|  |  |
| --- | --- |
| **DOKUMENTASI EKSTERNAL**  External Documentation |  |
| **PROJECT LAB BC08**  ***Project*** |
| **MATAKULIAH COMP6115 – OOAD**  ***Subject*** |
| SEMESTER GENAP 2019/2020 ***Even Semester Year 2019/2020*** |

## Judul Proyek

*Project**Title*

* **Sudut Meong**

## Pendahuluan

*Introduction*

* **Sudut Meong** is a local cat shop in Indonesia opened by Mr. Kevin Sudut. Mr. Kevin plans to improve its service by developing a system that can help them to manage their business activity within the shop. In this project we will develop the application using Java-based Programming with **MVC** (**Model View Controller**) architecture for a better development process.
* **Model**

The model layer is responsible for representing concepts in the business or information about the business situation. Model layer also responsible for giving access to the database via its public interfaces to acquiring and manipulating references to preexisting domain objects.

* **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.

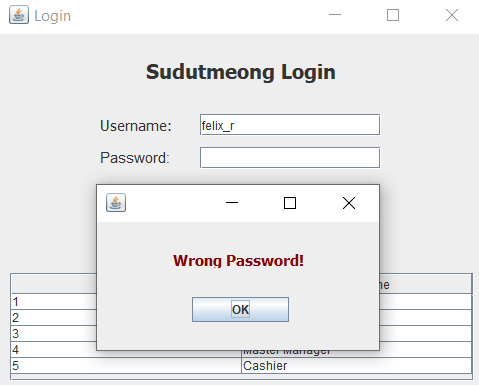
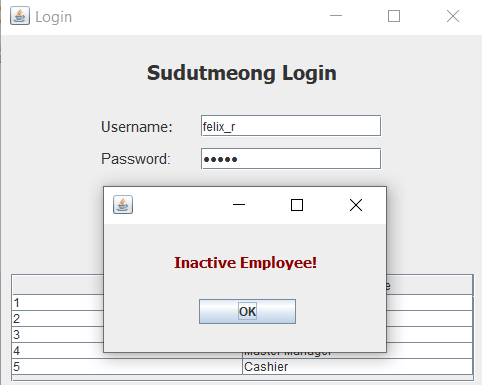
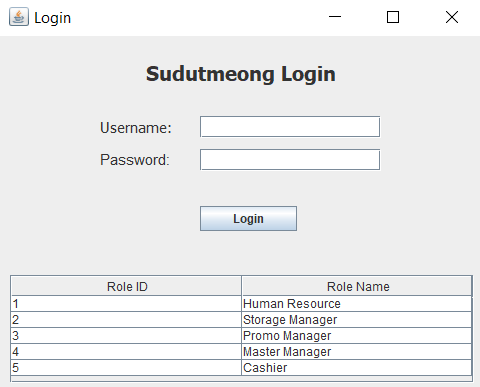
* **Controller**

This layer is responsible to validate all input from the view layer and all business logics are implemented in the controller layer. It also responsible for delegating requests from the user to the lower layer for further processing.

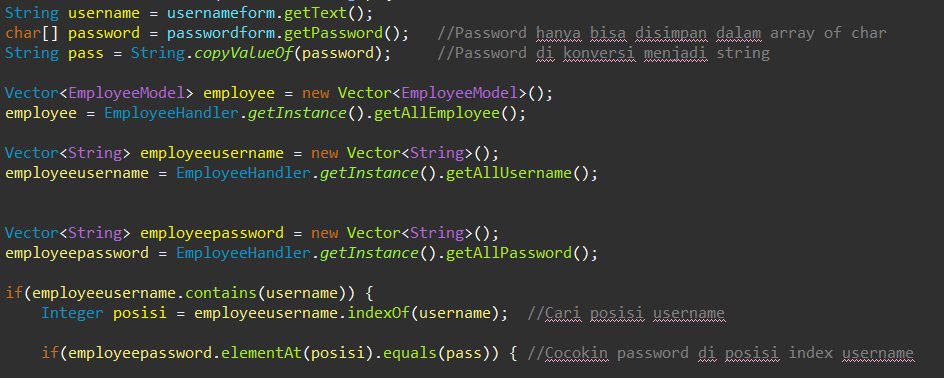
## Dokumentasi

*Documentation*

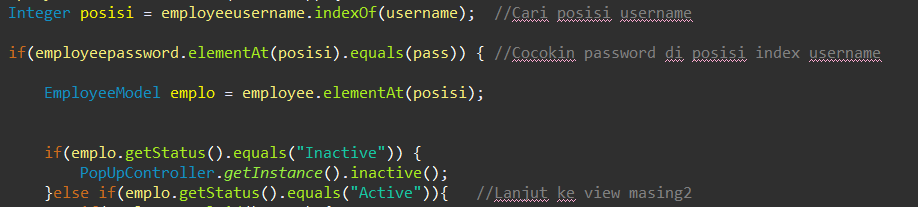
* **Prerequisite Run**
  + The program can only be run using JDK 8.
  + The program can only be run when the server (XAMPP) running.
  + For first time run, please use our backup database query (sudutmeong.sql) to fill the employee, voucher, and product database.
  + For quick login into each role, please use username from 1 to 5, this username does not require any password. For further information check Login documentation.
  + If the table in any view does not change after altering data, please use the refresh button provided (WORST CASE).
* **Login**
  + In this login view we can see a table containing all the *Role ID* and *Role Name* provided in this Sudutmeong system.
  + A popup dialog will show if the employee input the wrong password.
  + A popup dialog will show if the employee status is inactive (fired).

****

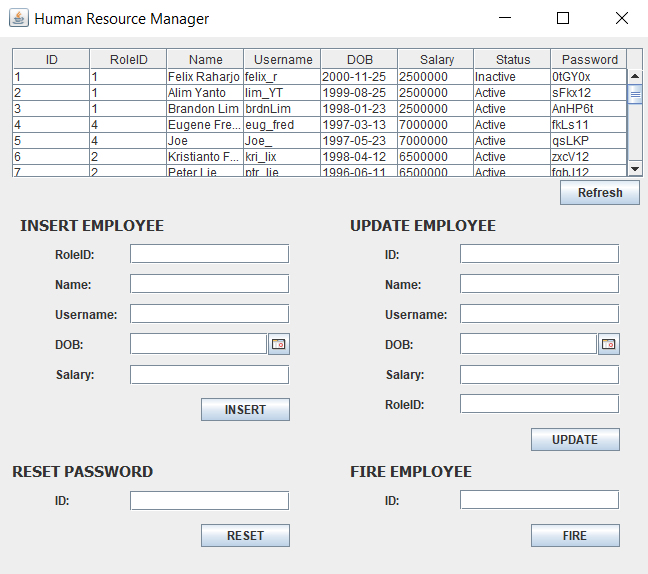
* + Validation code for checking if employee username exist and checking if password match that index of employee username:

