|  |  |
| --- | --- |
| **Project Case** |  |
| COMP6114  Pattern Software Design |
| **Computer Science** | **E233-COMP6114-KZ04-01** |
| ***Valid on*** *Even Semester Year 2022/2023* | **Revision 00** |

1. Kelompok tidak diperkenankan untuk:

*Members of the group are prohibited from:*

* + - Melihat sebagian atau seluruh jawaban kelompok lain,

*Seeing a part or the whole answer from other groups,*

* + - Menyadur sebagian atau seluruh jawaban dari buku, catatan, video, dan jenis referensi lainnya,

*Retell a part or the whole answer from books, notes, videos, and other references,*

* + - Menyadur sebagian atau seluruh jawaban dari internet,

*Retell a part or the whole answer from the internet,*

* + - Mengumpulkan jawaban yang tidak sesuai dengan tema soal,

*Submitting an answer with a different theme from the given case,*

* + - Melakukan tindakan yang menyebabkan jawaban dicontek oleh orang lain atau kelompok lain, baik disengaja maupun tidak disengaja,

*Doing action that could result the answer being copied by someone or other groups, intentionally or unintentionally,*

* + - Melakukan tindakan kecurangan lainnya.

*Committing other dishonest actions.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dicantumkan pada butir ke-1, maka nilai mahasiswa dan/atau kelompok yang melakukan kecurangan, baik menyontek atau dicontek, akan dinolkan sesuai dengan peraturan yang berlaku.

*If it has been proven that a group has committed dishonest actions outlined in point 1 above, the whole groups related to the incident, regardless of which one copies or has their answer copied, will be issued a score of zero according to the regulation.*

1. Jawaban yang dapat diterima dan dinilai adalah jawaban yang dikumpulkan sebelum batas waktu yang telah ditentukan.

*The answer must be submitted before the designated deadline to be accepted and graded,*

1. Jawaban akan dinilai berdasarkan teknik atau metode yang diajarkan pada kelas praktikum dengan menggunakan software yang sudah ditentukan.

*The scoring will be based on the materials taught during the practicum classes using the designated software. Using different software than requested may result in your answer not being graded.*

1. Jika Anda tidak membaca peraturan ini, maka Anda dianggap sudah membaca dan menyetujuinya.

*By taking this exam, you agree to these regulations, regardless of whether you have read it or not.*

1. Persentase penilaian untuk matakuliah ini adalah sebagai berikut:

*The score will be distributed as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | 60% | - |

1. Perangkat lunak yang digunakan pada matakuliah ini adalah sebagai berikut:

*This course uses the following software:*

|  |
| --- |
| **Software**  *Software* |
| Chrome / Firefox / Microsoft Edge  Visual Studio 2019 (ASP.NET)  Visual Studio 2019 – SAP Crystal Reports |

1. Ekstensi file yang harus dikumpulkan untuk matakuliah ini adalah sebagai berikut:

*Your answers must be in the following file extensions:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| Folder Project (SLN, CS, ASPX, ASPX.CS, CONFIG, XML, DLL, PDB, RPT, XSS, XSD, SQL, MDF, LDF, EDMX, Image Files (JPG / PNG)) | Folder Project (SLN, CS, ASPX, ASPX.CS, CONFIG, XML, DLL, PDB, RPT, XSS, XSD, SQL, MDF, LDF, EDMX, Image Files (JPG / PNG)) | - |

1. File yang harus dikumpulkan adalah keseluruhan jawaban beserta dengan aset yang digunakan (gambar, audio, video, dll) dan dokumentasi proyek yang berisikan link referensi aset dan penjelasan mengenai aplikasi yang dibuat (terlampir bersama dengan soal).

*Include other files that can support your project, such as: all files in your project, other files (image, audio, video, etc.) used in your project, \*.doc file (documentation of your project) that contains all pages in your project, reference links of additional files (image, audio, video, etc.) used in your project, the description about how to use your application, etc.*

## Soal

*Case*

**KpopZtation**

**KpopZtation** is a K-pop album shop in Indonesia. To expand its business, **KpopZtation** wants to create a website for their shop. In that website, people can find information about the **K-pop album** and order it. This website enables the customers to know what is inside **KpopZtation**.

