|  |  |
| --- | --- |
| **Project Case** | Diagram  Description automatically generated |
| ISYS6197 | ISYS6197003  Business Application Development |
| **Computer Science** | **O232-ISYS6197-KZ10-00** |
| ***Valid on*** *Odd Semester Year 2022/2023* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + - Melihat sebagian atau seluruh proyek kelompok lain,

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

* + - Menyadur sebagian maupun seluruh proyek dari buku,

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

* + - Mendownload sebagian maupun seluruh proyek dari internet,

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

* + - 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,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - 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* |
| Eclipse 2020.6  Java 11  Java FX  MySQL Java Connection Library 8.0.24  XAMPP 8.0.7 |

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

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

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| JAVA, CLASS | JAVA, CLASS, SQL | - |

## Soal

*Case*

**KpopZtation**

**KpopZtation** is a shop that sells K-pop album. To manage all the transactions and manage system of the album, KpopZtation’s owner asked you to create the program for the store. The program should be made using **Java Programming Language**.

There are some requirements to build this application:

Diagram

Description automatically generated

Execute the provided create and insert query file on “**kpopztation**” database! (“create+insert.sql”)

1. **Login Form**

This is the first form which will appear when the program is run. **Login** **Form** is used by users to **Login** so the users can access the **Main** **Form**.

Graphical user interface, application

Description automatically generated

Figure 1. Login Form

**Components**:

* **Label** for Title, Email, and Password.
* **TextField** for input Email.
* **PasswordField** for input Password.
* **Button** for Register and Submit.

**Description**:

* If user clicks **Submit** button, then **validate**:
  + **Email** field **must** **be** **filled**.
  + **Password** field **must** **be** **filled**.
  + **Email** and **Password** **must** be **correct** according to the **existing data in database**.
* If user has successfully signed in, program will **validate** either its role is “**User**” or “**Admin**”. Lastly, the **Login** **Form** will be closed, and **Main** **Form** will be displayed (Figure 6 or Figure 7 depends on user’s role).
* If user failed to login then display warning message.
* If user click “**Register**” button, then program will close **Login** **Form** and open **Register** **Form**.

Graphical user interface, application

Description automatically generated

Figure 2. Failed Login Warning

(**additional** **note**: **every** **error** occurrence will be shown on error message **Alert**).

1. **Register Form**

**Register Form** is used by users to **Register**.

Graphical user interface, application, email

Description automatically generated

Figure 3. Register Form

**Components**:

* **Label** for Title, Username, Email, Password, Gender, Address, and Sign In.
* **TextField** for Username and Email.
* **PasswordField** for input Password.
* **RadioButton** for input Gender (“**Male**” or “**Female**”).
* **TextArea** for input Address.
* **Button** for Reset, Register, and Login.

**Description**:

* If user clicks **Register** button, then **validate**:
  + **ID is generated** when Register Form is opened with **format**:

**USXXX**

Where **X**: Random Digit (0-9)

Example: US001, US123, US999

* + **Username** must be between **5 - 20 characters**.
  + **Email** must be in valid **format**:

**[email]@[provider].com**

* + - Character ‘@’ must **not be next to** ‘.’.
    - **Must not starts** and **ends** with ‘@’ nor ‘.’.
    - **Must contain** **exactly** **one** ‘@’.
    - **Must contain** **exactly** **one** ‘.’ after ‘@’ for separating [**provider**] and “**com”**.
    - **Must ends with** ‘.com’

Graphical user interface, text, application

Description automatically generated

Figure 4. Invalid Email Format Warning

(Example: Character ‘@’ must not be next to ‘.’.)

* + **Password** must **5 - 20 length of character**.
  + **Gender** must be selected either “**Male**” or “**Female**”.
  + **Address** must **ends with** ‘ Street’.

(**additional** **note**: **every** **error** occurrence will be shown on error message **Alert**).

* If user successfully register, **insert** new **user** data **to database**, then program will display a success insert message. Lastly, program will close **Register** **Form** and open **Login** **Form (Figure 1)**.

Graphical user interface, application

Description automatically generated

Figure 5. Register Success Message

1. **Main Form**

This is the main form for the program. At this form the user can select menu from the menu bar, such as **Transaction** **Menu** (**Buy Album** and **View Transaction**), **Manage** **Menu** (**Album** and **Artist**), and **Logout Menu** for **Logout**. The menu that will show is determined by the user role. If the user role is “**User**”, then the menus that available are **Transaction** **Menu** and **Logout Menu** only. If the user role is “**Admin**”, then the menus that available are **Manage** **Menu, Transaction Menu** and **Logout Menu**.

Graphical user interface, application, Word

Description automatically generated

Figure 6. User Main Form

Graphical user interface, application, Word

Description automatically generated

Figure 7. Admin Main Form

**Components**:

* **User** Role:
  + **Menu** for Menu Transaction and Logout.
  + **MenuItem** for Buy Album and View Transaction.
* **Admin** Role:
  + **Menu** for Menu Manage, Menu Transaction and Logout.
  + **MenuItem** for Album, Artist, and View Transaction.

