|  |  |
| --- | --- |
| **Practicum Case** |  |
| COMP6122 | COMP6122001  Framework Layer Architecture |
| **Computer Science** | **O221-COMP6122-CT01-10** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **Revision 00** |

**Learning Outcome**

* LO2 – apply design pattern in java
* LO3 – design object oriented in design pattern

**Topic**

* Session 10 – Behavioral Design Pattern II

## Sub Topics

* Mediator
* Iterator

1. **Mediator Pattern**

Mediator is a behavioral design pattern that lets you reduce chaotic dependencies between objects. The pattern restricts direct communications between the objects and forces them to collaborate only via a mediator object.

1. **Iterator Pattern**

Iterator is a behavioral design pattern that lets you traverse elements of a collection without exposing its underlying representation (list, stack, tree, etc.). The main idea of the Iterator pattern is to extract the traversal behavior of a collection into a separate object called an iterator.

## Soal

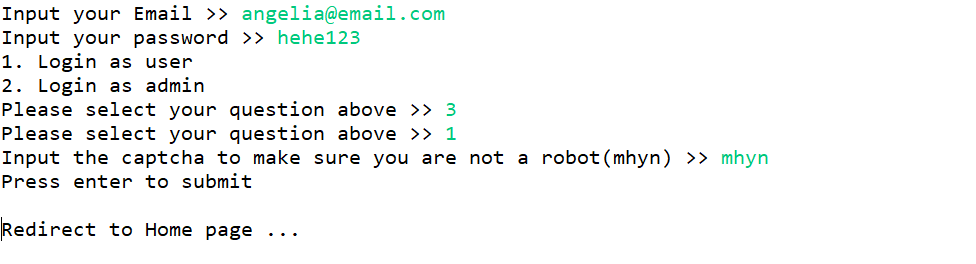
*Case*

**Angel Maya**

Angel Maya is one of the top social media in the world. At this moment, the manager of Angel Maya wants to launch a new login page, register page, and so on. You as the programmer of Angel Maya, the manager asks you to create a **login page** and create some **components** **that can be reusable** for Angel Maya for the future development, such as register page. Below are the following details of the program:

Currently, there are only **five components**, which are **email textbox**, **password textbox**, **admin** **secret textbox**, **captcha textbox**, **list selection**, and **submit** **button**.

* At first, the program will ask the user to input his/her **email** and **must not be empty** (email textbox).
* The program will ask the user to input his/her **password** and **must not be empty** (password textbox).
* The program will ask the user to chooseto **log in** as what. Currently, in the login page, there are only **two options** which are **login as an admin** or **login as a user** (list selection).
* If the user chose login as an **admin**, then the program will ask the user to input the **admin’s secret** and validate the input must **not be empty** (admin secret textbox). **Otherwise**, then the program will ask the user to input **captcha** to make sure the user is not a robot**,** which the captcha will always **random 4 lower case character** and validate the input must **not be empty** (captcha textbox).
* After finish all of the inputs, the program will ask the user to **press enter** to click the submit button.
* After submitted, the submit button automatically will validateall the components that have been submitted. Here are the **validations** for the components:
  + The **email** must contain **at least one** **‘@’ (without quotes)**.
  + The **password** must be **alphanumerical**.
  + If the user chose **login as a user**, then the program will validate the inputted **captcha is equal to generated captcha** by the system. **Otherwise**, then the program will validate the **inputted admin’s secret is equal to ‘4dminS3cr3t’ (without quotes)**.
  + If all **validations** are correct, then print **“redirect to the home page”**

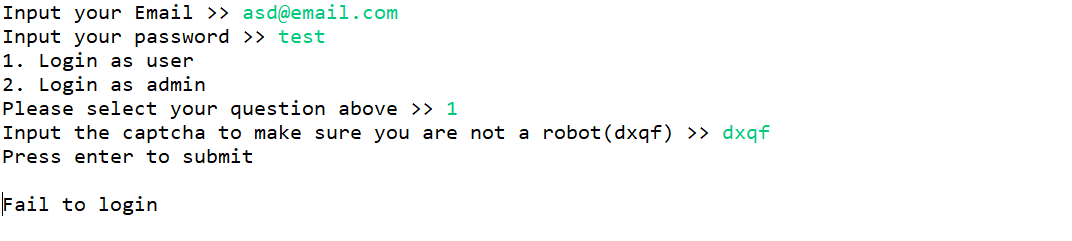


**Figure 1. Success Login as a User**



**Figure 2. Success Login as an Admin**

* + **Otherwise**, print an **error message.**



**Figure 3. Fail to Login**