|  |  |
| --- | --- |
| **Project Case** |  |
| COMP6114  Pattern Software Design |
| **Computer Science** | **E232-COMP6114-JE01-00** |
| ***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*

**RAAMEN**

**RAAMEN** is a web-based online ramen selling application. It is a great place for buyer to buy ramen and for seller to sell their shows. The CEO of **RAAMEN** wants to hire your team to build this application. Before working there, you are given an assignment to make a simple version of this application. You may find some features which don’t make sense. But remember, it is just an assignment to improve and test your skill.

As a web developer, you are asked to create a website using **ASP.NET** for this ramen 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 HBthe 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 delegates 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 repository.

* **Repository**

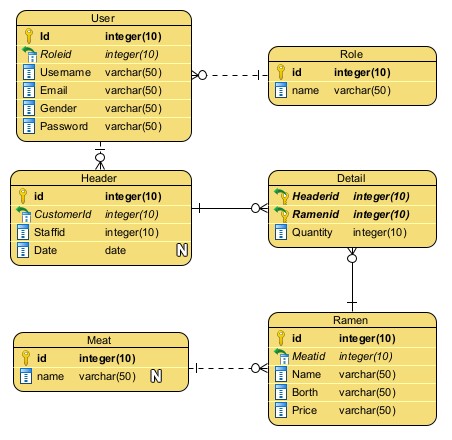
Repository layer is responsible for **giving access** to the database and model layer via its **public interfaces** to acquiring references to preexisting domain objects. It provides methods to **manipulate the object** which will encapsulate the actual manipulation operation of data in the database. 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**

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

.

**Figure 1. Entity Relationship Diagram of RAAMEN Database**

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

* + - **Staff**
* **View Customers (home page) (DAVID)**
* **Insert Ramen (manage ramen) (DAVID)**
* **Update Ramen (manage ramen) (DAVID)**
* **Delete Ramen (manage ramen) (DAVID)**
* **View Profile (profile) (DAVID)**
* **Update Profile (profile) (DAVID)**
* **View Unhandled Orders (order queue) (DAVID)**
* **View Handled Orders (order queue) (DAVID)**
* **Handle Orders (order queue) (DAVID)**
* **Login (login) (DAVID)**
* **Logout (navbar) (DAVID)** 
  + - **Member**
* **Ongoing order(status unhandled dari order yang dimiliki user tsbt) & view cart (home) (KEVIN)**
* **View Ramen (order ramen) (VINCENT)**
* **Order Ramen (order ramen) (VINCENT)**
* **Add Ramen Order to cart (order ramen) (VINCENT)**
* **Clear Cart (home) (KEVIN)**
* **Buy All Ramen in Cart (home) (KEVIN)**
* **View Transactions History (transaction (history)) (RIKA)**
* **View Transaction Detail (transaction (history)) (RIKA)**
* **View Profile (profile) (DAVID)**
* **Update Profile (profile) (DAVID)**
* **Login (login) (DAVID)**
* **Logout (navbar) (DAVID)** 
  + - **Admin**
* **Login (login) (DAVID)**
* **Register (register) (DAVID)**
* **View Customer (home) (DAVID)**
* **View Staffs (home) (DAVID)**
* **All Member & Staff Privilege (Sesuai navbar admin) (DAVID)**
* **View All Transaction (transaction (history)) (RIKA)**
* **View Report (report) (VINCENT)**
* **View Profile (profile) (DAVID)**
* **Update Profile (profile) (DAVID)**

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 RAAMEN buyers or seller. 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 15 and alphabet with space only. |
| Email | Must ends with ‘.com’. |
| Gender | Gender Must be chosen. |
| Password | Must be the same with confirm password. |
| Confirmation  Password | Must be the same with password. |

* + - 1. **Navigation Bar**

Provide a navigation bar after the user logged-in, to make it easier for user to **navigate between pages**.

* If the current user is a **customer**, then show:
* **Order Ramen**, redirect user to the Order Ramen page.
* **History**, redirect user to the transactions page.
* **Profile**, redirect user to update profile page.
* **Logout,** logs out the user from the website and destroys all cookies.
* If the current user is a **staff**, then show:
* **Home**, redirect user to the home page.
* **Manage Ramen**, redirect user to manage ramen page.
* **Order Queue**, redirect user to order queue page.
* **Profile**, redirect user to update profile page.
* **Logout**, logs out the user from the website and destroys all cookies.
* If the current user is an **admin**, then show:
* **Manage Ramen**, redirect user to manage ramen page.
* **Order Queue**, redirect user to order queue page.
* **Profile**, redirect user to update profile page.
* **History**, redirect user to the transactions page.
* **Report**, redirect user to report page.
* **Logout**, logs out the user from the website and destroys all cookies.
  + - 1. **Home**

When users visit this page, display the current user role, then if the logged in user is a **staff** **retrieve all the customer data** and **display** it in this page, and if the user is an **admin** also **retrieve all the staff data** and **display** it.

* + - 1. **Order Ramen**

This page is only accessible for customers. This page shows a **Ramen detail data** such as ramen name, meat, broth, price.

If the current user is **Customer** provide an **order button**. If the button is clicked, **add** the selected **RAAMEN** to the **cart** if the user selects **clear cart,** clear **all the cart contents**, if the user select **buy cart**. **Create a new unhandled transaction** for all ramen orders in the cart.

* + - 1. **Profile**

This page allows the user to **change** **their profile information**. The user will be redirected to this page when they click the **account button** provided at **navigation bar**. Display **error message** if the user inputted **invalid** data. All 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 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 15 and alphabet with space only. |
| Email | Must ends with ‘.com’. |
| Gender | Gender Must be chosen. |
| Password | Must be the same with the current password. |

* + - 1. **History**

This page is **not accessible** for **staff**. 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. **Transaction Detail**

**This page is not accessible for staff, in this page the user can see all the detailed data of the transaction selected.**

* + - 1. **Manage Ramen**

This page is **not** **accessible for** **Customer**. In this page the user will be presented with all the ramen data. In this page the staff can press the button to update and delete to alter the current ramen data upon clicking the update button. Redirect the user to update ramen page using URL query string parameter. The user can also click the insert button and be redirected to the insert ramen page. The following table shows existing fields in the page and the **validation** for each field:

* + - 1. **Insert Ramen**

This page is not accessible to the customer. This page will allow the user to input the contents and insert the new ramen data. Also provide a back button that upon clicking will redirect the user to Manage Ramen. The following table shows the existing fields in the page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| Name | Must contains ‘Ramen’. |
| Meat | Must be selected. |
| Broth | Cannot be Empty. |
| Price | Price must be at least 3000. |

* + - 1. **Update Ramen**

This page is not accessible to the customer. This page will allow the user to alter the contents and update the selected ramen data. Also provide a back button that upon clicking will redirect the user to Manage Ramen. The following table shows the existing fields in the page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| Name | Must contains ‘Ramen’. |
| Meat | Must be selected. |
| Broth | Cannot be Empty. |
| Price | Price must be at least 3000. |

* + - 1. **Transactions Queue (Order Queue)**

This page is not **accessible for** **Customer**. This page allows the user to view ongoing transactions that have and has not been handled by any staff or admin. In this page the user can also handle unhandled transaction by pressing the handle transaction button.

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

This page is **accessible only by the Admin**. The administrator will be redirected to this page when they click the **reports button** provided at the **navigation bar**. This page will display the report of **all RAAMAN 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.