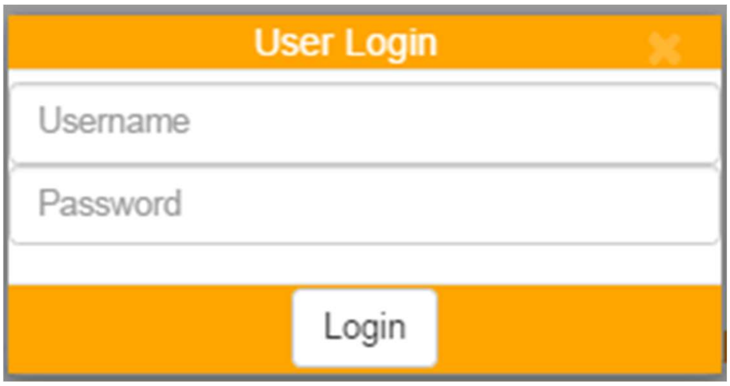
GET IN TOUCH

CONTACT INFO

G&O Book Store Inc.
Londra Asfatı No :42
Istanbul TURKEY
Mobile: +0212 4324234
Fax: 146452-0026
Email: info@gobookstore.com

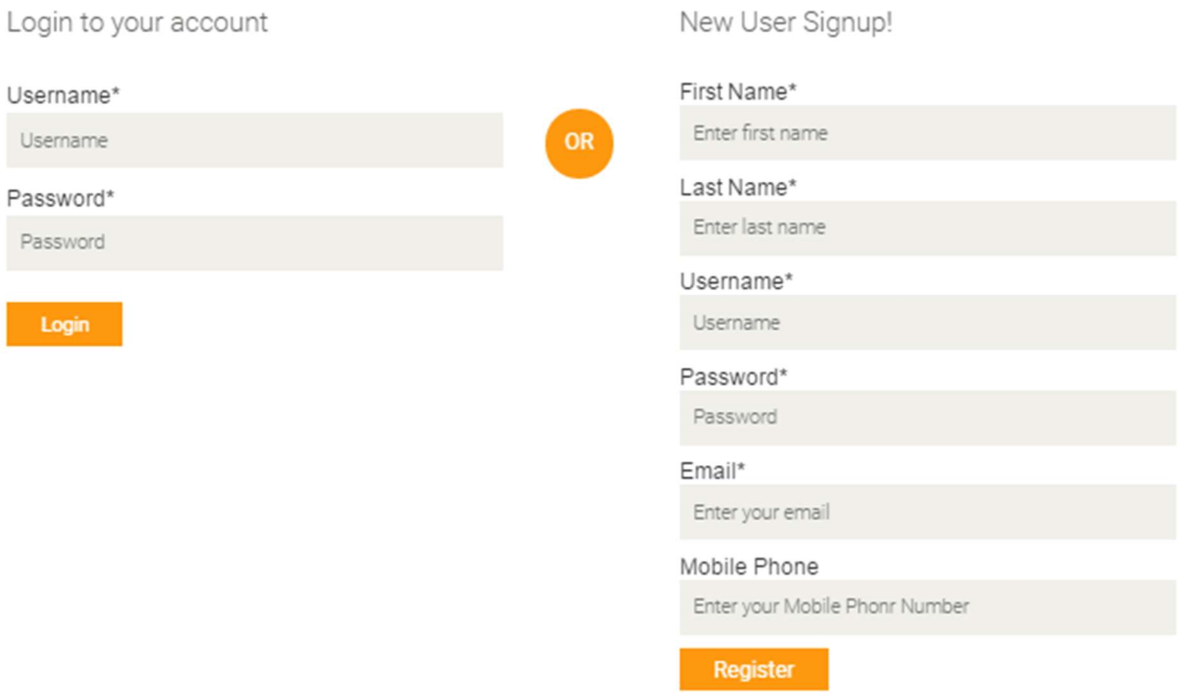
SOCIAL NETWORKING

This is Contact page and users can send a mail to us or get the address, mobile etc informations.

