**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**BÁO CÁO BÀI TẬP LỚN**

**BÙI TRỌNG ĐỨC**

duc.bt200157@sis.hust.edu.vn

|  |  |
| --- | --- |
| **Giảng viên:** | TS. Trịnh Tuấn Đạt |
| **Bộ môn:** | Thiết kế và xây dựng phần mềm |
| **Lớp:** | 143801 |
| **Nhóm:** | 6 |
| **HÀ NỘI, 12/2023** | |

**MỤC LỤC**

[CHƯƠNG 1. PHÂN TÍCH YÊU CẦU (BÀI 2) 1](#_Toc154532344)

[1.1 Biểu đồ use case 1](#_Toc154532345)

[1.1.1 Biểu đồ use case tổng quan 1](#_Toc154532346)

[1.1.2 Biểu đồ use case phân rã “Xem thông tin chấm công” 3](#_Toc154532347)

[1.1.3 Chỉnh sửa thông tin chấm công 4](#_Toc154532348)

[1.2 Đặc tả Use case 5](#_Toc154532349)

[1.2.1 Use case “Xem chấm công của bản thân nhân viên văn phòng” 5](#_Toc154532350)

[1.3 Từ điển thuật ngữ 7](#_Toc154532351)

[1.3.1 Comment 8](#_Toc154532352)

[1.3.2 Click 8](#_Toc154532353)

[1.3.3 Import 8](#_Toc154532354)

[1.3.4 Export 8](#_Toc154532355)

[1.4 Đặc tả phụ trợ 8](#_Toc154532356)

[1.4.1 Hiệu năng 8](#_Toc154532357)

[1.4.2 Độ tin cậy 8](#_Toc154532358)

[1.4.3 An toàn, bảo mật 8](#_Toc154532359)

[1.4.4 Bảo trì 8](#_Toc154532360)

[1.4.5 Khả năng sử dụng 9](#_Toc154532361)

[1.4.6 Khả năng mở rộng 9](#_Toc154532362)

[CHƯƠNG 2. THIẾT KẾ MỨC PHÂN TÍCH (BÀI 3) 10](#_Toc154532363)

[2.1 Thiết kế biểu đồ tương tác 10](#_Toc154532364)

[2.1.1 Use case “Xem báo cáo chấm công của bản thân nhân viên văn phòng” 10](#_Toc154532365)

[2.2 Thiết kế biểu đồ lớp 16](#_Toc154532366)

[2.2.1 Use case “Xem chấm công của bản thân nhân viên văn phòng” 16](#_Toc154532367)

[CHƯƠNG 3. THIẾT KẾ GIAO DIỆN VÀ SUBSYSTEM (BÀI 4) 17](#_Toc154532368)

[3.1 Sơ đồ chuyển đổi màn hình 17](#_Toc154532369)

[3.1.1 Sơ đồ chuyển đổi màn hình của UC Xem chấm công của bản thân nhân viên văn phòng 17](#_Toc154532370)

[3.2 Đặc tả màn hình 18](#_Toc154532371)

[3.2.1 Đặc tả màn hình OfficerHomeView 18](#_Toc154532372)

[3.2.2 Đặc tả màn hình OfficerDetailView 20](#_Toc154532373)

[3.2.3 Đặc tả màn hình Hiển thị thông báo xác nhận 21](#_Toc154532374)

[3.3 Biểu đồ trình tự cho Subsystem 22](#_Toc154532375)

[3.4 Biểu đồ lớp cho Subsystem 23](#_Toc154532376)

[CHƯƠNG 4. BIỂU ĐỒ TRÌNH TỰ VÀ LỚP MỨC THIẾT KẾ (BÀI 5) 24](#_Toc154532377)

[4.1 Thiết kế biểu đồ trình tự mức thiết kế 24](#_Toc154532378)

[4.1.1 Use case “Xem chấm công của bản thân nhân viên văn phòng” 24](#_Toc154532379)