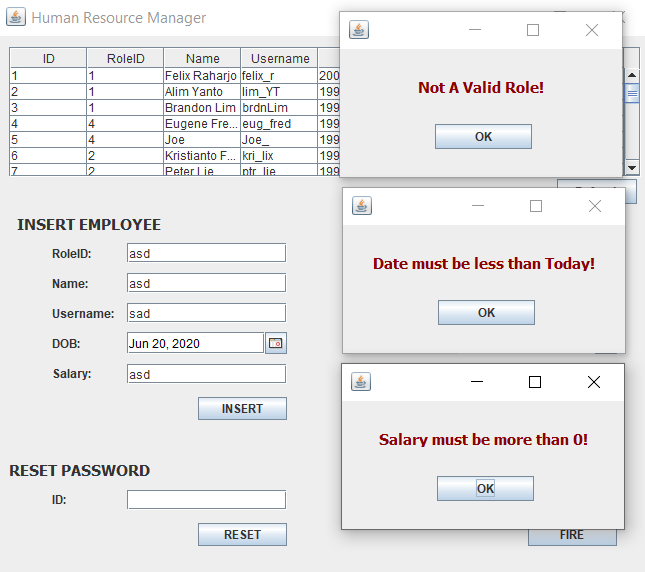
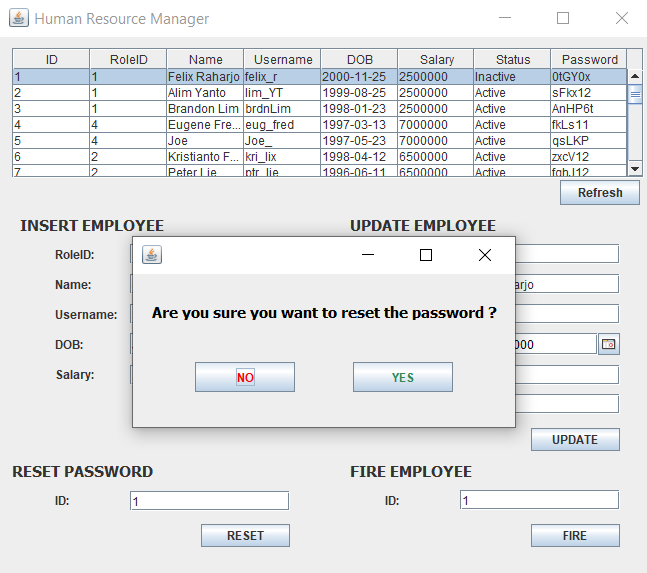


* + Validation code for checking employee status:

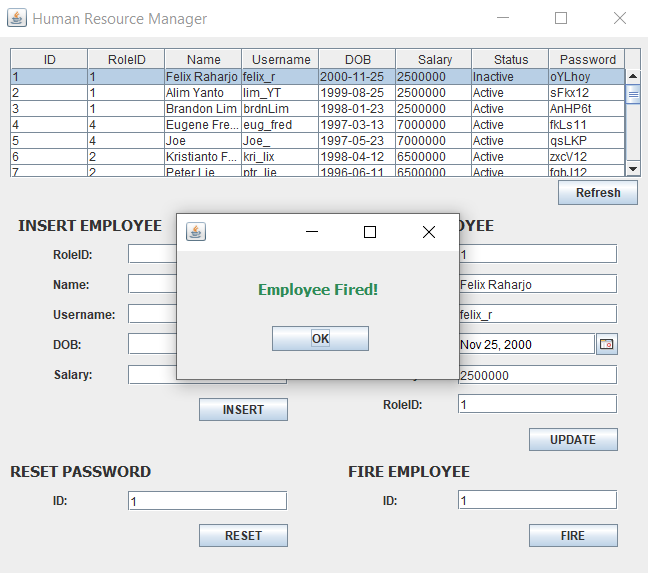
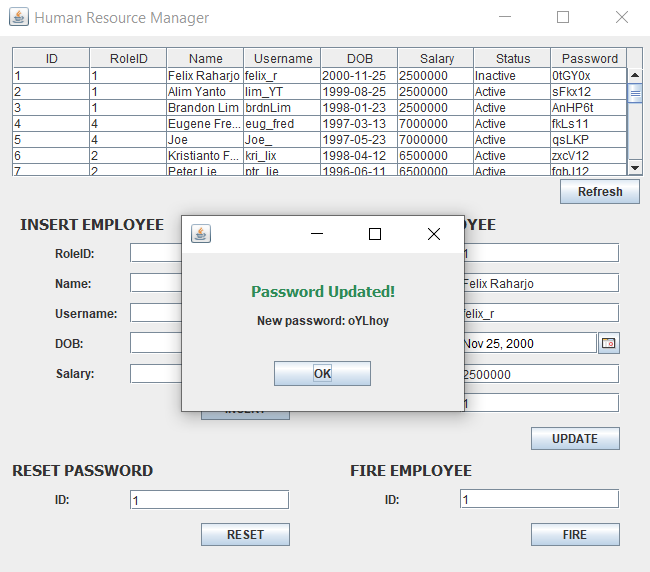
****

* **Human Resource Manager**
  + In this HRM view we can see a table containing all the data from each employee of Sudutmeong store.
  + Various popup dialogs will show if the given input is not available in database or given input is an invalid input.
  + A popup confirmation dialog will show if the action considered as *advance action* (cannot be undone in future) such as reset password and fire employee.





* + After making *advance action* such as reset password and fire employee, please use refresh button provided to update the table data and the view as well.

****

****

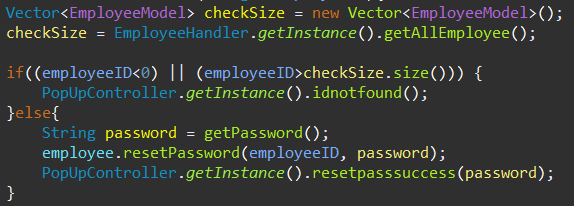
* + Validation code for new employee insertion:



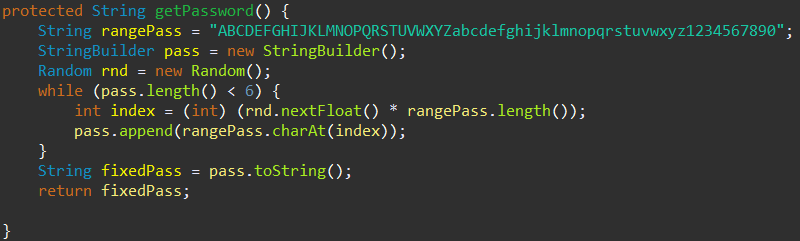
* + Validation code for updating employee data:



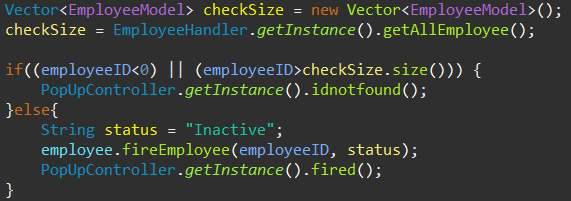
* + Validation code for resetting employee password:



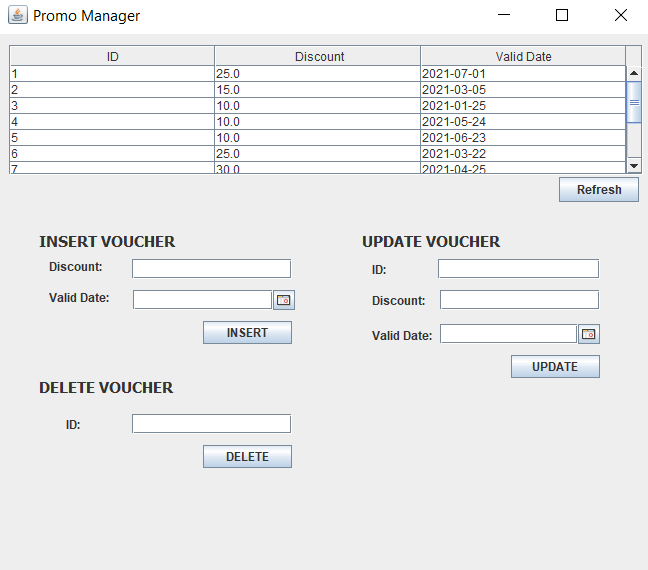
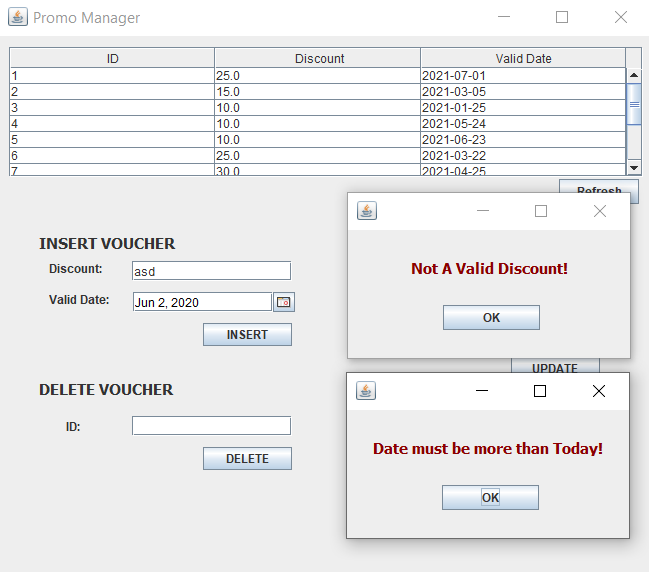
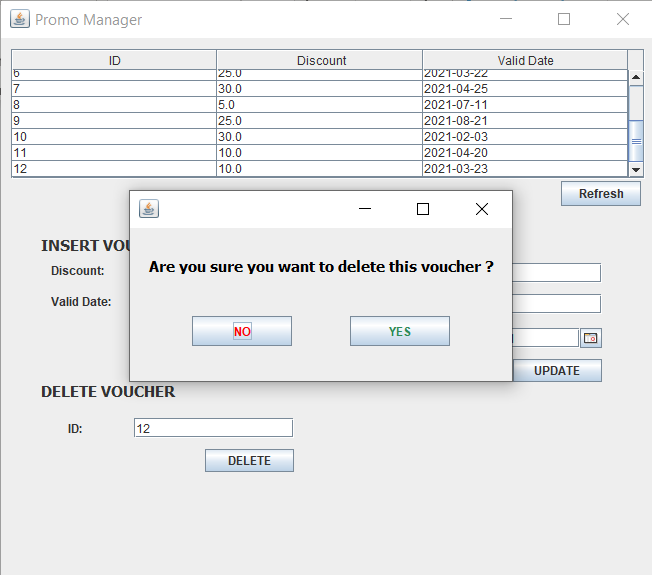
* + Code used to generate password consist of six random character:



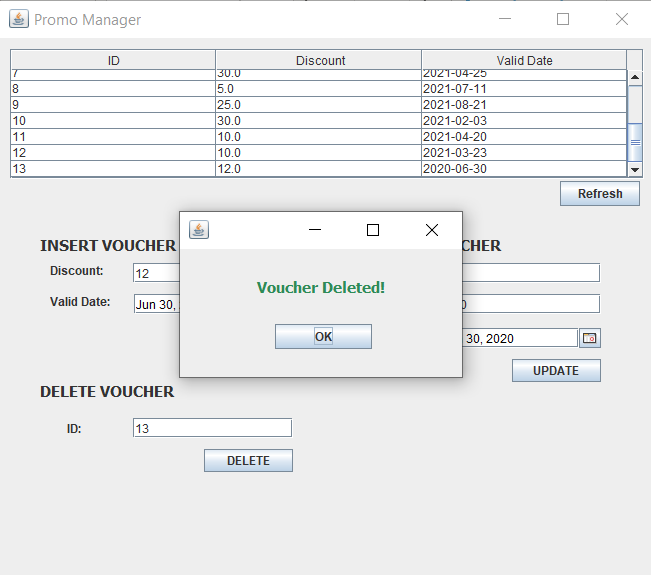
* + Validation code for firing employee:



* **Promo Manager**
  + In this PM (Promo Manager) view we can see a table containing all the data from each voucher exist in database.
  + Various popup dialogs will show if the given input is not available in database or given input is an invalid input.
  + A popup confirmation dialog will show if the action considered as *advance action* (cannot be undone in future) such as delete voucher.



* + After making *advance action* such as delete voucher, please use refresh button provided to update the table data and the view as well.