**Description**:

There are some **MenuItem** that available (**to be clicked**):

* **Both** Role
  + If user clicks **Logout** Menu Item, then the program will close **Main Form** and open **Login Form**.
* **User** Role

There will be two **Menu** displayed:

* + - * **Transaction**

Graphical user interface, text, application

Description automatically generated

Figure 8. User: Transaction Menu

There are 2 **MenuItem** available (**to be clicked**) in **Transaction Menu** (**Buy Album** and **View Transaction**).

* + - * + If user click **Buy Album**, program will open **Buy Album Form** inside the **Main Form**.
        + If user click **View Transaction**, program will open **Transaction Form** inside the **Main** **Form**.
      * **Logout**
        + If user clicks **Logout**, program will close Main Form and open Login Form.
* **Admin** Role

There will be three **Menu** displayed:

* + - * **Manage**

Graphical user interface, application

Description automatically generated

Figure 9. Admin: Manage Menu

There are 2 **MenuItem** available (**to be clicked**) in **Manage** **Menu** (**Artist and Album**).

* + - * + If user clicks **Artist**, program will open Manage Artist Form inside the Main Form.
        + If user clicks **Album**, program will open Manage Album Form inside the Main Form.
      * **Manage**

Graphical user interface, application

Description automatically generated

Figure 10. Admin: Transaction Menu

There are 1 **MenuItem** available (**to be clicked**) in **Transaction** **Menu** (**View Transaction**).

* + - * + If user clicks **View Transaction**, program will open **View Transaction** **Form** inside the **Main** **Form**.
      * **Logout**
        + If user clicks **Logout**, program will close Main Form and open Login Form.

1. **Buy Album Form**

This form **displays all the currently available album**. In this form, user can click any album to see the album detail and add it to cart. In this form, cart is also shown at the bottom view of the form. This form is made with **Window**.

Graphical user interface, application

Description automatically generated

Figure 11. Buy Album Form

**Components**:

* **TableView** for Album Table and Cart Table.
* **Label** for Album Title, Album ID, Album Name, Artist Name, Quantity, and Cart Title.
* **TextField** (**read**-**only**) for Album ID, Album Name, and Artist Name.
* **Spinner** for input Quantity.
* **Button** for Add to Cart and Checkout.

**Description**:

User can **only edit quantity field** and the **other fields can’t be edited**. If user clicks **Add to Cart** button, program will **validate** is there any **album chosen** or not. If the album already chosen, then program will **validate** the **quantity** that **can’t be less than 1** and **can’t be more than its Album Stock**.

Graphical user interface, application

Description automatically generated

Figure 12. Choose Album Warning

Graphical user interface, application

Description automatically generated

Figure 13. Quantity Warning

If all conditions are fulfilled, then the album will be added to the user’s cart:

* If the album **hasn’t existed yet**, **insert** new **album** to user’s cart and show message.

Graphical user interface, application

Description automatically generated

Figure 14. Album Added Message

* If user clicks **Checkout** button, then program will **clear the cart** and **insert all album** to **Transaction** (**Header** and **Detail**) and shows message.

Graphical user interface, text, application

Description automatically generated

Figure 15. Checkout Success Message

(**additional** **note**: **every** **error** occurrence will be shown on error message **Alert**).

1. **View Transaction**

**View Transaction** allows user and admin to **view all the transaction history**. This form is made with **Window**.

Table

Description automatically generated

Figure 16. View Transaction

**Components**:

* **Label** for Transaction List Title and Transaction Detail Title.
* **TableView** for Header Transaction Table and Detail Transaction Table.

**Description**:

If user **clicks** one of the **header transaction table**, it will **show** all the **detail** **transaction** that user bought on that transaction.

Graphical user interface, table

Description automatically generated

Figure 17. View Transaction User

For **Admin**, it will also show 2 tables that are already explain above, but the difference is between Admin and User that, Admin will see **all Transaction** that has been made by **all User**.

Graphical user interface, table

Description automatically generated

Figure 18. View Transaction Admin

1. **Manage Album Form**

**Manage** **Album** **Form** is made for admin to manage album. Admin can do **insert** new album, **update** any existing album data, and **remove**/**delete** album. This form is made with **Window**.

Table

Description automatically generated

Figure 19. Manage Album Form

**Components**:

* **Label** for Title, Album ID, Album Name, Album Price, Album Stock, and Artist Name.
* **TableView** for Album Table.
* **TextField** for input Album ID (**read**-**only**), Album Name, and Album Price.
* **ComboBox** for choose Artist Name.
* **Spinner** for input Album Stock.
* **Button** for Insert, Update, Delete, Submit, and Cancel.

