

```

1 usage 1 thanhhd
public DeliveryInfo processDeliveryInfo(HashMap info) throws InterruptedException, IOException, InvalidDeliveryInfoException {
    LOGGER.info(msg: "Process Delivery Info");
    LOGGER.info(info.toString());
    validateDeliveryInfo(info);
    DeliveryInfo deliveryInfo = new DeliveryInfo(
        String.valueOf(info.get("name")),
        String.valueOf(info.get("phone")),
        String.valueOf(info.get("province")),
        String.valueOf(info.get("address")),
        String.valueOf(info.get("instructions")),
        new DistanceCalculator());
    System.out.println(deliveryInfo.getProvince());
    return deliveryInfo;
}

/**
 * The method validates the info
 * @param info
 * @throws InterruptedException
 * @throws IOException
 */
1 usage 1 thanhhd
public void validateDeliveryInfo(HashMap<String, String> info) throws InterruptedException, IOException, InvalidDeliveryInfoException {
    if (validatePhoneNumber(info.get("phone"))
        || validateName(info.get("name"))
        || validateAddress(info.get("address"))) return;
    else throw new InvalidDeliveryInfoException();
}

```

```

public boolean validatePhoneNumber(String phoneNumber) {
    if (phoneNumber.length() != 10) return false;
    if (!phoneNumber.startsWith("0")) return false;
    try {
        Integer.parseInt(phoneNumber);
    } catch (NumberFormatException e) {
        return false;
    }
    return true;
}

1 usage 1 thanhhd
public boolean validateName(String name) {
    if (Objects.isNull(name)) return false;
    String patternString = "^[a-zA-Z\\s]*$";
    Pattern pattern = Pattern.compile(patternString);
    Matcher matcher = pattern.matcher(name);
    return matcher.matches();
}

1 usage 1 thanhhd
public boolean validateAddress(String address) {
    if (Objects.isNull(address)) return false;
    String patternString = "^[a-zA-Z\\s]*$";
    Pattern pattern = Pattern.compile(patternString);
    Matcher matcher = pattern.matcher(address);
    return matcher.matches();
}
}

```

PlaceOrderController.java có mức độ logical cohesion vì các phương thức validateDeliveryInfo, validatePhoneNumber, validateAddress, validateName không quá liên quan đến nhau.

```

public class Cart {

    9 usages
    private List<CartItem> lstCartItem;

    1 thanhld
    public Cart() { lstCartItem = new ArrayList<>(); }

    1 usage 1 thanhld
    public void addCartMedia(CartItem cm) { lstCartItem.add(cm); }

    1 usage 1 thanhld
    public void removeCartMedia(CartItem cm) { lstCartItem.remove(cm); }

    3 usages 1 thanhld
    public List getListMedia() { return lstCartItem; }

    1 usage 1 thanhld
    public void emptyCart() { lstCartItem.clear(); }

    1 usage 1 thanhld
    public int getTotalMedia(){
        int total = 0;
        for (Object obj : lstCartItem) {
            CartItem cm = (CartItem) obj;
            total += cm.getQuantity();
        }
        return total;
    }
}

```

Cart.java có mức độ Communicational cohesion vì các method trong class đều sử dụng dữ liệu từ CartItem.java

```

public class Utils {

    no usages
    public static DateFormat DATE_FORMATTER = new SimpleDateFormat( pattern: "yyyy/MM/dd HH:mm:ss");
    no usages
    private static Logger LOGGER = getLogger(Utils.class.getName());
    static {
        System.setProperty("java.util.logging.SimpleFormatter.format", "[%4$-4s] [%1$tF %1$tT] [%2$-7s] %5$s %n");
    }

    1 thanhld
    public static Logger getLogger(String className) { return Logger.getLogger(className); }
}

```

Utils.java có mức độ coincidental cohesion vì phương thức DateFormatter định dạng ngày tháng và phương thức trả về logger không hề liên quan với nhau.

```
public ViewCartController getBController() { return (ViewCartController) super.getBController(); }
```

