|  |  |
| --- | --- |
| **Project Case** |  |
| COMP6114 | COMP6114001  Pattern Software Design |
| **Computer Science** | **E252-COMP6114-LO02-00** |
| ***Valid on*** *Even Semester Year 2024/2025* | **Revision 00** |

1. Mahasiswa tidak diperkenankan untuk:

*Students are prohibited from:*

* + - Berdiskusi dan/atau bekerja sama dengan mahasiswa lainnya,

*Discussing and/or cooperating with other students,*

* + - Melihat sebagian atau seluruh jawaban mahasiswa lainnya,

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

* + - Membuka dan/atau menyalin jawaban dari buku, catatan, video, dan jenis referensilainnya,

*Open and/or copy answer from books, notes, videos, and other references,*

* + - Membuka dan/atau menyalin jawaban dari internet,

*Open and/or copy 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 mahasiswa dan/atau 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 student and/or group has committed dishonest actions outlined in point 1 above, the students and/or 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 yang diajarkan dalam praktikum dengan menggunakan software yang telah 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* |
| Visual Studio 2022  Visual Studio 2022 - SAP Crystal Reports  Web Browser (Google Chrome) |

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* |
| ASP.NET Folder Project (SLN, CS, ASPX, ASPX.CS, CONFIG, XML, DLL, PDB, RPT, XSS, XSD, SQL, MDF, LDF, EDMX, Asset Files) | ASP.NET Folder Project (SLN, CS, ASPX, ASPX.CS, CONFIG, XML, DLL, PDB, RPT, XSS, XSD, SQL, MDF, LDF, EDMX, Asset Files) | - |

## Soal

*Case*

**LOrd Card Shop**

**LOrd Card Shop** is an online platform dedicated to simplifying the buying and selling of collectible cards. The website offers features for users to purchase, sell, or simply explore information about their favorite cards. With an intuitive interface and efficient management system, **LOrd** Card Shop is the perfect solution for card collectors and enthusiasts.

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

* **View**

View 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 is 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** which will encapsulate the actual manipulation operation of data in the 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 note that an object returned by the factory must in aconsistent state.

* **Model**

The model layer is responsible for **representing concepts** in the business or information about the business situation. The model layer is handled using **entity** **framework** tool.

A diagram of a card

Description automatically generated

**Figure 1. Entity Relationship Diagram of Lord Card Shop Database**

There are **three types of user's roles** in this website, which are **admin**, **customer**, and **guest** (non-logged-in user). Below are the minimum **access** you need to create for each role.

* + - **Admin**
* **View Customers**
* **Insert Card**
* **Update Card**
* **Delete Card**
* **View Profile**
* **Update Profile**
* **View Unhandled Orders**
* **View Handled Orders**
* **Handle Orders**
* **Logout**
* **View Transactions Report**
  + - **Customer**
* **View Card List**
* **View Card Detail**
* **Order Card**
* **Add Card to Cart**
* **Clear Cart**
* **Checkout Card in Cart**
* **View Transactions History**
* **View Transaction Detail**
* **View Profile**
* **Update Profile**
* **Logout**
  + - **Guest**
* **Login**
* **Register**

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 |
| Username | 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 Lord Card Shop. Validate this page is only **accessible to guests**. Display an **error message** if the user inputs **invalid** personal data. The following table shows the existing fields on this page and the validation for each field:

|  |  |
| --- | --- |
| Field | Information |
| Username | Length must be between 5 and 30 and alphabet with space only. |
| Email | Length must contatin ‘@’. |
| Password | Length must be at least 8 characters and alphanumeric. |
| Gender | Must be chosen. |
| Confirmation Password | Must be the same with password. |
| Role | Default customer. |

* + - 1. **Navigation Bar**

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

* If the current user is a **customer**, then show:
* **Home**, redirect user to the home page.
* **Order Card**, redirect user to order card page.
* **Profile**, redirect user to update profile page.
* **History**, redirect user to transaction history page.
* **Logout**, logs out the user from the website and destroys all cookie.
* **Cart**, redirect user to cart page.
* If the current user is a **admin**, then show:
* **Home**, redirect user to the home page.
* **Manage Card**, redirect user to add show page.
* **View Transaction**, redirect user to transaction report page.
* **Transaction Report,** redirect user to view transaction report page.
* **Logout**, logs out the user from the website and destroys all cookie.
* **Order Queue**, redirect user to handle transaction page.