As a web developer, you are asked to create a website using **ASP.NET** for this online shop. You are also **required** to use **Domain Driven Design** method to develop the website. Please note that using web service is **optional** in this project. The **required** layers are:

* **View**

View layer, or Presentation Layer, is responsible for showing information to the user and interpreting the user's commands. This layer is the home for all user interfaces in the project.

* **Controller**

This layer is responsible to **validate** all input from the view layer. It also responsible for **delegating request**s from the user to the lower layer for further processing.

* **Handler**

This layer is responsible to handle all **business logic** required in the application. It will delegate the task to query from the database, including select, insert, update and delete, to the repository layer. Please notes that there can be a single handler that accesses multiple repositories.

* **Repository**

Repository layer responsible for **giving access** to the database and model layer via its public interfaces to acquiring references to preexisting domain objects. It provides methods to manipulate the object, such as add and delete, which will **encapsulate the actual manipulation operation** of data in the data store (or database). Repository also provides methods that select objects based on some criteria and return fully **instantiated** objects or collection of objects whose attribute meets those criteria.

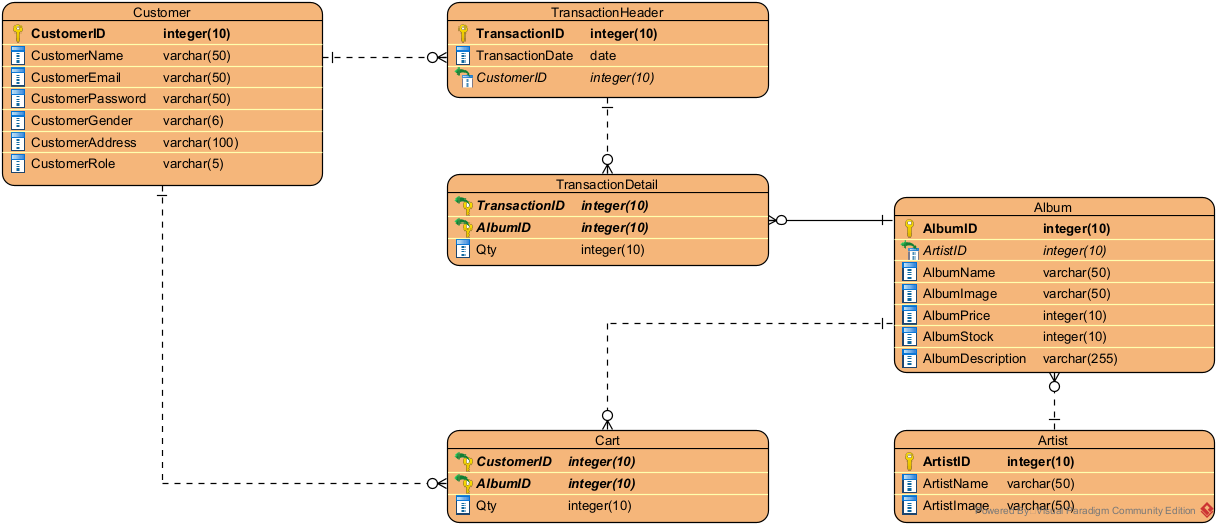
* **Factory**

You need to encapsulate all **complex object creation** in this layer. For example, when the client needs to create an aggregate object (an object that holds a reference to another object), the object factory must provide an interface for creating these objects. It is important to notes that an object returned by the factory must in **a consistent** state.

* **Model**

The model layer is responsible for **representing concepts** in the business or information about the business situation.

The following is the **Entity Relationship Diagram** for **KpopZtation’s Database:**



**Figure 1. Entity Relationship Diagram of KpopZtation’s Database**

There are three types of user's roles in this website: **Admin**, **Customer**, and **Guest** (**non-logged-in user**). Below are the minimum pages you need to create for each role:

* + - **Admin**
* View Artists
* Insert Artist
* Update Artist
* Delete Artis
* View Albums
* Insert Album
* Update Album
* Delete Album
* Update Profile
* Delete Account
* View Transactions Report
  + - **Customer**
* View Artists
* View Albums
* Order Album
* View Profile
* View Carts
* Update Profile
* Delete Account
* View Transactions History
* View Transaction Detail
  + - **Guest**
* Login
* Register
* View Artists
* View Albums

The descriptions for each page are as follows

* + - 1. **Login**

This page is the **landing page** of the website where the user will attempt to log in to the website. This page is only accessible to **guests**.