[4.2 Biểu đồ lớp mức thiết kế 29](#_Toc154532380)

[4.2.1 Use case “Xem chấm công của bản thân nhân viên văn phòng” 29](#_Toc154532381)

[CHƯƠNG 5. CẢI THIỆN BIỂU ĐỒ TRÌNH TỰ VÀ LỚP MỨC THIẾT KẾ (BÀI 6) 33](#_Toc154532382)

[5.1 Cách chỉnh sửa và ý nghĩa mang lại 33](#_Toc154532383)

[5.1.1 Áp dụng mẫu thiết kế Singleton 33](#_Toc154532384)

[5.1.2 Áp dụng mẫu thiết kế Factory 33](#_Toc154532385)

[5.1.3 Áp dụng nguyên tắc Cohesion and Coupling 33](#_Toc154532386)

[5.1.4 Áp dụng nguyên lý SOLID 33](#_Toc154532387)

[5.2 Thiết kế biểu đồ trình tự mức thiết kế 35](#_Toc154532388)

[5.2.1 Use case “Xem chấm công của bản thân nhân viên văn phòng” 35](#_Toc154532389)

[5.3 Biểu đồ lớp mức thiết kế 39](#_Toc154532390)

[5.3.1 Use case “Xem chấm công của bản thân nhân viên văn phòng” 40](#_Toc154532391)

[CHƯƠNG 6. Lập trình và kiểm thử đơn vị 44](#_Toc154532392)

[6.1 Thiết kế test case cho module: SqliteOfficerAttendanceRespository 44](#_Toc154532393)

[6.1.1 Mô tả module: 44](#_Toc154532394)

[6.1.2 Thiết kế test case cho phương thức getAttendancesOfEmployee(): 44](#_Toc154532395)

# PHÂN TÍCH YÊU CẦU (BÀI 2)

## Biểu đồ use case

### Biểu đồ use case tổng quan

A diagram of people with text

Description automatically generated

Giải thích về các tác nhân:

* **Nhân viên**: Người dùng cấp thấp nhất, chỉ có thể xem được các thông tin chấm công liên quan đến mình
* **Nhân viên văn phòng**: Kế thừa nhân viên, chỉ xem được thông tin chấm công dạng nhân viên văn phòng.
* **Công nhân**: Kế thừa nhân viên, chỉ xem được thông tin chấm công dạng công nhân.
* **Trưởng đơn vị**: Người dùng cấp cao hơn nhân viên, có thể xem được thông tin chấm công của các thành viên trong đơn vị của mình quản lý.
* **Quản lý nhân sự**: Người dùng cấp cao nhất, có thể xem được thông tin chấm công của toàn bộ nhân viên công ty, có quyền chỉnh sửa thông tin chấm công, xem báo cáo, import/export dữ liệu chấm công.
* **Máy quét vân tay**: Hệ thống bên ngoài có nhiệm vụ cung cấp API cho hệ thống lấy dữ liệu chấm công
* **Hệ thống quản lý nhân sự**: Hệ thống bên ngoài có nhiệm vụ cung cấp API cho hệ thống chấm công lấy thông tin của nhân viên.
* **Hệ thống kế toán**: Hệ thống bên ngoài có khả năng lấy thông tin chấm công từ hệ thống chấm công để tính lương của nhân viên cho từng tháng.