If there is a **user logged-in**, then **show the user’s name** which will be fetched using **session**; **otherwise** just show “**Guest**”.There will be a **search box** at the top of the page to **filter** shows based on its **name.** The filtering process mustbe implemented by using **the database filtering logic**. The filtered shows will be displayed in home page.

* + - 1. **Home**

**Display** a **welcome message** with the **user's username** when they visit this page.

* + - 1. **Order Card**

This page is **accessible only to customers**. This page displays **all card information**, including the name, price, provides an **Add to Cart** **button** and **Detail button for redirect to Card Detail Page**. When clicked, the button will **add the card to the user's shopping cart**.

* + - 1. **Card Detail**

This page is **accessible only to customers**. This page displays **all card information**, including the name, price, card type, card description and **Back button.**

* + - 1. **Cart**

This page is **accessible only to customers**. They will be redirected here after clicking the **cart button** in the **navigation bar**. On this page, customer can view all the **card**s they have added to their cart and proceed with the **checkout**. The displayed **card information** includes the show name, price, and description.

* + - 1. **Check Out**

This page is **accessible only to customers**. The customer will be redirected to this page when they clicked the **checkout button** provided at the **cart page**. On this page, buyers can view all the **card** they have added to their cart and proceed with the **checkout**. The displayed **card information** includes the show name, price, description, and **total price**.

* + - 1. **Transaction History**

This page is **accessible** for **both admin** and **customer**. If the user is a customer **displays all transaction** that the **user has made**, also provide a **button** **foreach** **transaction**, upon **clicking** **the** **button** the user will be **redirected** to the **detail** **page** **of** **the** **transaction**. If the user is an **admin** display **all** the **transaction** done also **providing** a **button** to **view** the **transaction** **detail using URL query string parameter**.

* + - 1. **Profile**

This page allows the user to **change** **their profile information**. The user will be redirected to this page when they clicked the **profile button** provided at **navigation bar**. Display **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 |
| Username | Length must be between 5 and 30 and alphabet with space only. |
| Email | Length must contatin ‘@’. |
| Password | Length must be at least 8 characters and alphanumeric. |
| Gender | Must be chosen. |

If **new password** field is filled, **validate**:

|  |  |
| --- | --- |
| Field | Information |
| Old Password | Must be the same password as the current password. |
| New Password | Length must be at least 8 characters and alphanumeric. |
| Confirmation Password | Must be the same with password. |

* + - 1. **Manage Card**

This page is only **accessible for** **admin.** This page displays **all cards in the database**, including the name, price, description, and provides a **delete** **button** and an **update button**. If the administrator clicks the delete button, the **card** will be removed from the database. Additionally, there is a button to **insert a new card**.

* + - 1. **Add Card**

This page is only **accessible for** **admin.** This page allows an administrator to **insert a new card** into the database. The user will be redirected to this page when they click the **add card button** provided at the **manage card page.** Display **error message** if the administrator inputted **invalid data**. When the user has inputted all the fields correctly and pressedthe **insert button**, the input show 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 | Length must be between 5 and 50 and alphabet with space only. |
| Price | Must be greater or equal than 10000. |
| Description | Description must not be empty. |
| Type | Type must be Spell or Monster |
| Foil | Foil must yes or no |

* + - 1. **Edit Card**

This page is only **accessible for** **admin.** This page allows an administrator to **insert a new show** into the database. The user will be redirected to this page when they click the **update button** provided at the **manage card page.** Display **error message** if the administrator inputted **invalid data**. When the user has inputted all of the fields correctly and pressedthe **update button**, the card data will be updated in the database. The following table shows existing fields on this page and the validation for each field:

|  |  |
| --- | --- |
| Field | Information |
| Name | Length must be between 5 and 50 and alphabet with space only. |
| Price | Must be greater or equal than 10000. |
| Description | Description must not be empty. |
| Type | Type must be Spell or Monster |
| Foil | Foil must yes or no |

* + - 1. **View Transactions Report**

This page is **accessible only by the admin**. The administrator will be redirected to this page when they click the **transactions button** provided at the **navigation bar**. This page will display the report of **all sales data** using **crystal report**. The report will have to show **grand total income** and each transaction’s **entire information** plus its **sub total** value.

* + - 1. **Handle Transaction**

This page is **only** **accessible for the Admin**. This page allows the user to view on going transactions that have and **have not been handled**. In this page the admin can also **handle** unhandled transaction by **pressing** the **handle transaction button**. Upon clicking the handle transaction button, update the status of the transaction from **unhandled** to **handled.**