|  |  |
| --- | --- |
| **Project Case** |  |
| COMP6114  Pattern Software Design |
| **Computer Science** | **E212-COMP6114-IR02-00** |
| ***Valid on*** *Even Semester Year 2020/2021* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + 1. Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from another groups*

* + 1. Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + 1. Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + 1. Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

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

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

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

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri, dan proyek untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| 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)) |

## Soal

*Case*

**NeinteenFlower**

**NeinteenFlower** is a newly established flower shop in Bluejack Land. To start its business, itneeds to create a website for their shop. In that website, people can find information about the flowers which **NeinteenFlower** sells and buy it. This website also allows employees to manage its users and flowers

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 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 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.

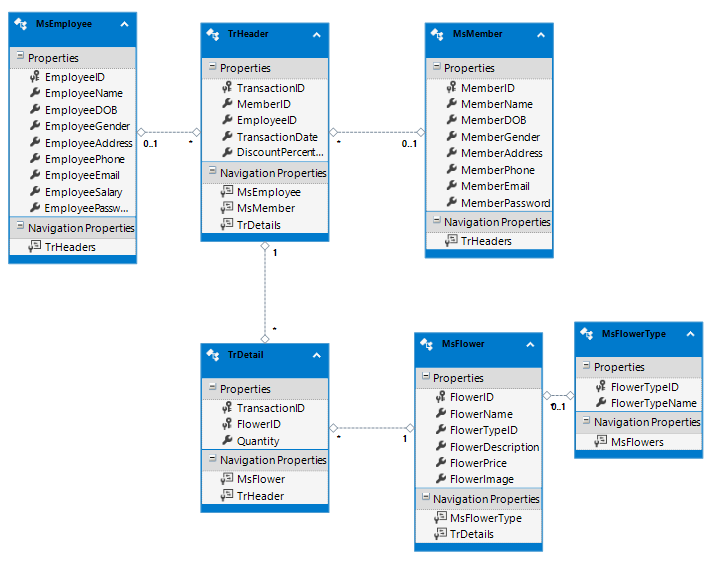
* **Factory**

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 **NeinteenFlower**’s **Database:**



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

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

* + - **Administrator**
* Manage Member
* Insert Member
* Update Member
* Manage Employee
* Insert Employee
* Update Employee
  + - **Member**
* Pre Order
* View Transaction History
  + - **Employee**
* Manage Flower
* Insert Flower
* Update Flower
  + - **Guest**
* Login
* Register
* Forgot Password

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** and willprovide a hyperlink to **Register Page** and **Forgot Password Page** at the bottomof the login form.

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 or  Must be the same as the captcha in forgot password page |

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 of one day** if a user had checked it so that the next time the user attempts another login, the **email** and **password** field will be **auto-filled** with the user’s previous login information, **fetched from the cookie**.

1. **Register**

This page allows guests to **register** themselves as **NeinteenFlower** members. Validate this page is only accessible to **guests**. The user will be redirected to this page when they clickedthe Register hyperlinkprovided at the Login Page.Display an error messageif the user inputs invalidpersonal data. The following table shows the existing fields on this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Email** | Must be unique and using a correct email format |
| **Password** | Minimal length is 3 characters and 20 characters is the maximal |
| **Name** | Minimal length is 3 characters and 20 characters is the maximal and  Must be letter |
| **Age** | Minimal age is 17 years old |
| **Gender** | Must be chosen |
| **Phone Number** | Must be numeric |
| **Address** | Must contain the word "Street" |

1. **Forgot Password**

This page allows guests to **reset** the password themselves. Validate this page is only accessible to **guests**. The user will be redirected to this page when they clickedthe Forgot Password hyperlinkprovided at the Login Page.Display an error messageif the user inputs invalidpersonal data. The following table shows the existing fields on this page and the validation for each field:

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Email** | Email must exist in database |
| **Captcha** | Random 3 letters and 3 numbers, example : abc123 |
| **New Password** | Must be the same as captcha |

1. **Home**

After success logged in, the user will be redirected to the home page and this page is only accessible by **logged-in users**.There will be a welcome message containing the user’s username which will be **fetched using session**,displayed at the top of the homepage.