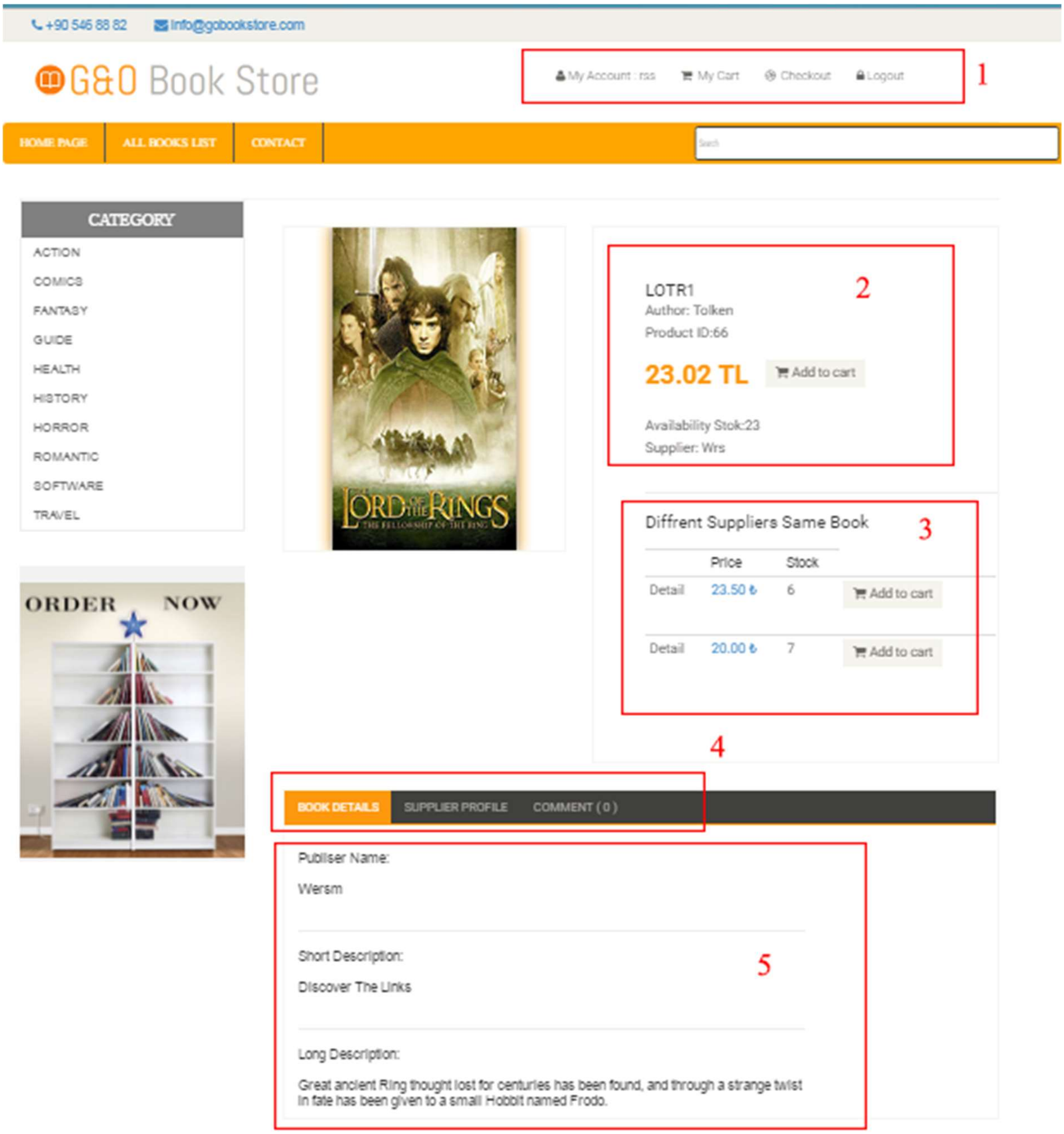


User Login

This is login section. If users have registered to system they can log in to system and start to buy books.



This is Signup section. Users of web site can create an account and they can become a member of system. And they can start to buy books.