The following table shows the existing fields in the login form and validations for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Email** | Must be filled and appropriate with the data in the database |
| **Password** | Must be filled and appropriate with the data in the database |

If the user entered an **incorrect** username or password, then an **error message** will be shown corresponding to the validation triggering it. The login page will also have a **remember me** checkbox. The remember mecheckbox will save your login information by implementing a **cookie** which also has an **expired time** if a user had checked it so that the next time the user attempts another login, it will log the user in automatically.

1. **Register**

This page allows guests to **register** themselves as KpopZtation customer. Validate this page is only accessible to **guests**. Display an **error message** if the user inputs invalidpersonal data. The following table shows the existing fields on this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Name** | Must be filled and between 5 and 50 characters |
| **Email** | Must be filled and unique among the customer’s email |
| **Gender** | Must be selected |
| **Address** | Must be filled and ends with ‘Street’ |
| **Password** | Must be filled and alphanumeric |

1. **Navigation Bar**

**Provide a navigation bar to make it easier for user to navigate between pages:**

* + - **If no user is logged-in:**
  + **Home: Redirect user to the homepage**
  + **Login: Redirect user to the login page**
  + **Register: Redirect user to the register page**
    - **If the current user is a customer:**
  + **Home: Redirect user to the homepage**
  + **Cart: Redirect user to the carts page**
  + **Transaction: Redirect user to the transactions page**
  + **Update Profile: Redirect user to update profile page**
  + **Logout:** Logs out the user from the website and **destroys all cookie variables**
    - **If the current user is an admin:**
  + **Home: Redirect user to the homepage**
  + **Transaction: Redirect user to transaction report page**
  + **Update Profile: Redirect user to update profile page**
  + **Logout:** Logs out the user from the website and **destroys all cookie variables**

1. **Home**

This page allows user to see all KpopZtation’s **artists**. This page allows user to **redirect into Artists Detail Page if user click the card**. This page allows **admin** to **insert, update** and **delete** the artist. In this page, there is an **Insert Artist** button. If **admin** click **Insert Artist** button, then it will redirect them to **Insert Artist page.**

Each album in the lists have ‘**Update’** and ‘**Delete’** buttons:

* If **Admin** click **Delete** button, **delete** selected artist.
* If **Admin** click **Update** button, it will redirect them to **Update Artist page** and the selected Artist ID will be passed using the URL’s query string parameter.

1. **Artist Detail**

If user click the **Artist card** in homepage, redirect user to this details page. This page displays the **artist image, name**, and **album details owned by this artist** such as **album image**, **name, price** and **description**. This page allows user to redirect into **Album Detail Page if user click the card**. This page allows **admin** to **insert, update** and **delete** the album. If **admin** click **Insert Album** button, then it will redirect them to **Insert Album page.** Each album in the lists have **‘Update’** and **‘Delete’** buttons:

* If **Admin** click **Delete** button, **delete** selected **album**.
* If **Admin** click **Update** button, it will redirect them to **Update Album** page and the selected Artist ID will be passed using the URL’s query string parameter.

1. **Insert Artist**

This page is only accessible for **admin.** This page allows an **admin** to insert **a new artist** into the database, this page can be access when **admin** clicks **Insert Album** button in **Home** **Page**.Display **an** **error message** if the **admin** inputted invaliddata. When admin has inputted all of the fields correctly and pressedthe **Insert Button**, the inputted artist information will be **added** and **saved** into the database. The following table shows existing fields on this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Name** | Must be filled and unique among the artist’s name. |
| **Image** | Must be chosen, file extension must be .png, .jpg, .jpeg, or .jfif, and file size must be lower than 2MB. |

1. **Update Artist**

This page is only accessible for **admin.** This page allows an admin to **update** **an existing** artist in the database. The administrator will be redirected to this page when they clicked the **Update Button** provided in the **Home Page**. All the to-be-updated artist’s current data will be shown in this page and the administrator can **change** those data before submitting it again by pressing the Update Button. The artist’s data will be fetched from the database by using the artist’s ID provided from the URL’s query string parameter. When the admin has inputted all the fields correctly and pressed the Update Button, the artist’s information in the database will be **updated**. Display **an error message** if the admin inputted invaliddata.

