|  |  |
| --- | --- |
| **Project Case** |  |
| COMP6225  Object-Oriented Database |
| **Computer Science** | **C182-COMP6225-DO01-00** |
| ***Valid on*** *Even Semester Year 2018/2019* | **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 other 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. Jangan lupa untuk melihat kriteria penilaian proyek yang ditempel di papan pengumuman, atau tanya asisten anda.

*Don’t forget to look at the project assessment criteria that posted on the announcement board, or ask your teaching assistant.*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

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

|  |  |
| --- | --- |
| **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* |
| Microsoft Visual C# 2010 Express |

## 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:*

|  |  |
| --- | --- |
| **Proyek**  *Project* | **UAP**  *Final Exam* |
| Folder Project (SLN, CS, MDF, LDF, RESX, EDMX, CSPROJ, USER, EXE, PDB, MANIFEST, CACHE, TLOG, TXT, RESOURCES, CSDL, MSL, SSDL, DLL, SETTINGS, image files(JPG/PNG/GIF)) | Folder Project (SLN, CS, MDF, LDF, RESX, EDMX, CSPROJ, USER, EXE, PDB, MANIFEST, CACHE, TLOG, TXT, RESOURCES, CSDL, MSL, SSDL, DLL, SETTINGS, image files(JPG/PNG/GIF)) |

## Soal

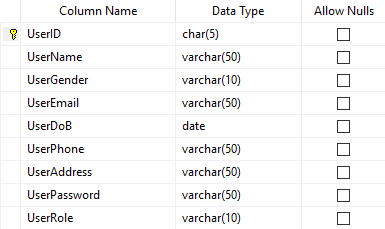
*Case*

**Discover O’laptop**

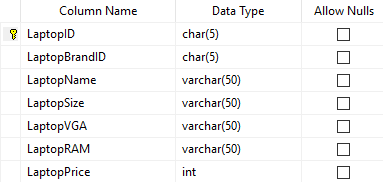
**Discover O’laptop** is a laptop store based in Jakarta, Indonesia, operated by Bluejack Company since 2018. **Discover O’laptop** needs you as a desktop developer to develop system application for managing and tracking the transaction with C# programming language. The requirements are as follows :

**Database**:

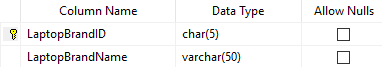
* + - 1. **User**

****

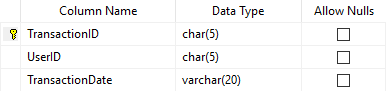
* + - 1. **Laptop**

****

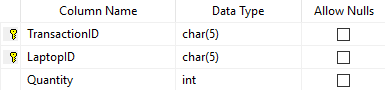
* + - 1. **Laptop Brand**

****

* + - 1. **Header Transaction**

****

* + - 1. **Detail Transaction**

****

**Application**:

1. Main Form

****

**Main Form for Staff**

****

**Main Form for Member**

**Main Form** is a MDI form that contains submenus to open another forms. The **Main Form** always in maximized state and **opened after login**.

**Components**:

* **Menu Strip** for main menu

**Main Menu**:

* **Account**
  + **Change Password** : show **Manage Laptop Form**
  + **Log Out** : to logout and back to **Login Form**
* **Manage**
  + **Laptop** : show **Manage Laptop Form**
  + **Laptop Brand** : show **Manage Laptop Brand Form**
  + **User** : show **Manage User Form**
* **Buy** : show **Buy Laptop Form**
* **View**
  + **Transaction** : show **View Transaction Form**
  + **History** : show **View History Form**
* **Exit** : Close the application

**Description**:

* After login, menus wich can be accessed:
  + For user as **Admin**:

1. Change Password .
2. Log Out .
3. Manage Laptop
4. Manage Laptop Brand
5. Manage User
6. View Transaction
7. Exit
   * For user as **Member**:

1. Change Password .

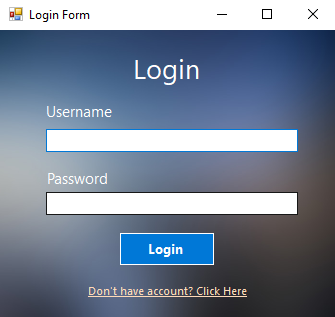
2. Log Out

3. Buy

4. View History

5. Exit

1. Login Form

****

This form is used for login and will appear when the application started.