LOTR1
Author: Tolken
Product ID:66
23.02 TL
Availability Stok:23
Supplier: Wrs

Diffrent Suppliers Same Book

	Price	Stock	
Detail	23.50 ₺	6	Add to cart
Detail	20.00 ₺	7	Add to cart

Publiser Name:

Wersm

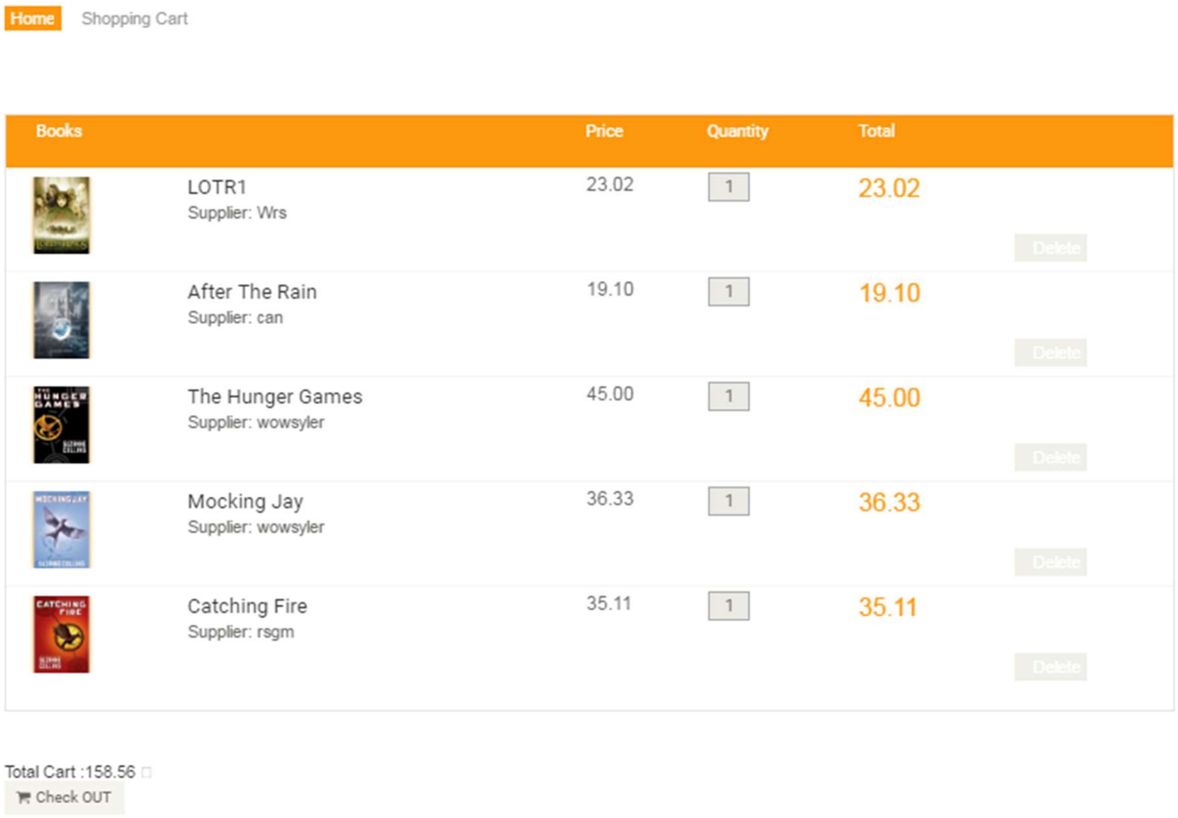
Short Description:

Discover The Links

Long Description:

Great ancient Ring thought lost for centuries has been found, and through a strange twist in fate has been given to a small Hobbit named Frodo.

Sec1 : You can see your account informations in My account section and also in that section you can update your informations. My cart section shows you your cart and you can see all the item which you added to cart before, total amount, price of books, quantity of books etc. And you can continue to checkout. Checkout section open a page that you can see book and total amount etc informations and fields to fill for shipping. Logout section log you out of account.
Sec2 : You can see the book's informations on that section.
Sec3 : You can select a different supplier for that book.
Sec4 : You can access to book's detail or supplier's information that sells that book or see the comment have done to that book.
Sec5 : You get the informations in here which is selected in section 4.