The following table shows existing fields in the page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Name** | Must be filled and unique among the artist’s name. |
| **Image** | Must be chosen, file extension must be .png, .jpg, .jpeg, or .jfif, and file size must be lower than 2MB. |

1. **Insert Album**

This page is only accessible for **admin.** This page allows an **admin** to insert **a new album** for **the selected artist** on the artist detail page into the database, this page can be access when **admin** clicks **Insert Album** button in **Artist Detail Page**. **The album’s data will be fetched from the database by using the album’s ID provided from the URL’s query string parameter.** Display **an** **error message** if the **admin** inputted invaliddata. When admin has inputted all of the fields correctly and pressedthe **Insert Button**, the inputted artist information will be **added** and **saved** into the database. The following table shows existing fields on this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Name** | Must be filled and smaller than 50 characters. |
| **Description** | Must be filled and smaller than 255 characters. |
| **Price** | Must be filled and between 100000 and 1000000 |
| **Stock** | Must be filled and more than 0 |
| **Image** | Must be chosen, file extension must be .png, .jpg, .jpeg, or .jfif, and file size must be lower than 2MB. |

1. **Update Artist**

This page is only accessible for **admin.** This page allows an admin to **update** **an existing** album in the database. The administrator will be redirected to this page when they clicked the **Update Button** provided in the **Artist Detail Page**. All the to-be-updated album’s current data will be shown in this page and the administrator can **change** those data before submitting it again by pressing the Update Button. The artist’s data will be fetched from the database by using the album’s ID provided from the URL’s query string parameter. When the admin has inputted all the fields correctly and pressed the Update Button, the artist’s information in the database will be **updated**. Display **an error message** if the admin inputted invaliddata.

The following table shows existing fields in the page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Name** | Must be filled and smaller than 50 characters. |
| **Description** | Must be filled and smaller than 255 characters. |
| **Price** | Must be filled and between 100000 and 1000000 |
| **Stock** | Must be filled and more than 0 |
| **Image** | Must be chosen, file extension must be .png, .jpg, .jpeg, or .jfif, and file size must be lower than 2MB. |

1. **Album Detail**

This page is accessible for **all users**. If user click the **Details button** in homepage, redirect user to this details page. This page displays all the **album details** such as **album’s name, album’s description, album’s price,** and **album’s stock**. There is also a “**Add to Cart**” button. Validate the button to only appear if the user’s login as **Customer.** Customer could fill the **quantity** number before adding the item to cart.

The following table shows existing fields in the page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Quantity** | Must be filled and can’t be more than the stock album |

1. **Cart Page**

This page is accessible only for **customers**. The customer will be redirected to this page when they clicked the Cart Button provided at the Navigation Bar. This page displays all albums that have been added into customer’s shopping cart. Display cart information such as **album’s picture**, **album’s name**, **album’s quantity**, and **album’s** **price**. There are two buttons in this page, **Remove** button for each album on cart and **Checkout** button. If **customers** click **Remove** button, then **remove selected album** from the **cart**, and if **customers** click **Checkout** button, then **all shopping cart** **data** will be **deleted** and **insert** current transaction data into customer’s **transaction history**, then redirect user to home page.

1. **Transaction History**

This page is only accessible for **customers** to show **customer’s** transaction history. The information of transaction that must be shown are **transaction id, transaction date, customer’s name, courier, album’s picture**, **album’s name**, **album’s quantity**,and **album’s price**.

1. **Transaction Report**

This page is accessible only for the **admin**. The administrator will be redirected to this page when they clicked the Transactions Button provided at the Navigation Bar. This page will display the report transactions data using **Crystal Report**. The report will have to show the header transaction information such as **transaction id**, **customer id**, **transaction date** and **Grand Total value**, also the details such as **transaction id, album name, quantity, album price,** and **Sub Total value**.

1. **Update Profile**

This page allows the user to **change** their profile information. The user will be redirected to this page when they clicked the Update Profile Button provided at Navigation Bar. Display an error message if the user inputted invalid data. All of the to-be-updated profile's current data will be shown on this page and the user can change those data before submitting it again by pressing the Update Button. When the user has inputted all of the fields correctly and pressed the Update Button, the user’s information in the database will be **updated**. The following table shows existing fields in the page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Name** | Must be filled and between 5 and 50 characters |
| **Email** | Must be filled and unique among the customer’s email |
| **Gender** | Must be selected |
| **Address** | Must be filled and ends with ‘Street’ |
| **Password** | Must be filled and alphanumeric |