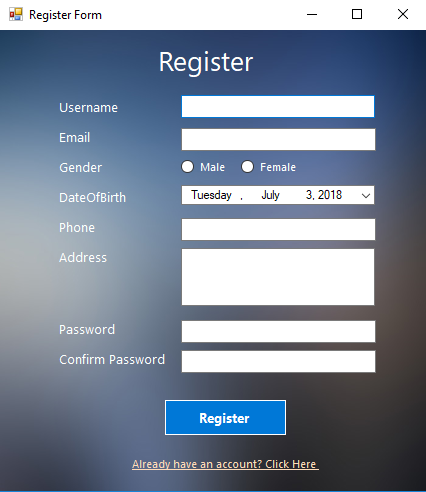
**Components**:

* **Label** for Title, Username, and Password
* **TextBox** for Username and Password
* **Button** for Login
* **LinkLabel** for Register

**Description**:

* The initial state of form:
  + **Username** and **Password** **TextBox** are empty
* If **Login Button** is clicked, then validate:
  + If **Username** is empty, then show error message “**Username must be filled!**”
  + If **Password** is empty, then show error message “**Password must be filled!**”
  + If **Username** and **Password** doesn’t match with any data in **User** table, then show error message “**Invalid Username or Password!**”
  + If **Username** and **Password** match with data in **User** table, then **Login** **Form** will be closed and application will redirect to the **Main Form** and shows the menu according to **UserRole** (**Admin** or **Member**)
* If **Register LinkLabel** is clicked, then:
  + Close the **Login Form** and show the **Register Form**

1. Register Form



Register Form is used for registering new member.

**Components**:

* **Label** for Title, Username, Email, Gender, DateOfBirth, Phone, Address, Password, and Confirm Password
* **TextBox** for Username, Email, Phone, Password , and Confirm Password
* **RadioButton** for Male and Female
* **DateTimePicker** for DateOfBirth
* **RichTextBox** for Address
* **Button** for Register
* **LinkLabel** for Login

**Description**:

* The initial state of form:
  + All fields to input data are empty
  + DateTimePicker DOB set to current date
* If **Register Button** is clicked, then validate:
  + If **Username** is empty, then show error message “**Username must be filled!**”
  + If **Username** already exist in **User** table, then show error message “**Username already exists**”
  + If **Email** is empty, then show error message “**Email must be filled!**”
  + If **Email** does not contain ‘**@**’ and ‘**.**’, then show error message “**Email must contain ‘@’ and ‘.’**”
  + If **Email** start with ‘**@**’ or ‘**.**’, then show error message “**Email cannot start with ‘@’ or ‘.’**”
  + If **Email** end with ‘**@**’ or ‘**.**’, then show error message “**Email cannot end with ‘@’ or ‘.’**”.
  + If **Email** contain ‘**@**’ and ‘**.**’, but placed beside each other, then show error message “**’@’ and ‘.’ cannot be placed beside each other**”
  + If **Gender** is not chosen, then show error message “**Gender must be chosen**!”
  + If **DateOfBirth** is greater than current date, then show error message “**DOB cannot be greater than current date!**”
  + If **Phone** is empty, then show error message “**Phone number must be filled!**”
  + If **Phone** does not consist of digits only, then show error message “**Phone must be numeric!**”
  + If **Phone** length is not 12 digits, then show error message “**Phone must be 12 digits**”
  + If **Address** is empty, then show error message “**Address must be filled!**”
  + If **Address** does not contain “Street”, then show error message “**Address must contain ‘Street’**”
  + If **Password** is empty, then show error message “**Password must be filled!**”
  + If **Password** length is less than 5 characters, then show error message “**Password length must be 5 characters or more”**
  + If **Confirm Password** does not match with **Password**, then show error message “**Confirm Password must be the same with Password!**”
  + If all data have been inputted correctly, then insert data into **User** table, by generating the following **UserID**:

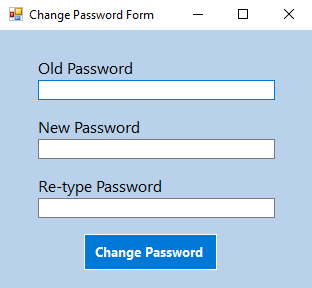
**USXXX**

where, **XXX** = [0-9][0-9][0-9]

For example, if the last **UserID** is “**US002**”, then the program will generate it into **“US003”**. If it is the first user, then the program will generate it into “**US001**”. Then, insert the data into **User** table

* If **Login LinkLabel** is clicked, then:
  + Close the **Register Form** and show the **Login Form**

1. Change Password Form



**Change Password Form** is used to change the password of the **Admin** and **Member**.

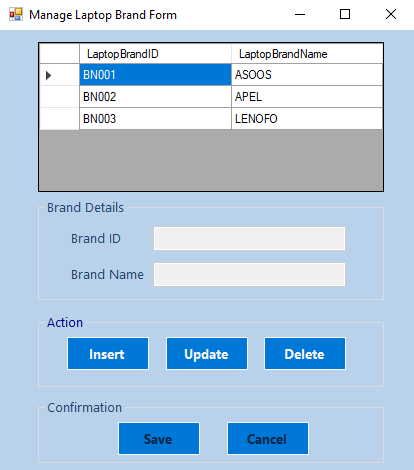
**Components**:

* **Label** for Old Password, New Password, and Re-type Password
* **TextBox** for Old Password, New Password, and Re-type Password
* **Button** for Change Password

**Description**:

* The initial state of form:
  + All fields to input data are empty
* If **Change Password Button** is clicked, then validate:
  + If **Old Password** is empty, then show error message “**Old Password must be filled!**”
  + If **Old Password** doesn’t match with **current** **logged** **in** **User Password**, then show error message “**Old Password doesn’t match with the current password**”
  + If **New Password** is empty, then show error message “**New Password must be filled!**”
  + If **New Password** length is less than 5 characters, then show error message “**New Password length must be 5 characters or more**”
  + If **Re-type Password** doesn’t match with **New Password**, then show error message “**Re-type Password doesn’t match with New Password!**”
  + If all data have been inputted correctly, then show message “**Successfully Change Password**”, the application will update the **current** **logged** **in** **User Password**,and the form will be closed

1. Manage Laptop Brand Form

****

This form used to insert, update, or delete laptop brand from **Laptop Brand** table in database.

**Components**:

* **Label** for Brand ID and Brand Name
* **TextBox** for Brand ID and Brand Name
* **GroupBox** for Brand Details, Action, and Confirmation
* **Button** for Insert, Update, Delete, Save, and Cancel
* **DataGridView** to show data from **Laptop Brand** table

**Description**:

* The initial state of form:
  + Show all data from **Laptop Brand** table in **DataGridView**
  + All fieldsto input **Laptop Brand** data aredisabled
  + Save and Cancel **Button** are **disabled**
* If **DataGridView** is clicked, then show all information from selected row to the existing fields
* If **Insert Button** is clicked, then:
  + BrandName **TextBox** is **enabled** and **empty**
  + **Brand ID** is automatically generated from **Laptop Brand** tablewith format:

**BNXXX**

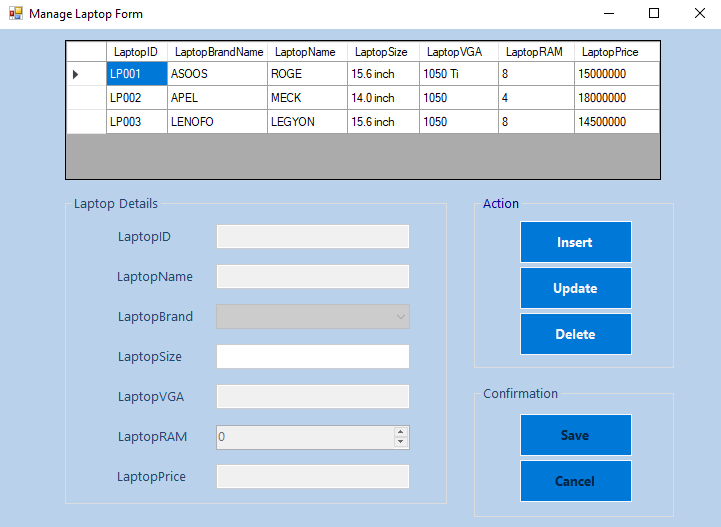
where, **XXX** = [0-9][0-9][0-9]

For example, if the last **Brand ID** is “**BN002**”, then the program will generate it into “**BN003**”. If it is the first brand, then the program will generate it into “**BN001**”

* + Insert, Update, and Delete **Button** will be **disabled**
  + Save and Cancel **Button** will be **enabled**
* If **Update Button** is clicked, then:
  + If user hasn’t selected any data in **DataGridView**, then show an error message **“Please select data first!”**
  + Otherwise, all fields to input Laptop Brand data except **Brand ID** are **enabled** and filled with information based on selected data in **DataGridView**
  + Insert, Update, and Delete **Button** will be **disabled**
  + Save and Cancel **Button** will be **enabled**
* If **Delete Button** is clicked, then:
  + If user hasn’t selected any data in **DataGridView**, then show an error message “**Please select data first!**”
  + Otherwise, show confimation message, “**Are you sure want to delete this data?**”. If user confirm to delete, the corresponding **Laptop Brand** will be deleted from database

* If **Save Button** is clicked, then validate:
  + **Laptop Brand Name** length must be between 2 – 15 characters. If not, then show error message “**Brand Name length must be between 2-15 characters!**”
  + If **Laptop Brand Name** already exist in **Laptop Brand** table, then show error message “**Brand Name already exist!**”
  + If all data have been inputted correctly, then the application will show message “**Data inserted/updated successfully!”**,then insert or update the data into **Laptop Brand** tableaccording to the selected button before, and the form will return to initial state
* If **Cancel Button** is clicked, then the form will return to initial state

1. Manage Laptop Form

****

This form is used to insert, update, or delete laptop from **Laptop** table in database

**Components**:

* GroupBoxfor Laptop Details, Action, and Confirmation
* Label for Laptop ID, Laptop Name, Laptop Brand, Laptop Size, Laptop VGA, Laptop RAM, and Laptop Price
* DataGridViewto show data from **Laptop** table
* TextBoxfor Laptop ID, Laptop Name, Laptop Size, Laptop VGA, and Laptop Price
* ComboBoxfor Laptop Brand
* NumericUpDown for Laptop RAM
* Buttonfor Insert, Update, Delete, Save, and Cancel

**Description**:

* The initial state of form:
  + Show all data from **Laptop** table in **DataGridView**
  + Show all brand from **Laptop Brand** table in Laptop Brand **ComboBox**
  + All fields to input are **disabled**
  + Save and Cancel **Buttons** are **disabled**
* If **DataGridView** is clicked, then show all information from selected row to the existing fields
* If **Insert Button** is clicked, then:
  + All fields to input **Laptop** data except **Laptop ID** are enabled and empty
  + **Laptop ID** is automatically generated from **Laptop** tablewith format:

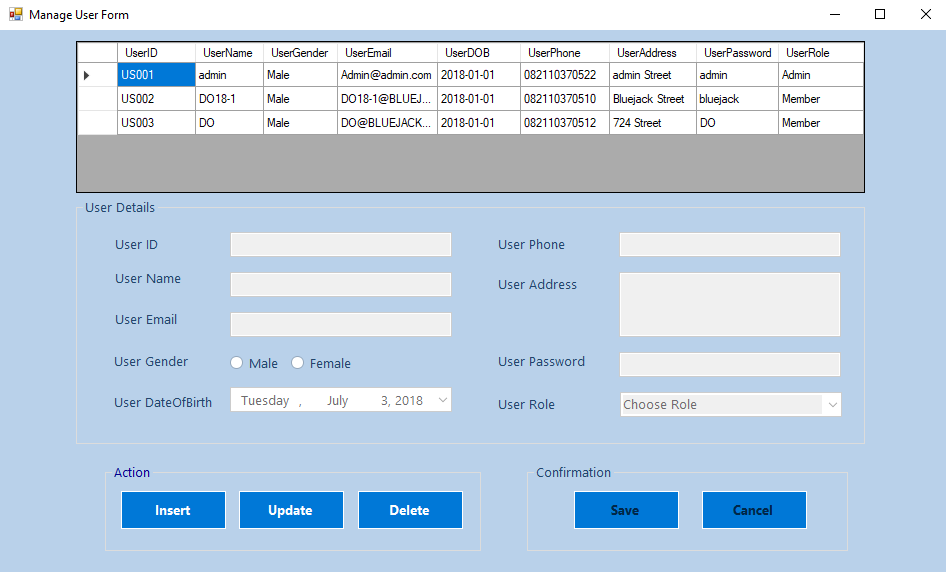
**LPXXX**

where, **XXX** = [0-9][0-9][0-9]

For example, if the last **Laptop ID** is **“LP002”**, then the program will generate it into **“LP003”**. If it is the first laptop, then the program will generate it into **“LP001”**.

* + Insert, Update, and Delete **Button** will be **disabled**
  + Save and Cancel **Button** will be **enabled**
* If **Update Button** is clicked, then:
  + If user hasn’t selected any data in **DataGridView**, show an error message “**Please select data first!**”
  + Otherwise, all fields to input Laptop data except **Laptop ID** are **enabled** and filled with information based on selected data in **DataGridView**
  + Insert, Update, and Delete **Button** will be **disabled**
  + Save and Cancel **Button** will be **enabled**
* If **Delete Button** is clicked, then:
  + If user hasn’t selected any data in **DataGridView**, then show error message “**Please choose data first!**”
  + Otherwise, show confimation message, “**Are you sure want to delete this data?**”. If user confirm to delete, the corresponding **Laptop** will be deleted from database
* If **Save Button** is clicked, then validate :
  + If **Laptop Name** is empty, then show error message “**Laptop Name must be filled!**”
  + If **Laptop Brand** has not been chosen, then show error message “**Laptop Brand must be chosen!**”
  + If **Laptop Size** is empty, then show error message “**Laptop Size must be filled!**”
  + If **Laptop** **Size** does not end with ‘inch’, then show error message “**Laptop Size must end with ‘inch’**”
  + If **Laptop VGA** is empty, then show error message “**Laptop VGA must be filled!**”
  + If **Laptop RAM** is less than 2, then show error message “**Laptop RAM must be 2 or more!**”
  + If **Laptop Price** is empty, then show error message “**Laptop Price must be filled!**”
  + If **Laptop Price** is not numeric then show error message “**Laptop Price must be numeric!**”
  + If all data have been inputted correctly, then the application will show message “**Data inserted/updated successfully!”**,then insert or update the data into **Laptop** tableaccording to the selected button before, and the form will return to initial state
* If **Cancel Button** is clicked, then the form will return to initial state

1. Manage User Form

****

This form used to insert, update, or delete user from **User** table in database.