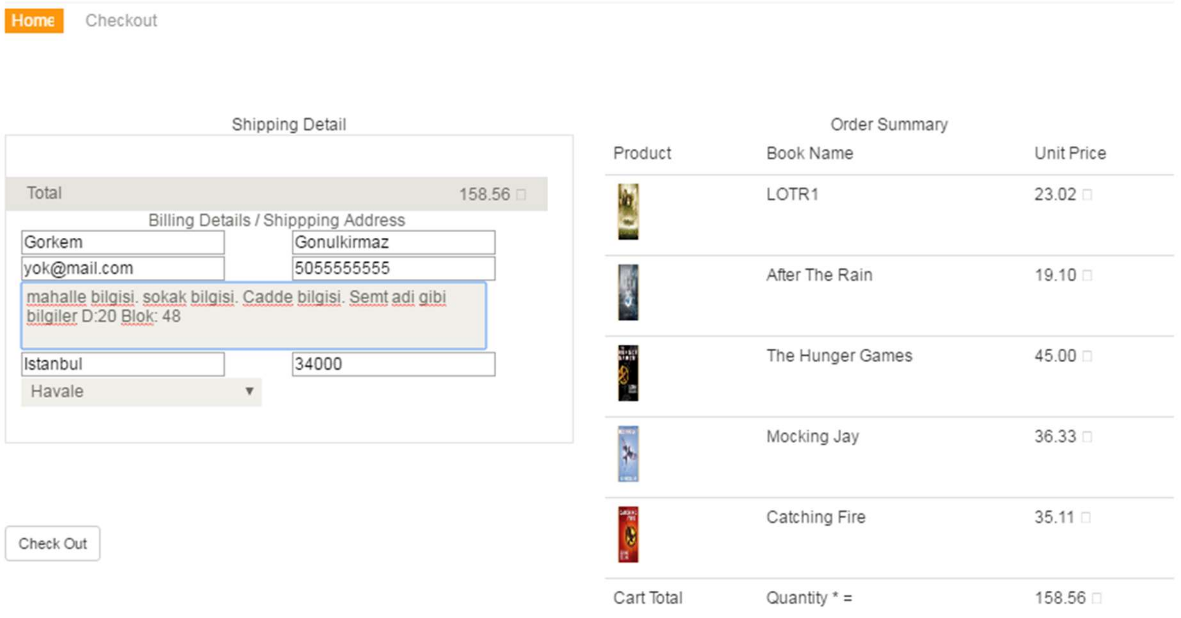
Home Shopping Cart

Books		Price	Quantity	Total	
	LOTR1 Supplier: Wrs	23.02	1	23.02	Delete
	After The Rain Supplier: can	19.10	1	19.10	Delete
	The Hunger Games Supplier: wowsyler	45.00	1	45.00	Delete
	Mocking Jay Supplier: wowsyler	36.33	1	36.33	Delete
	Catching Fire Supplier: rsgm	35.11	1	35.11	Delete

Total Cart :158.56 □

Check OUT

This is My Cart section, in here you can see the books you added to cart and informations about that books. And at the bottom you can see the total amount and checkout link.



Home Checkout

Shipping Detail

Total 158.56 □

Billing Details / Shippping Address

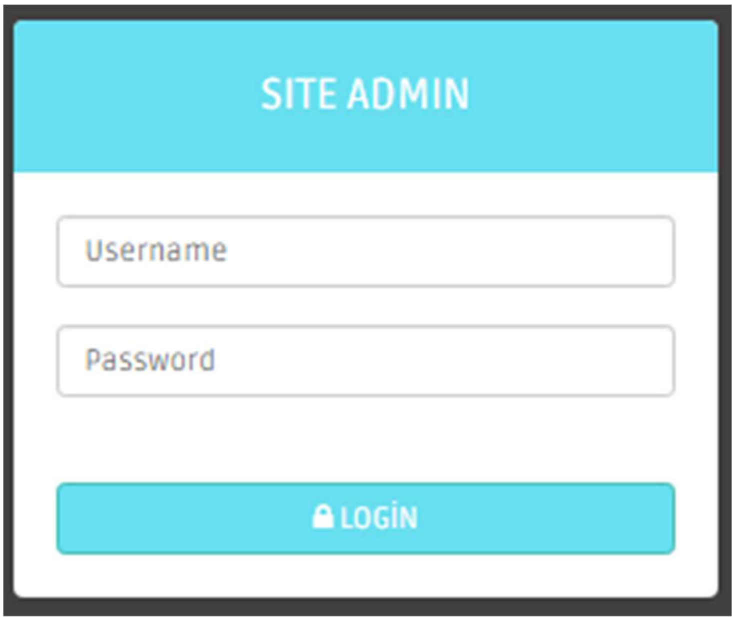
Gorkem	Gonulkirmaz
yok@mail.com	5055555555
mahalle bilgisi. sokak bilgisi. Cadde bilgisi. Semt adi gibi bilgiler D:20 Blok: 48	
Istanbul	34000
Havale	