Giải thích về các use case:

Composite use case:

* **Xem thông tin chấm công**: Use case tổng quát của các use case thực hiện các hành vi nghiệp vụ lên máy chấm công vân tay.
* **Chỉnh sửa thông tin chấm công**: Use case tổng quát của các use case thực hiện các hành vi nghiệp vụ lên

Actual use case:

* **Cung cấp dữ liệu chấm công nhân viên qua API**
* **Đăng kí tài khoản**
* **Đăng nhập**
* **Lấy dữ liệu chấm công nhân viên qua API**

### A diagram of a diagram Description automatically generatedBiểu đồ use case phân rã “Xem thông tin chấm công”

### Chỉnh sửa thông tin chấm công

A diagram of a person's diagram

Description automatically generated

## Đặc tả Use case

### Use case “Xem chấm công của bản thân nhân viên văn phòng”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC003 | Tên Use case | Xem chấm công của bản thân nhân viên văn phòng |
| Tác nhân | Nhân viên văn phòng, Hệ thống quản lý nhân sự | | |
| Tiền điều kiện | Đã đăng nhập vào hệ thống | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | |  | Nhân viên văn phòng | Chọn chức năng Xem chấm công nhân viên văn phòng | |  | Hệ thống | Gọi đến hệ thống quản lý nhân sự để lấy thông tin nhân viên | |  | Hệ thống quản lý nhân sự | Trả về thông tin của nhân viên | |  | Hệ thống | Lấy các bản ghi chấm công của nhân viên trong tháng hiện tại | |  | Hệ thống | Tính toán thông tin chấm công tổng hợp (số ca đi làm, số giờ đi muộn, số giờ về sớm) của nhân viên theo tháng hiện tại | |  | Hệ thống | Hiển thị thông tin chấm công tổng hợp của nhân viên và bản ghi chấm công (ngày, ca sáng, ca chiều) trong tháng hiện tại | |  | Nhân viên văn phòng | Chọn ngày xem chấm công chi tiết | |  | Hệ thống | Lấy bản ghi chấm của nhân viên trong ngày yêu cầu | |  | Hệ thống | Hiển thị bản ghi chấm công của nhân viên trong ngày yêu cầu | |  | Nhân viên văn phòng | Chọn chức năng Xem chi tiết chấm công | |  | Hệ thống | Hiện thị thông tin chấm công chi tiết trong ngày đã chọn (ca làm việc, số giờ đi muộn, số giờ về sớm) | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 8a. | Hệ thống | Thông báo lỗi: Thời điểm yêu cầu chưa xảy ra | | | |
| Hậu điều kiện | Không | | |

\* Dữ liệu của từng nhân viên gồm các trường dữ liệu sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Ví dụ** |
|  | Họ tên | Họ tên của nhân viên | Bùi Trọng Đức |
|  | Mã nhân viên | Mã định danh của nhân viên | 000001 |

\* Dữ liệu của từng bản ghi chấm công nhân viên văn phòng gồm các trường dữ liệu sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Ví dụ** |
|  | Ngày | Thông tin ngày chấm công | 1-12-2023 |
|  | Mã nhân viên | Mã để định danh từng công nhân | CN-2501 |
|  | Buổi sáng | Có đi làm hay không | True |
|  | Buổi chiều | Có đi làm hay không | False |
|  | Thời gian đi muộn buổi sáng |  | 0.1 |
|  | Thời gian về sớm buổi sáng |  | 0 |
|  | Thời gian đi muộn buổi chiều |  | 0 |
|  | Thời gian về sớm buổi chiều |  | 0.5 |

\*Biểu đồ hoạt động:

A diagram of a diagram

Description automatically generated with medium confidence

## Từ điển thuật ngữ

Sau đây là một số thuật ngữ được sử dụng trong tài liệu:

### Comment

* Thêm miêu tả cho một cái gì đó, cụ thể trong tài liệu này là thêm miêu tả cho mã nguồn

### Click

* Hành động bấm vào nút bên trái chuột 2 lần liên tiếp trong một khoảng thời gian rất ngắn.

### Import

* Hành động nhập dữ liệu bằng file.

### Export

* Hành động xuất dữ liệu ra một file.