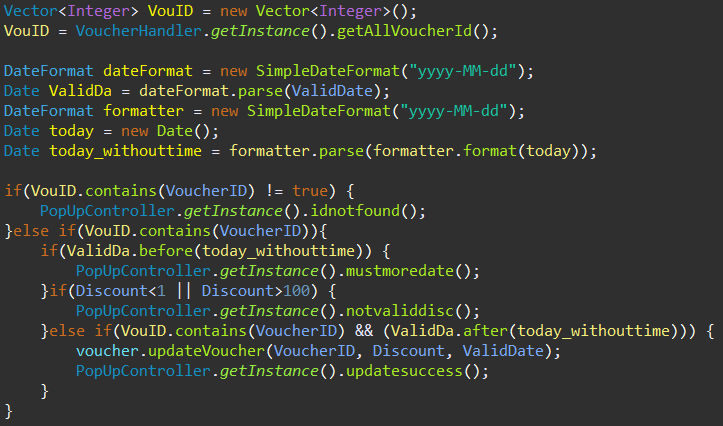




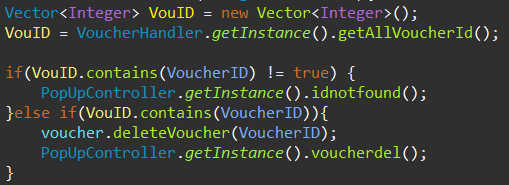
* + Validation code for new voucher insertion:



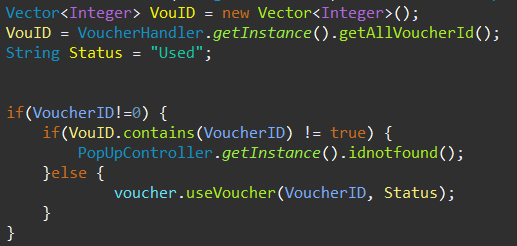
* + Validation code for updating voucher data:



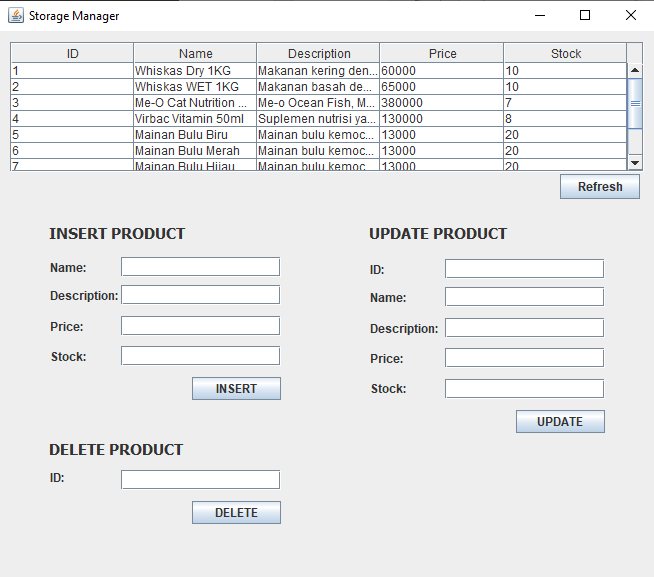
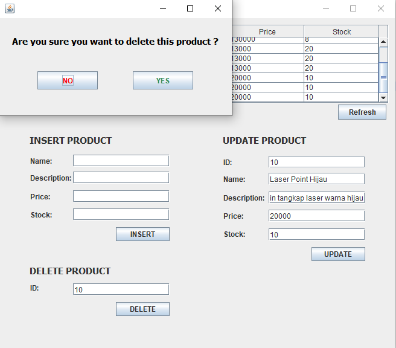
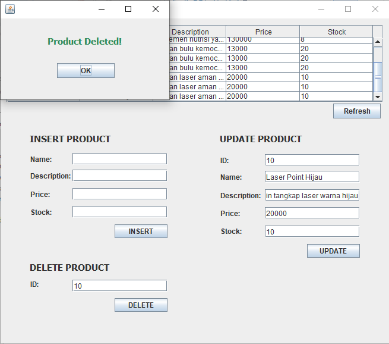
* + Validation code for deleting voucher:



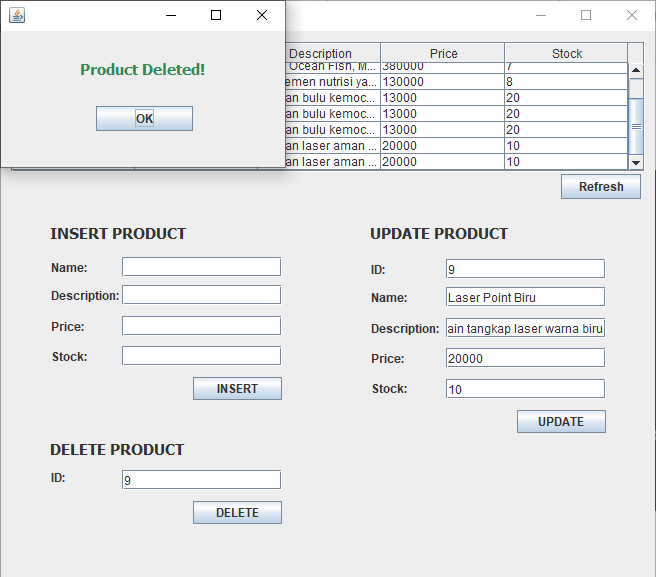
* + Validation code if a transaction uses voucher:



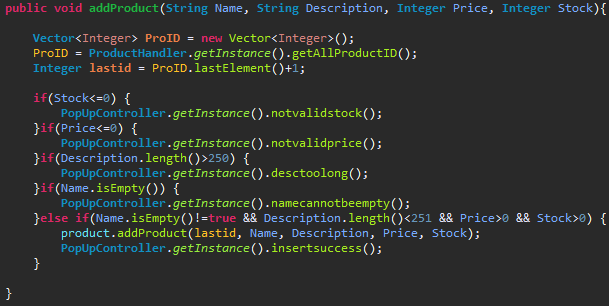
* **Storage Manager**
  + In this SM (Storage Manager) view we can see a table containing all the data from each product exist in database.
  + Various popup dialogs will show if the given input is not available in database or given input is an invalid input.
  + A popup confirmation dialog will show if the action considered as *advance action* (cannot be undone in future) such as delete product.

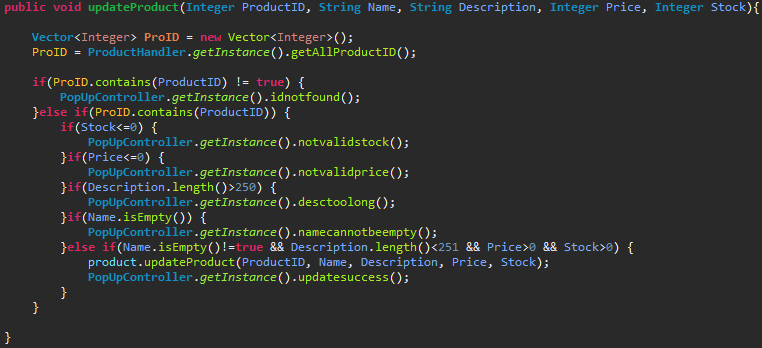
* + After making *advance action* such as delete product, please use refresh button provided to update the table data and the view as well.