Check Out

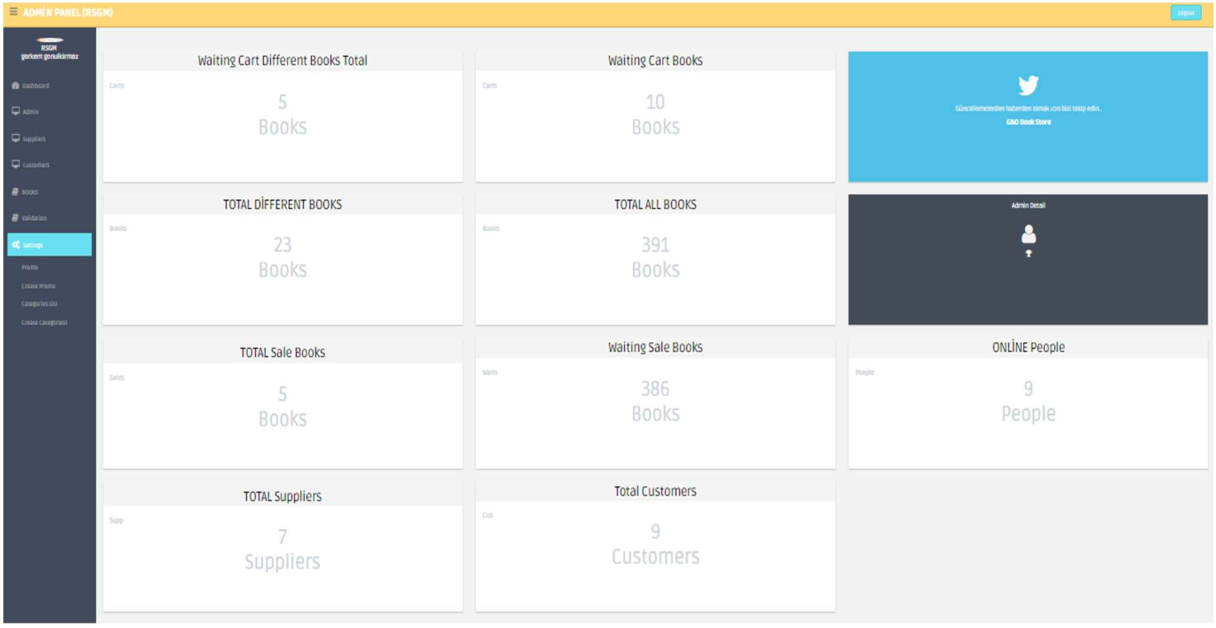
Order Summary

Product	Book Name	Unit Price
	LOTR1	23.02 □
	After The Rain	19.10 □
	The Hunger Games	45.00 □
	Mocking Jay	36.33 □
	Catching Fire	35.11 □
Cart Total	Quantity * =	158.56 □

This is Checkout section, in here you can see books and unit price of books. Total amount is shown in below of order summary. On the left you have shipping details. After you can click to check out button and your shipping process will be begin.



In admin page admin log in to system with use the username and pass to access admin panel.