This page will show some buttons to redirect the user to another page in the website. Every **logged-in** userwill have the following buttons available:

* **Home**:Redirects the user to Home Page
* **Logout**:Logs out the user from the website and **destroys all cookie variables**

If the user is a **member**, the following buttons will become visible:

* + **View Transaction History**: Redirects the user to View Transaction History
  + **Preorder**: Redirects the user to Preorder page

If the user is an **employee,** the following button will become visible:

* **Manage Flower**: Redirects the user to Manage Flower Page

If the user is an **admin**, the following button will become visible:

* **Manage Member:** Redirects the user to Manage Member Page
* **Manage Employee:** Redirects the user to Manage Employee Page

User that has member role will also be shown all flowers from the database, which will also have a **Pre Oder** button at each table row**.** The Pre Order Button will redirect the user to Pre Order Page and the selected flower’s ID will be passed using the URL’s query string parameter.

1. **PreOrder**

This page is only accessible for **member.** This page allows a **member** to preorder the flower they choose from Home Page. The user will be redirected to this page when they clicked the Pre Order Buttonprovided at the Home Page**.** Display error message if the member inputted invaliddata. When the user has inputted all of the fields correctly and pressedthe Preorder button, the inputted transaction 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** |
| **Quantity** | Must be between 1 to 100 inclusively |

1. **Manage Flower**

This page is only accessible for **employee.** This page allow an employee to manage all flower exist in the database. The user will be redirected to this page when they clicked Manage Flower Button provided at Home Page. On this page, there will provide Insert Flower hyperlink. Users will be shown all flowers from database, which will also have Update button and Delete button at each table row**.** The Update Button will redirect the user to Update Flower Page and the selected flower’s ID will be passed using the URL’s query string parameter. The Delete Button will delete the flower from database.

1. **Insert Flower**

This page is only accessible for **employee.** This page allows an employee to insertanew flower into the database. The user will be redirected to this page when they clicked the Insert Flower hyperlink provided at the Manage Flower Page**.** Display error message if the employee inputted invaliddata. When the user has inputted all of the fields correctly and pressedthe Insert Button, the inputted flower 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 minimal length is 5 characters |
| **Image** | Extension must ends with “.jpg” and save the path into database (the path must not be an absolute path) |
| **Description** | Must be longer than 50 characters |
| **Flower Type** | Must be either "Daisies", "Lilies" or "Roses" |
| **Price** | Must be numeric and between 20 and 100 inclusively |

1. **Update Flower**

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

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

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Name** | Must be filled and minimal length is 5 characters |
| **Image** | Extension must ends with “.jpg” and save the path into database (the path must not be an absolute path) |
| **Description** | Must be longer than 50 characters |
| **Flower Type** | Must be either "Daisies", "Lilies" or "Roses" |
| **Price** | Must be numeric and between 20 and 100 inclusively |

1. **Manage Member**

This page is only accessible for **administrator.** This page allow an administrator to manage all member exist in the database. The user will be redirected to this page when they clicked Manage Member Button provided at Home Page. On this page, there will provide Insert Member hyperlink. Users will be shown all members from database, which will also have Update button and Delete button at each table row**.** The Update Button will redirect the user to Update Member Page and the selected member’s ID will be passed using the URL’s query string parameter. The Delete Button will delete the member from database.