**Components**:

* **GroupBox** for User Details, Action, and Confirmation
* **Label** for User ID, UserName, User Email, User Gender, User DateOfBirth ,User Phone, User Address, User Password, and User Role
* **TextBox** for User ID, UserName, User Email, User Phone, and User Password
* **RadioButton** for Male and Female
* **DateTimePicker** for User DateOfBirth
* **RichTextBox** for User Address
* **ComboBox** for User Role
* **Button** for Insert, Update, Delete, Save, and Cancel
* **DataGridView** to show data from **User** table

**Description:**

* The initial state of form:
  + Show all data from **User** table in **DataGridView**
  + All fields to input **User** data are **disabled**
  + User Role **ComboBox** must be filled with “**Admin**” and “**Member**”
  + User DateOfBirth **DateTimePicker** set to current date
  + Save and Cancel **Buttons** are **disabled**
* If **DataGridView** is clicked, then show all information from selected row to the existing fields
* If **Insert Button** is clicked, then:
  + All fileds to input **User** data except **User ID** are enabled and empty
  + **UserID** is automatically generated from **User** tablewith format:

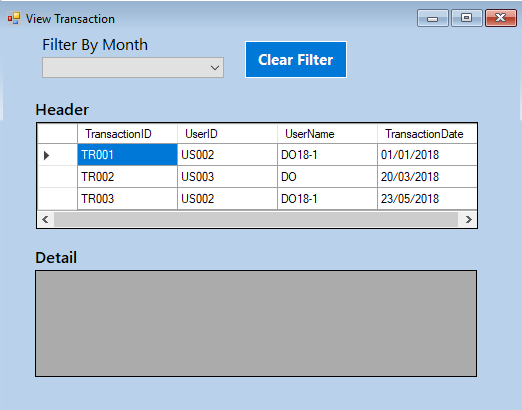
**USXXX**

where, **XXX** = [0-9][0-9][0-9]

For example, if the last **UserID** is “**US002**”, then the program will generate it into **“US003”**. If it is the first user, then the program will generate it into “**US001**”

* + Insert, Update, and Delete **Button**s will be **disabled**
  + Save and Cancel **Buttons** will be **enabled**
* If **Update Button** is clicked, then:
  + If user hasn’t selected any field in **DataGridView**, show an error message “**Please select data first!**”
  + Otherwise, all fields to input User data except **User ID** are **enabled** and filled with information based on selected data in **DataGridView**
  + Insert, Update, and Delete **Button** will be **disabled**
  + Save and Cancel **Button** will be **enabled**
* If **Delete Button** is clicked, then:
  + If user hasn’t selected any field in **DataGridView**, show an error message “**Please select data first!**”
  + Otherwise, show confimation message, **“Are you sure want to delete this data**?”. If user confirm to delete, the corresponding **User** will be deleted from database
* If **Save Button** is clicked, then validate :
  + If **Username** is empty, then show error message “**Username must be filled!**”
  + If **Username** already exist in **User** table, then show error message “**Username already exists**”.
  + If **Email** is empty, then show error message “**Email must be filled!**”
  + If **Email** does not contain ‘**@**’ and ‘**.**’, then show error message “**Email must contain ‘@’ and ‘.’**”
  + If **Email** start with ‘@’ or ‘.’, then show error message “**Email cannot start with ‘@’ or ‘.’**”
  + If **Email** end with ‘@’ or ‘.’, then show error message “**Email cannot end with ‘@’ or ‘.’**”.
  + If **Email** contain ‘@’ and ‘.’, but ‘@’ symbol is next to ‘.’, then show error message “**’@’ sign cannot be placed directly beside ‘.’ sign**”
  + If **Gender** is not chosen, then show error message “**Gender must be chosen**!”
  + If **DateOfBirth** is greater than current date, then show error message “**DOB cannot be greater than current date!**”
  + If **Phone** is empty, then show error message “**Phone number must be filled!”**
  + If **Phone** does not consist of digits only, then show error message “**Phone must be numeric!**”
  + If **Phone** length is not 12 digits, then show error message “**Phone must be 12 digits**”
  + If **Address** is empty, then show error message “**Address must be filled!**”
  + If **Address** does not contain “Street”, then show error message “**Address must contain ‘Street’**”.
  + If **Password** is empty, then show error message “**Password must be filled!**”
  + If **Password** length is less than 5 characters, then show error message “**Password length must be 5 characters or more”**
  + If Role has not been selected, then show error message “**Role must be chosen**”
  + If all data have been inputted correctly, then the application will show message “**Data inserted/updated successfully!”**, then insert or update data into **User** tableaccording to the selected button before, and the form will return to initial state
* If **Cancel Button** is clicked, then the form will return to initial state.