Waiting Cart Different Books Total
Carts
5
Books

Waiting Cart Books
Carts
10
Books

TOTAL DİFFERENT BOOKS
Books
23
Books

TOTAL ALL BOOKS
Books
391
Books

TOTAL Sale Books
Sales
5
Books

Waiting Sale Books
Waits
386
Books

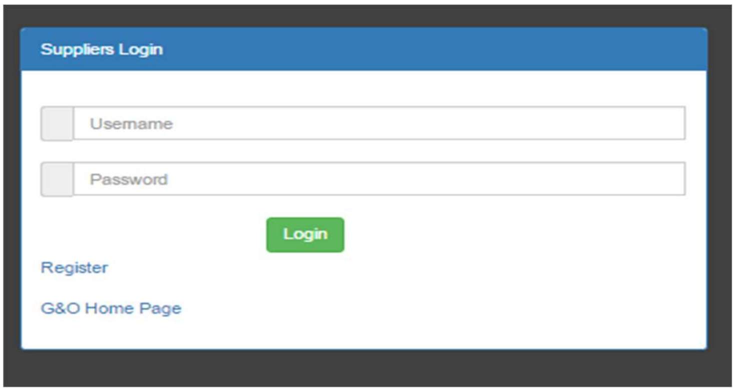
ONLİNE People
People
9
People

TOTAL Suppliers
Supp
7
Suppliers

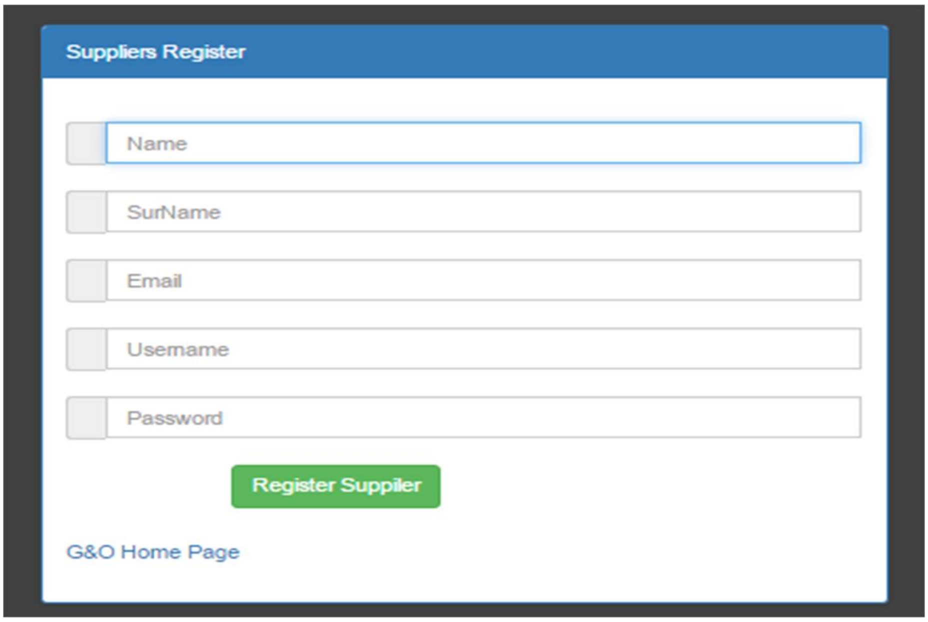
Total Customers
Cus
9
Customers

In dashboard admin get the informations shown in picture.