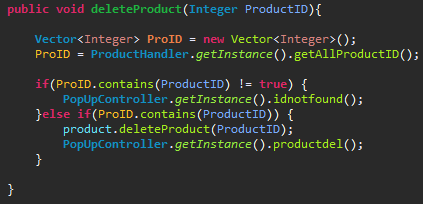
* Validation code for new product insertion:



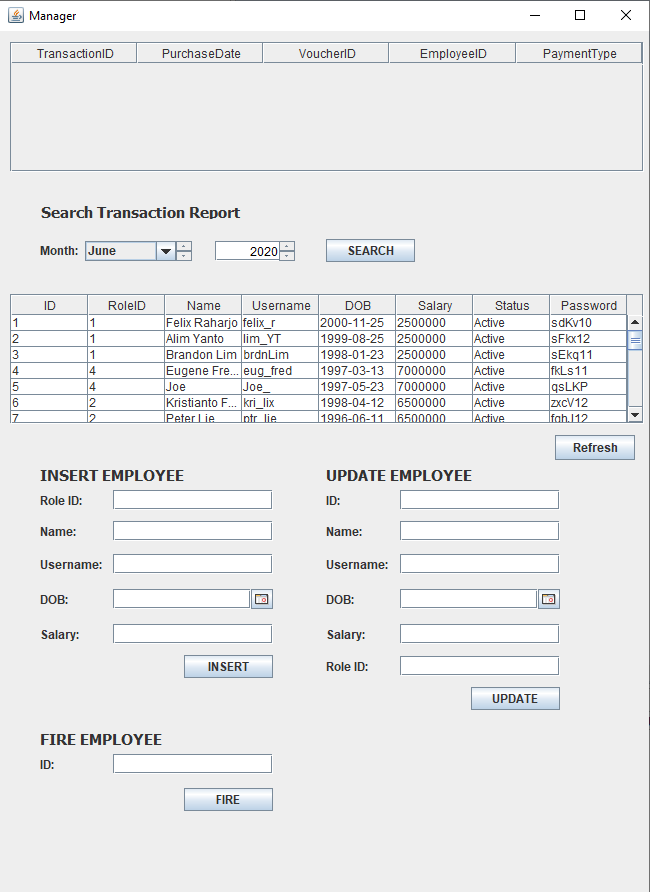
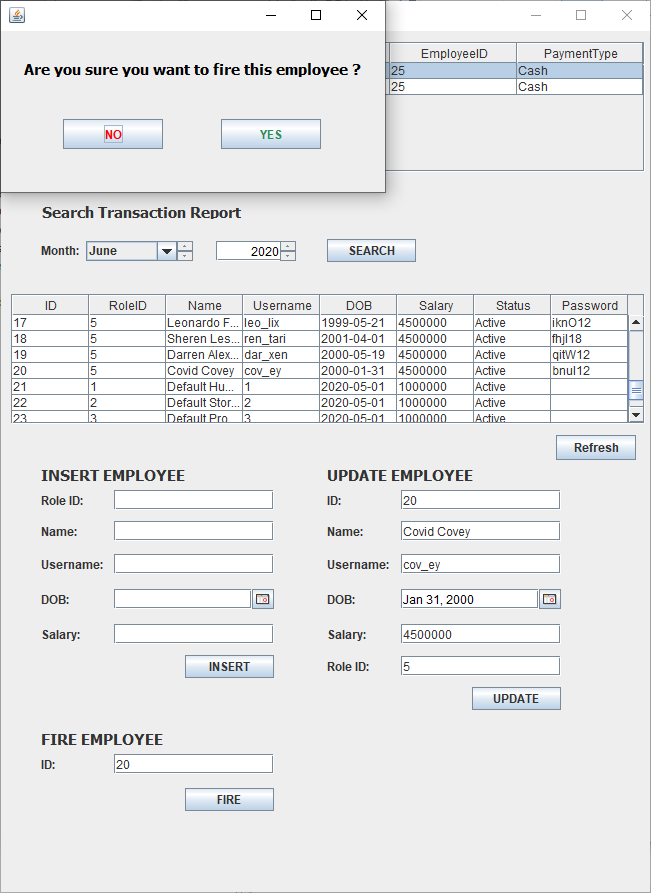
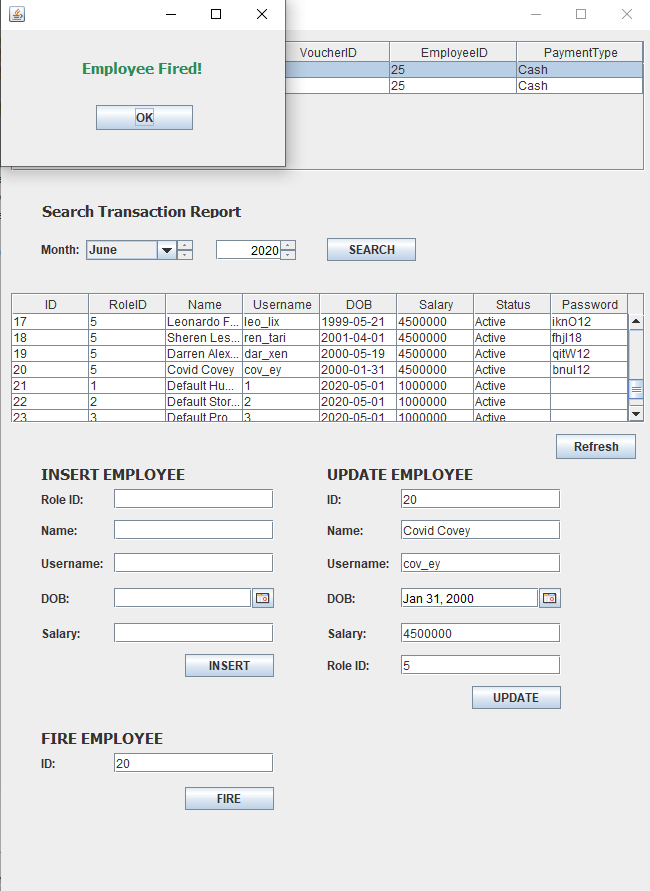
* + Validation code for updating product data:



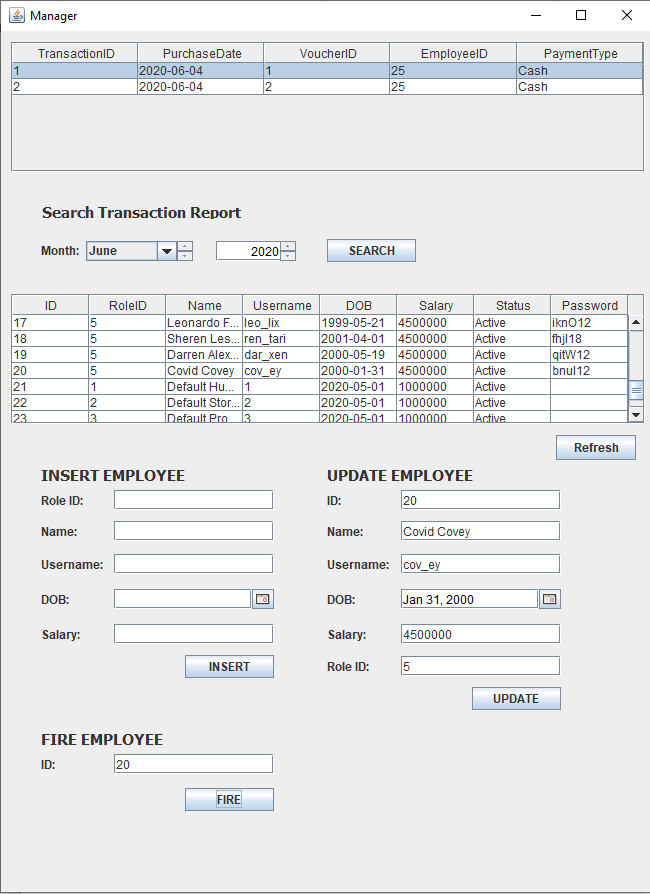
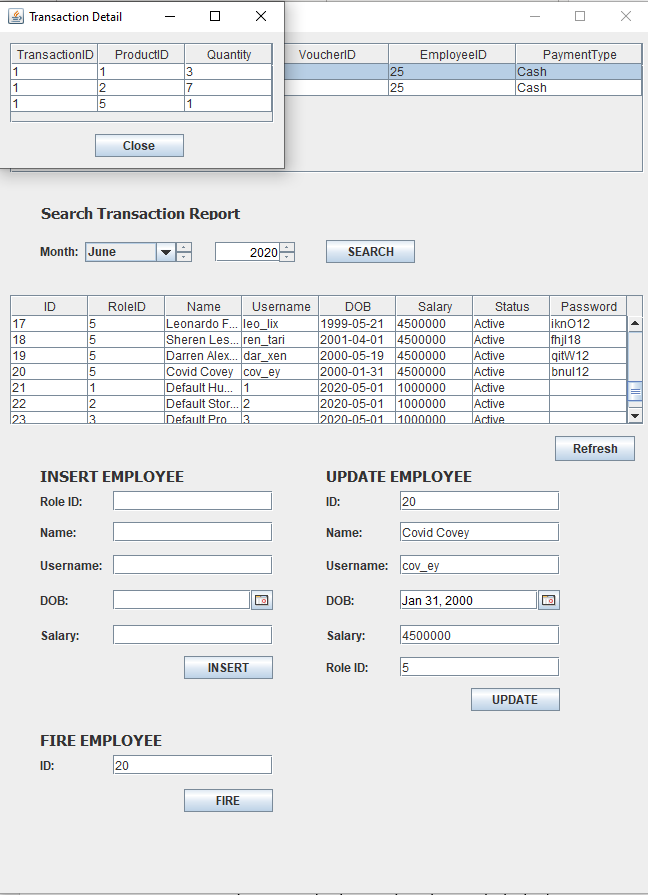
* + Validation code for deleting product:



* **Master Manager**
  + In this M (Manager) view we can see a table containing all the data from each transaction, transaction item, and employee exist in database.
  + Various popup dialogs will show if the given input is not available in database or given input is an invalid input.
  + A popup confirmation dialog will show if the action considered as *advance action* (cannot be undone in future) such as delete voucher.
  + This view will show two table (from top) transaction and employee.

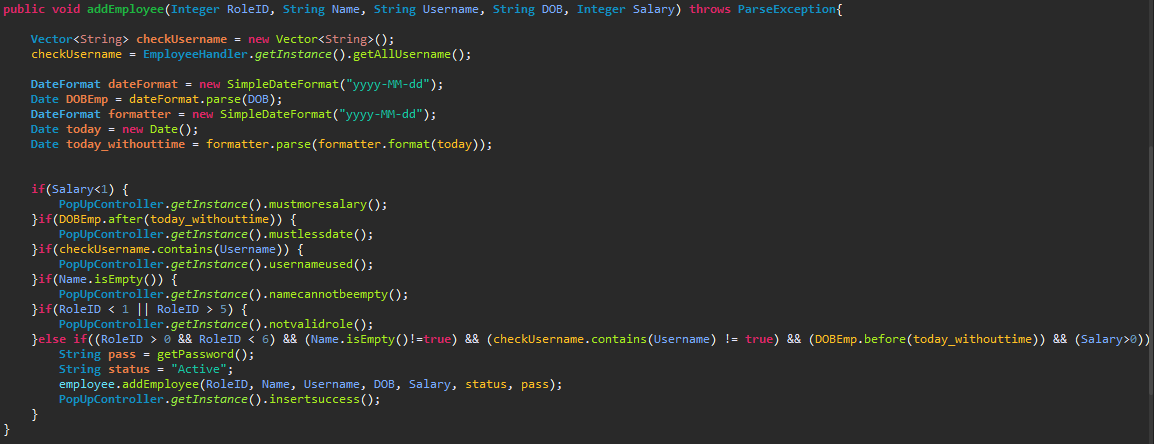
* Clicking on transaction in transaction table will show transaction detail.

* Function for search transaction history grouped by month and year.



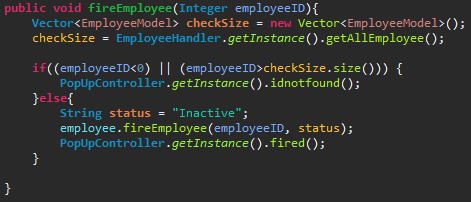
* + Validation code for new employee insertion:



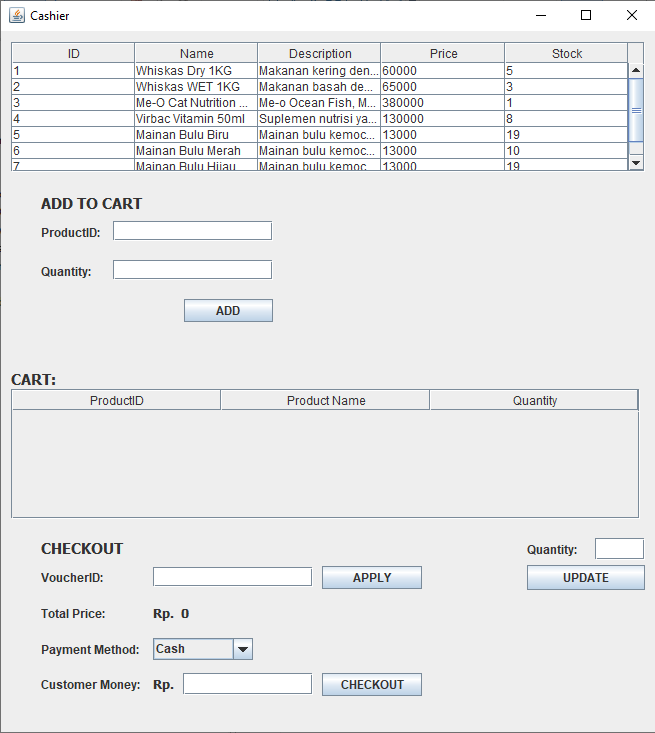
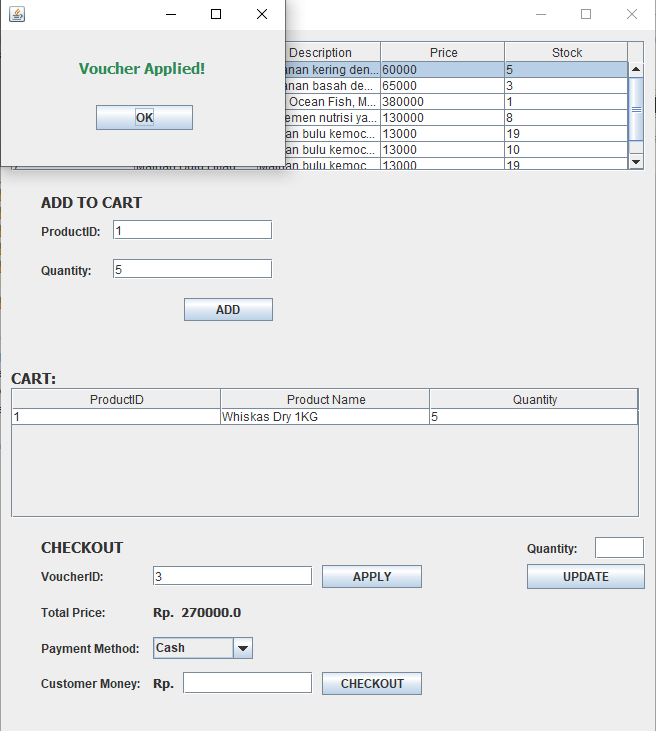
* + Validation code for updating employee data:



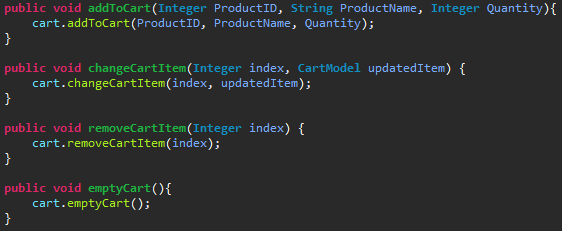
* + Validation code for firing employee:

****

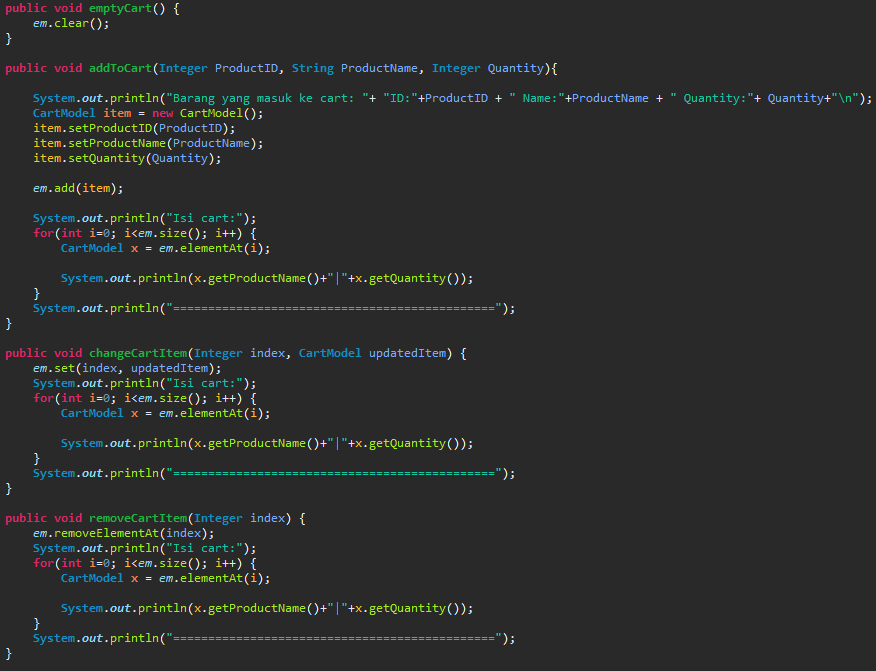
* **Cashier**
  + In this CM (Cart Manager) view we can see a table containing all the data from each product in database and cart.
  + Various popup dialogs will show if the given input is not available in database or given input is an invalid input.
  + A popup confirmation dialog will show if the action considered as *advance action* (cannot be undone in future) such as delete voucher.
  + This view will show two table (from top) product and cart.

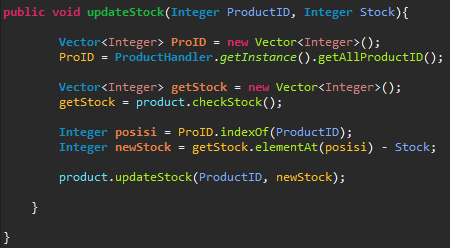
* Function for add item to cart, update, delete, and empty cart.



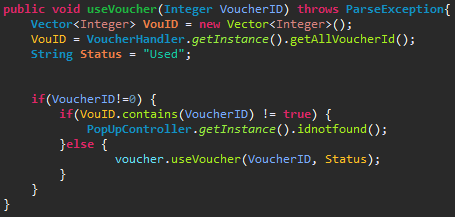
* Function for add item to cart, update, delete, and empty cart. (cont.)



* Validation code for update stock in product:



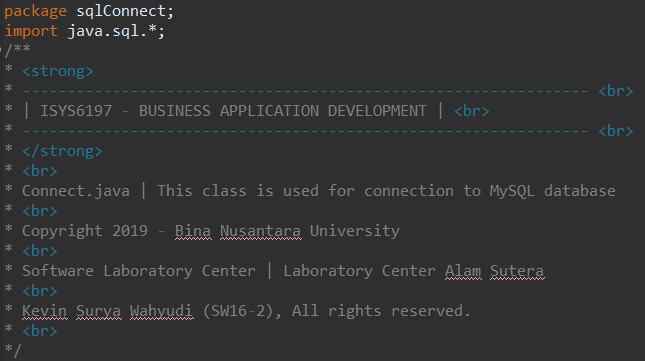
* Validation code for use voucher:



## Referensi

*References*

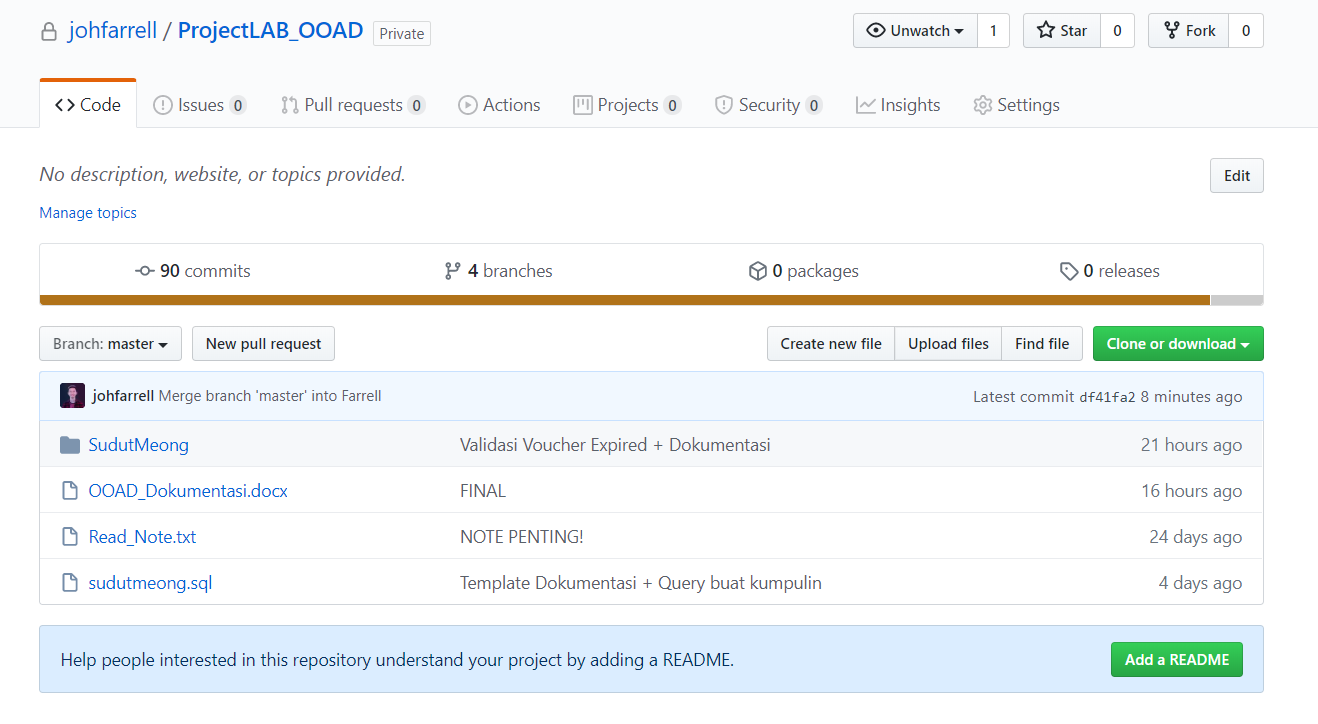
* **Java SQL Connector**
  + Java SQL Connector from Kevin Surya Wahyudi (SW16-2) || SLC Alam Sutra

****

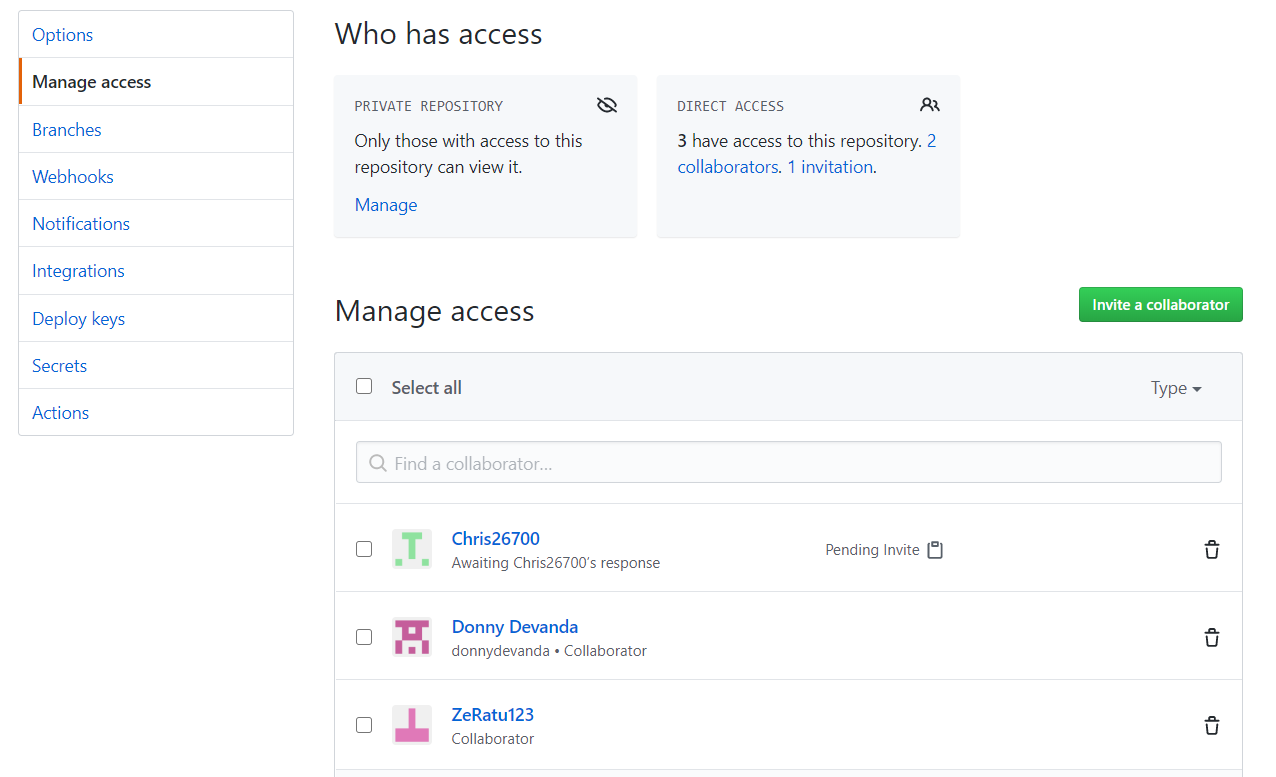
* **MVC Concept and Implementation**
* Martin Wijaya (Aslab)

<https://drive.google.com/file/d/1FzI7eea2lM3XIN8cpfjUzVzYYVW9zXUl/view>

* **GitHub Repository**
* This group project made with a private GitHub repository consist of group members.
* The link for the repo: <https://github.com/johfarrell/ProjectLAB_OOAD>
* Screenshot of GitHub repository and settings:

****

Johannes Farrell (Repository Owner)

****

Christopher Alexander (Collaborator)

Richard Geovani (Collaborator)

Muhammad Donny (Collaborator)

**Disiapkan Oleh,**

*Prepared by,*

* 2201760554 – Johannes Farrell L.
* 2201789025 – Christopher Alexander
* 2201734096 – Richard Geovani S.
* 2201829621 – Muhammad Donny D.