1 usage  thanhhd

```
public void requestToViewCart(BaseScreenHandler prevScreen) throws SQLException {  
    setPreviousScreen(prevScreen);  
    setScreenTitle("Cart Screen");  
    getBController().checkAvailabilityOfProduct();  
    displayCartWithMediaAvailability();  
    show();  
}
```

1 usage  thanhhd

```
public void requestToPlaceOrder() throws SQLException, IOException {  
    try {  
        // create placeOrderController and process the order  
        PlaceOrderController placeOrderController = new PlaceOrderController();  
        if (placeOrderController.getListCartMedia().size() == 0){  
            PopupScreen.error("You don't have anything to place");  
            return;  
        }  
  
        placeOrderController.placeOrder();  
  
        // display available media  
        displayCartWithMediaAvailability();  
  
        // create order  
        Order order = placeOrderController.createOrder();  
  
        // display shipping form
```

CartScreenHandler.java có mức độ temporal cohesion vì các phương thức liên quan đến trình tự thực hiện.

```
public HomeController getBController() { return (HomeController) super.getBController(); }
```

1 usage 1 thanhld

```
protected void setupData(Object dto) throws Exception {  
    setBController(new HomeController());  
    this.authenticationController = new AuthenticationController();  
    try{  
        List medium = getBController().getAllMedia();  
        this.homeItems = new ArrayList<>();  
        for (Object object : medium) {  
            Media media = (Media)object;  
            MediaHandler m = new MediaHandler(ViewsConfig.HOME_MEDIA_PATH, media);  
            m.attach( observer: this);  
            this.homeItems.add(m);  
        }  
    } catch (SQLException | IOException e){  
        LOGGER.info( msg: "Errors occurred: " + e.getMessage());  
        e.printStackTrace();  
    }  
}
```

1 usage 1 thanhld

```
protected void setupFunctionality() throws Exception {  
  
    aimsImage.setOnMouseClicked(e -> {  
        addMediaHome(this.homeItems);  
    });  
  
    cartImage.setOnMouseClicked(e -> {
```

HomeController có mức độ temporal cohesion vì các phương thức liên quan đến trình tự thực hiện.

```

public AuthenticationController getBController() { return (AuthenticationController) super.getBController(); }

1 usage  👤 thanhhd
protected void setupData(Object dto) throws Exception {
}

1 usage  👤 thanhhd
protected void setupFunctionality() throws Exception {
}

no usages  👤 thanhhd
@FXML
void login(MouseEvent event) throws IOException, InterruptedException, SQLException {
    try {
        getBController().login(email.getText(), password.getText());
        PopupScreen.success("Login Successfully!");
        backToHomeScreen(event);
    } catch (Exception ex) {
        PopupScreen.error(ex.getMessage());
    }
}

1 usage  👤 thanhhd
@FXML
void backToHomeScreen(MouseEvent event) throws IOException, InterruptedException, SQLException {
    this.homeScreenHandler.show();
}

```

LoginScreenHandler có mức độ temporal cohesion vì các phương thức liên quan đến trình tự thực hiện.

```

private static PopupScreen popup(String message, String imagePath, Boolean undecorated) throws IOException{
    PopupScreen popup = new PopupScreen(new Stage());
    if (undecorated) popup.stage.initStyle(StageStyle.UNDECORATED);
    popup.message.setText(message);
    popup.setImage(imagePath);
    return popup;
}

2 usages  👤 thanhhd
public static void success(String message) throws IOException{
    popup(message, imagePath: ViewsConfig.IMAGE_PATH + "/" + "tickgreen.png", undecorated: true)
        .show(autoClose: true);
}

👤 thanhhd
public static void error(String message) throws IOException{
    popup(message, imagePath: ViewsConfig.IMAGE_PATH + "/" + "tickerror.png", undecorated: false)
        .show(autoClose: false);
}

no usages  👤 thanhhd
public static PopupScreen loading(String message) throws IOException{
    return popup(message, imagePath: ViewsConfig.IMAGE_PATH + "/" + "loading.gif", undecorated: true);
}

👤 thanhhd
public void setImage(String path) { super.setImage(icon, path); }

```

PopupScreen.java có mức độ logical cohesion vì các phương thức popup message không liên quan đến nhau.