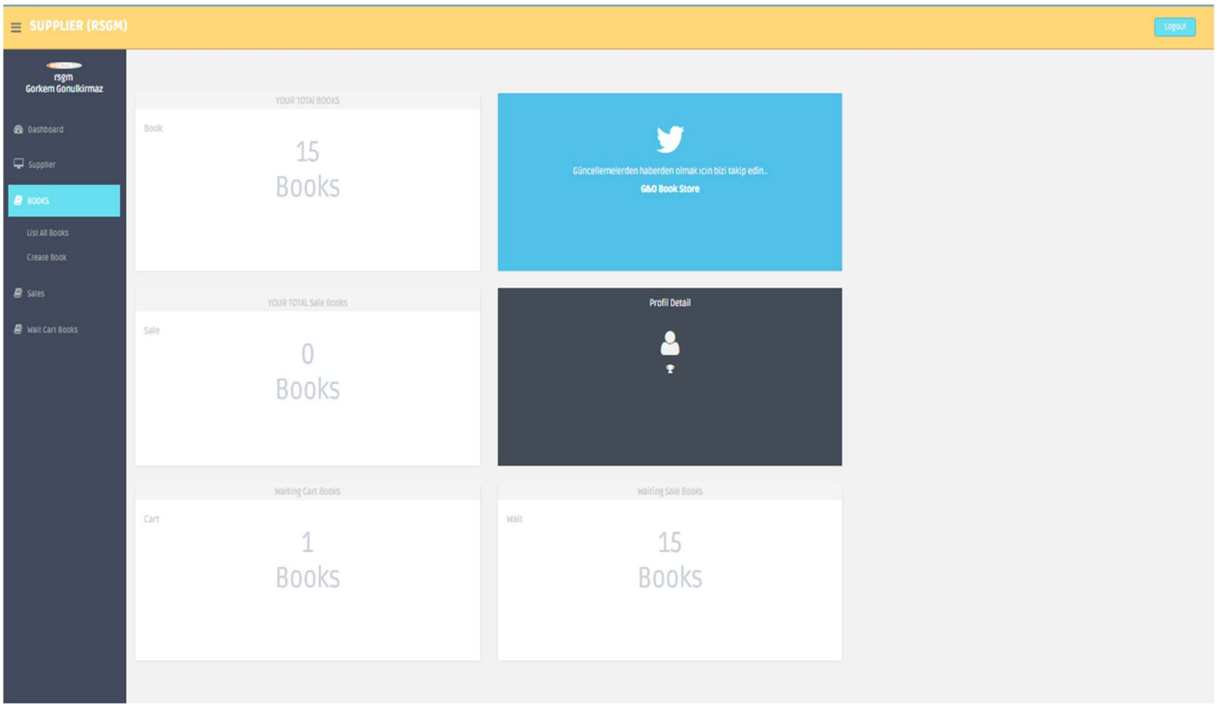
In the panel section admin have controls of web site which is presented in use case section.



This is Supplier Login page. If supplier registered to system they can login in that section and add their books to system in order to sell.



This is register page for supplier registration. Suppliers create a account to access supplier panel.



YOUR TOTAl BOOKS

Book

15 Books

Güncellemelerden haberden olmak icin bizi takip edin..

G&O Book Store

YOUR TOTAL Sale Books

Sale

0 Books

Waiting Cart Books

Cart

1 Books

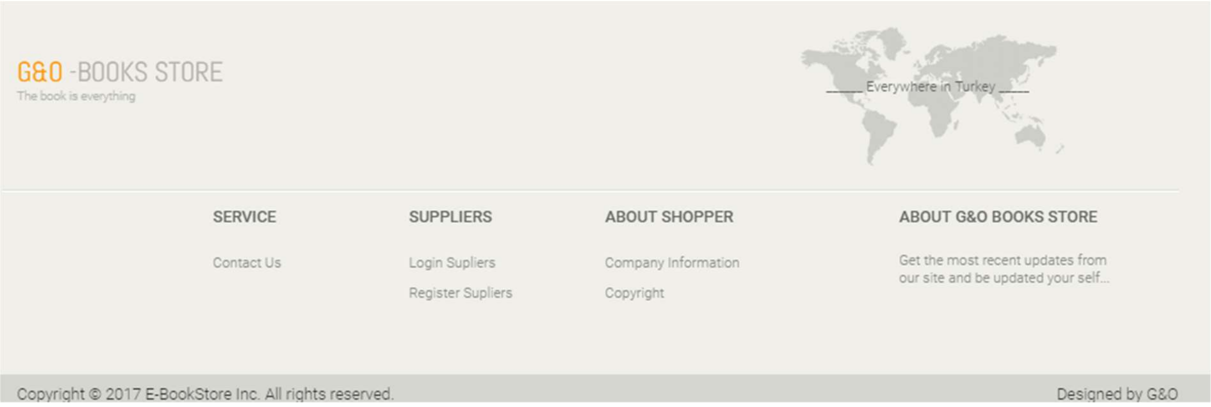
Waiting Sale Books

Wait

15 Books

This is supplier panel. Suppliers get small informations about their products on dashboard.

In panel section they have control on their account and products which is presented in use case section.



Get the most recent updates from our site and be updated your self...

Copyright © 2017 E-BookStore Inc. All rights reserved.

At the bottom of web site you see the footers section.

You can access to Contact page.

Also suppliers use that section in order to Login or Register to system.