1. **Insert Member**

This page is only accessible for **administrator.** This page allows an administrator to insertanew member into the database. The user will be redirected to this page when they clicked the Insert Member hyperlink provided at the Manage Member Page**.** Display error message if the administrator inputted invaliddata. When the user has inputted all of the fields correctly and pressedthe Insert Button, the inputted member 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** |
| **Email** | Must be unique and using a correct email format |
| **Password** | Minimal length is 3 characters and 20 characters is the maximal |
| **Name** | Minimal length is 3 characters and 20 characters is the maximal and  Must be letter |
| **Age** | Minimal age is 17 years old |
| **Gender** | Must be chosen |
| **Phone Number** | Must be numeric |
| **Address** | Must contain the word "Street" |

1. **Update Member**

This page is only accessible for **administrator.** This page allows an administrator to **update** **an existing** member in the database. The administrator will be redirected to this page when they clicked the Update Member Button provided at the Manage Member Page. All of the to-be-updated member’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 member’s data will be fetched from the database by using the member’s ID provided from the URL’s query string parameter. When the user has inputted all of the fields correctly and pressed the Update Button, the member’s information in the database will be **updated**. Display error message if the administrator inputted invaliddata.

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

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Email** | Must be unique and using a correct email format |
| **Password** | Minimal length is 3 characters and 20 characters is the maximal |
| **Name** | Minimal length is 3 characters and 20 characters is the maximal and  Must be letter |
| **Age** | Minimal age is 17 years old |
| **Gender** | Must be chosen |
| **Phone Number** | Must be numeric |
| **Address** | Must contain the word "Street" |

1. **Manage Employee**

This page is only accessible for **administrator.** This page allow an administrator to manage all employee exist in the database. The user will be redirected to this page when they clicked Manage Employee Button provided at Home Page. On this page, there will provide Insert Employee hyperlink. Users will be shown all employees from database, which will also have Update button and Delete button at each table row**.** The Update Button will redirect the user to Update Employee Page and the selected employee’s ID will be passed using the URL’s query string parameter. The Delete Button will delete the employee from database.

1. **Insert Employee**

This page is only accessible for **administrator.** This page allows an administrator to insertanew employee into the database. The user will be redirected to this page when they clicked the Insert Employee hyperlink provided at the Manage Employee Page**.** Display error message if the administrator inputted invaliddata. When the user has inputted all of the fields correctly and pressedthe Insert Button, the inputted employee 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** |
| **Email** | Must be unique and using a correct email format |
| **Password** | Minimal length is 3 characters and 20 characters is the maximal |
| **Name** | Minimal length is 3 characters and 20 characters is the maximal and  Must be letter |
| **Age** | Minimal age is 17 years old |
| **Gender** | Must be chosen |
| **Phone Number** | Must be numeric |
| **Address** | Must contain the word "Street" |
| **Salary** | Must be numeric and between 100 to 1000 inclusively |

1. **Update Member**

This page is only accessible for **administrator.** This page allows an administrator to **update** **an existing** employee in the database. The administrator will be redirected to this page when they clicked the Update Employee Button provided at the Manage Employee Page. All of the to-be-updated employee’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 employee’s data will be fetched from the database by using the employee’s ID provided from the URL’s query string parameter. When the user has inputted all of the fields correctly and pressed the Update Button, the employee’s information in the database will be **updated**. Display error message if the administrator inputted invaliddata.

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

|  |  |
| --- | --- |
| **Field** | **Information** |
| **Email** | Must be unique and using a correct email format |
| **Password** | Minimal length is 3 characters and 20 characters is the maximal |
| **Name** | Minimal length is 3 characters and 20 characters is the maximal and  Must be letter |
| **Age** | Minimal age is 17 years old |
| **Gender** | Must be chosen |
| **Phone Number** | Must be numeric |
| **Address** | Must contain the word "Street" |
| **Salary** | Must be numeric and between 100 to 1000 inclusively |

1. **View Transactions History**

This page is accessible only for **member**. The user will be redirected to this page when they clicked the View Transactions History Button at the Home Page. This page will display the report of all member’s transaction in **NeinteenFlower** using **Crystal Report**. The report will have to show each transaction’s **entire information** with **Sub Total** and **Grand Total** value

**If you do not understand, please ask your assistant!**