## Đặc tả phụ trợ

### Hiệu năng

* Trải nghiệm sử dụng mượt mà, ổn định.
* Thời gian phản hồi của ứng dụng dưới 1s.
* Cho phép 100 người dùng truy cập đồng thời.
* Cập nhật thông tin chấm công từ máy chấm công vân tay đến hệ thống chấm công với độ trễ ít hơn 1s

### Độ tin cậy

* Đảm bảo nhận diện thành công ít nhất 98% lần thử vân tay.
* Dữ liệu được backup thường xuyên đảm bảo không bị mất mát (1 lần/tuần).
* Có công cụ phục hồi dữ liệu từ cơ sở dữ liệu dự phòng nếu cơ sở dữ liệu chính gặp sự cố.
* Cho phép hệ thống hoạt động liên tục 24 giờ/ngày, 7 ngày/tuần, tăng khả năng chịu lỗi bằng việc duy trì hệ thống chấm công trên 2 node.
* Không quá 1 lỗi trên 800 dòng mã.

### An toàn, bảo mật

* Sử dụng chuẩn băm SHA-256 để mã hoá dữ liệu nhạy cảm như đặc trưng vân tay, mật khẩu của người dùng.
* Phân quyền người dùng để những người không liên quan có thể truy cập đến dữ liệu nhạy cảm.
* Yêu cầu xác thực người dùng (đăng nhập) trước khi thực hiện các hành động truy cập vào tài nguyên của hệ thống để tránh dữ liệu bị rò rỉ ra bên ngoài.
* Hệ thống được triển khai trên server cục bộ và được ngắt kết nối với internet để đảm bảo an toàn cho dữ liệu nhân viên (Hệ thống chỉ truy cập được bằng wifi của công ty).

### Bảo trì

* Phần mềm được tách ra thành các module cho từng tính năng để thuận tiện cho quá trình bảo trì.
* Phần mềm sẽ được duy trì trong khoảng 3 năm.
* Tất cả các phương thức trong mã nguồn đều phải được comment cũng như có tài liệu về cấu trúc hệ thống, cấu trúc mã nguồn.

### Khả năng sử dụng

* Giao diện Material thân thiện kèm tài liệu sử dụng, hầu như không cần đào tạo để sử dụng thành thạo.
* Giao diện người dùng có thể chạy được trên các trình duyệt phổ biến, truy cập chỉ bằng 1 click.

### Khả năng mở rộng

* Ứng dụng phải được thiết kế sao cho thuận tiện cho việc mở rộng quy mô về tính năng cũng như số lượng người dùng, số lượng dữ liệu liên tục mở rộng.