**Description**:

* All the fields can be edited **except** for Album ID.
* Admin can do **Insert**, **Update**, and **Delete** album from database.
  + If admin clicks **Insert** button, then **validate**:
    - Admin can only click, **Submit** or **Cancel**.
    - Admin can input **All the input field** except **Album ID, Album ID** will be auto increment.
    - **Album Name** must consist of **5 - 30 characters**.
    - **Album** **Price** must be **numeric** and **more than 0**.
    - **Album Stock** must **more than 0.**
    - **Artist Name** must be **chosen**.
    - If Admin click **Cancel Button**, then it will make them go to **initial state of the Manage Album Form (Figure 19).**

Graphical user interface, table

Description automatically generated

Figure 20. Insert Button Click

* + - If Admin click **Submit Button**, then it will **insert the values to the database**, and show a message:

Graphical user interface, application

Description automatically generated

Figure 21. Successfully Insert Data

* + - Then the table will be **refresh and shown the new inserted data.**
  + If admin clicks **Update** button, then **validate**:
    - Admin can only click, **Submit** or **Cancel**.
    - **Album** **ID** is **not** **empty** (by click a row in album table).

Graphical user interface, application

Description automatically generated

Figure 22. Choose Album Warning

* + - **Album Name** must consist of **5 - 30 characters**.
    - **Album** **Price** must be **numeric** and **more than 0**.
    - **Album Stock** must **more than 0.**
    - **Artist Name** must be **chosen**.
    - If Admin click **Cancel Button**, then it will make them go to **initial state of the Manage Album Form (Figure 19).**

Graphical user interface, table

Description automatically generated

Figure 23. Update Button Click

* + - If Admin click **Submit Button**, then it will **update the values to the database**, and show a message:

Graphical user interface, application

Description automatically generated

Figure 24. Successfully Update Data

* + - Then the table will be **refresh and shown the new updated data.**
  + If admin clicks **Remove** button, then **validate**:
    - Admin can only click, **Submit** or **Cancel**.
    - **Album** **ID** is **not** **empty** (by click a row in album table).
    - If Admin click **Cancel Button**, then it will make them go to **initial state of the Manage Album Form (Figure 19).**
    - If Admin click **Submit Button**, then it will **delete the data from the database**, and it will show message:

Graphical user interface, application

Description automatically generated

Figure 25. Successfully Insert Data

* + - Then the table will be **refresh and shown the all the data after deletion**.

(**additional note**: **every error** occurrence will be shown on error message Alert).

1. **Manage Artist Form**

**Manage** **Artist** **Form** is made for admin to manage artist. Admin can do **insert** new artist, **update** any existing artist data, and **remove**/**delete** artist. This form is made with **Window**.

Graphical user interface

Description automatically generated

Figure 26. Manage Artist Form

**Components**:

* **Label** for Title, Artist ID, and Artist Name.
* **TableView** for Artist Table.
* **TextField** for input Artist ID (**read**-**only**) and Artist Name.
* **Button** for Insert, Update, Delete, Submit, and Cancel.

**Description**:

* All the fields can be edited **except** for Artist ID.
* Admin can do **Insert**, **Update**, and **Delete** artist from database.
  + If admin clicks **Insert** button, then **validate**:
    - Admin can only click, **Submit** or **Cancel**.
    - Admin can input **All the input field** except **Artist ID, Artist ID** will be auto increment.
    - **Artist Name** must consist of **5 - 20 characters**.
    - If Admin click **Cancel Button**, then it will make them go to **initial state of the Manage Artist Form (Figure 26).**

Graphical user interface, application

Description automatically generated

Figure 27. Insert Button Click

* + - If Admin click **Submit Button**, then it will **insert the values to the database**, and show a message:

Graphical user interface, application

Description automatically generated

Figure 28. Successfully Insert Data

* + - Then the table will be **refresh and shown the new inserted data.**
  + If admin clicks **Update** button, then **validate**:
    - Admin can only click, **Submit** or **Cancel**.
    - **Artist** **ID** is **not** **empty** (by click a row in album table).

Graphical user interface, application

Description automatically generated

Figure 29. Choose Artist Warning

* + - **Artist Name** must consist of **5 - 20 characters**.
    - If Admin click **Cancel Button**, then it will make them go to **initial state of the Manage Artist Form (Figure 26).**

Graphical user interface, application

Description automatically generated

Figure 30. Button Update Click

* + - If Admin click **Submit Button**, then it will **update the values to the database**, and show a message:

Graphical user interface, application

Description automatically generated

Figure 31. Successfully Update Data

* + - Then the table will be **refresh and shown the new updated data.**
  + If admin clicks **Remove** button, then **validate**:
    - Admin can only click, **Submit** or **Cancel**.
    - **Artist** **ID** is **not** **empty** (by click a row in album table).
    - If Admin click **Cancel Button**, then it will make them go to **initial state of the Manage Artist Form (Figure 26).**
    - If Admin click **Submit Button**, then it will **delete the data from the database**, and it will show message:

Graphical user interface, application

Description automatically generated

Figure 33. Successfully Delete Data

* + - Then the table will be **refresh and shown the all the data after deletion**.

(**additional note**: **every error** occurrence will be shown on error message Alert).

Here are the **rules** that you must follow to create your project:

1. Use **appropriate software** for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
2. Use the **techniques taught during practicum**
3. Collect **appropriate files** for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya.
4. Include the **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 the reference links of additional files (image, audio, video, etc.) used in your project.

**If you do not understand, please ask your assistant! Do not make your own assumption!**