1. View Transaction Form

****

**View Transaction Form** is used to view data from **Header Transaction** and **Detail Transaction** table in database.

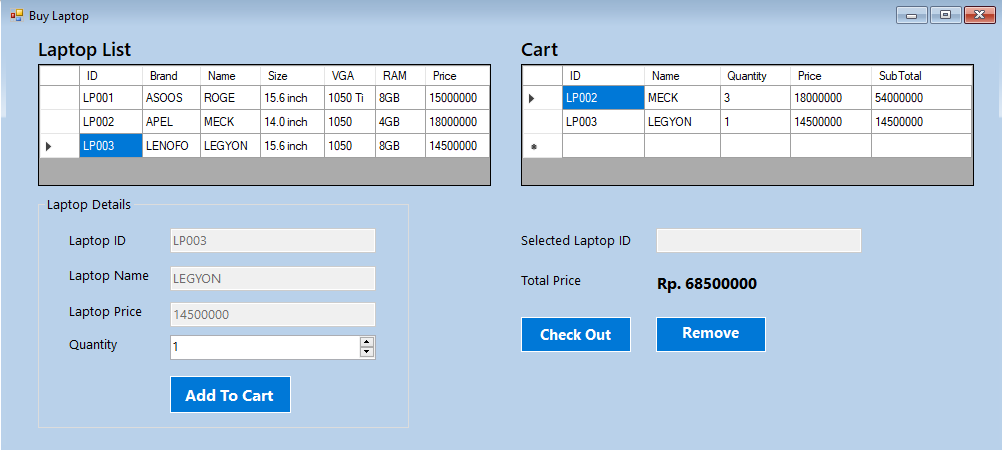
**Components**:

* **Label** for Filter By Month, Header, and Detail
* **DataGridView** to show data from **Header Transaction** and **Detail Transaction** table
* **ComboBox** for Filter By Month
* **Button** for Clear Filter

**Description:**

* The initial state of form:
  + Show all data from **Header Transaction** table in **Header** **DataGridView**
  + Filter By Month **ComboBox** is filled with month name from January - December
* If **DataGridView** of **Header Transaction** is clicked, then **DataGridView** of **Detail Transaction** will show the **Detail Transaction** of the selected **Transaction ID**
* If Filter By Month **ComboBox** value is changed, then **Header DataGridView** will show **Transaction** **Header** data based on the selected month
* If **Clear Filter Button** is clicked, then remove the filter and show all data in **Header DataGridView**

1. Buy Laptop Form

****

**Buy Laptop Form** is used for **member** to buy laptop.

**Components**:

* **GroupBox** for Laptop Details
* **Label** for Laptop List, Cart, Laptop ID, Laptop Name, Laptop Price, Quantity, Selected Laptop ID, Total Price, Total Price (content)
* **TextBox** for Laptop ID, Laptop Name, Laptop Price, Selected Laptop ID
* **NumericUpDown** for Quantity
* **Button** for Add To Cart , Check Out, and Remove
* **DataGridView** to show data from **Laptop** table and show data from **member’s cart**

**Description**:

* The initial state of form:
  + Show all data from **Laptop** table in **DataGridView**
  + All input fields in Laptop Details **GroupBox** are empty and disabled **except** Quantity
  + Selected Laptop ID **TextBox** is empty and disabled
  + **DataGridView** **Cart’s** **column** contains **ID**, **Name**, **Quantity**, **Price**, and **SubTotal** (obtained from **Quantity** \* **Price**)
* If **DataGridView** for **Laptop** is clicked, then show all information from selected row to the existing fields
* If **DataGridView** for **Cart** is clicked, then show the selected ID in **Selected Laptop ID TextBox**
* If **Add to Cart Button** is clicked, then validate:
  + If user hasn’t selected any data in **Laptop** **DataGridView**, then show error message “**Please select data first!**”
  + If **Quantity** is less than 1, then show error message **“Quantity must be more than or equals to 1!”**
  + If all data have been inputted correctly, then add the **selected Laptop** to **Cart** and filled the **Total Price Label** with sum of **each** **subtotal** in **member’s cart**
* If **Remove Button** is clicked, then validate:
  + If user hasn’t selected any laptop in **Cart** **DataGridView**, show an error message “**Please select data first!**”
  + Otherwise, remove the **Selected Laptop ID** from member’s cart
* If **Check Out Button** is clicked, then validate:
  + If there is no laptop in cart, then show error message **“No laptop in cart!”**
  + Otherwise, **insert** the data to **Header Transaction** and **Detail Transaction Table** with the following rule:
    - **TransactionID** is automatically generated from **Header Transaction** tablewith format:

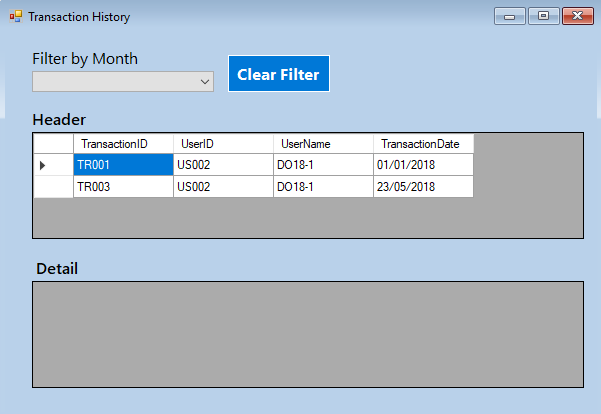
**TRXXX**

where, **XXX** = [0-9][0-9][0-9]

For example, if the last **TransactionID** is **“TR002”**, then the program will generate it into **“TR003”**. If it is the first transaction, then the program will generate it into **“TR001”**

* + - **TransactionDate** in **Header Transaction** table will be filled with the current date
* If **Form** is closed, then remove all data from member’s cart

1. View History Form

****

**View History Form** is used to view data from **Header Transaction** and **Detail Transaction** table of the **current logged in member**.

**Components**:

* **Label** for Filter By Month, Header, and Detail
* **DataGridView** to show data from **Header Transaction** and **Detail Transaction** table
* **ComboBox** for Filter By Month
* **Button** for Clear Filter

**Description**:

* The initial state of form:
  + Show all data from **Header Transaction** table in **Header** **DataGridView**
* If **DataGridView** of **Header Transaction** is clicked, then **DataGridView** of **Detail Transaction** will show the **Detail Transaction** of the selected **Transaction ID**
* If Filter By Month **ComboBox** value is changed, then **Header DataGridView** will show **Transaction** **Header** data based on the selected month
* If **Clear Filter Button** is clicked, then remove the filter and show all data in **Header DataGridView**

**References**:

* https://www.google.co.id/search?q=laptop&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjMl9DT\_JvbAhXKp48KHdq1BfgQ\_AUICigB#imgrc=VXIzxRi9ORlGEM:

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. Additional rules:
   * **LINQ** operation in database is optional (students are allowed to use other techniques for database that taught during practicum).
   * All validations must use the **techniques** taught during **practicum**. (Another language or Regular Expression is **prohibited**).
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
   * \*.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