# THIẾT KẾ MỨC PHÂN TÍCH (BÀI 3)

## Thiết kế biểu đồ tương tác

### Use case “Xem báo cáo chấm công của bản thân nhân viên văn phòng”

#### Biểu đồ trình tự (mức phân tích)

\* Main Scenario: Nhân viên văn phòng xem chi tiết ngày chấm công

A diagram of a project

Description automatically generated

\* Scenario 2: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo tháng

A diagram of a company

Description automatically generated

\* Scenario 3: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo quý

A screenshot of a computer screen

Description automatically generated

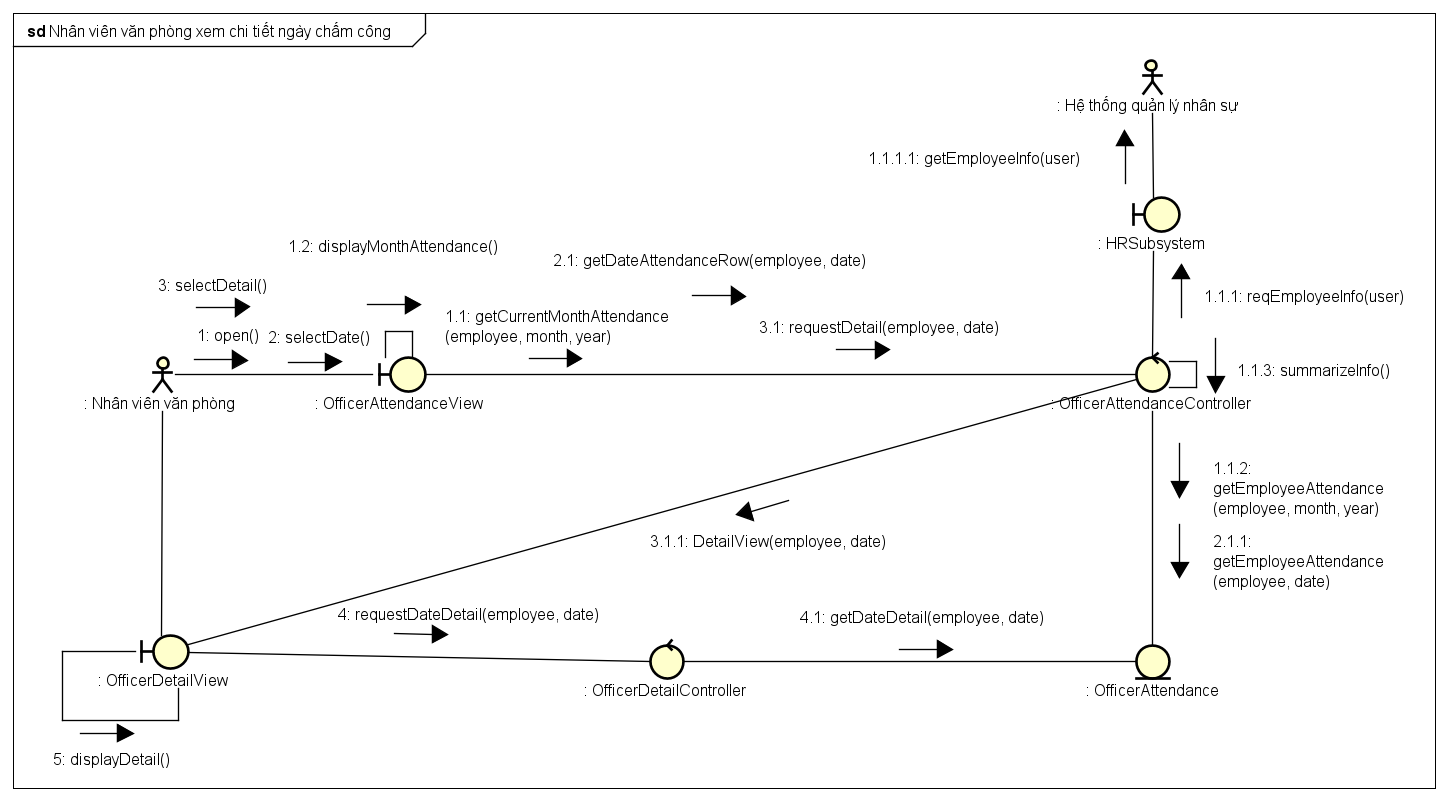
\* Scenario 4: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo năm

A screenshot of a diagram

Description automatically generated

#### Biểu đồ giao tiếp biểu diễn tương đương biểu đồ trình tự

\* Main Scenario: Nhân viên văn phòng xem chi tiết ngày chấm công



## Thiết kế biểu đồ lớp

### Use case “Xem chấm công của bản thân nhân viên văn phòng”

A screenshot of a computer

Description automatically generated

# THIẾT KẾ GIAO DIỆN VÀ SUBSYSTEM (BÀI 4)

## Sơ đồ chuyển đổi màn hình

### Sơ đồ chuyển đổi màn hình của UC Xem chấm công của bản thân nhân viên văn phòng

A diagram of a company structure

Description automatically generated

## Đặc tả màn hình

### Đặc tả màn hình OfficerHomeView

A screenshot of a computer

Description automatically generated

\* Đặc tả điều khiển

|  |  |  |
| --- | --- | --- |
| **Điều khiển** | **Hành vi** | **Chức năng** |
| Hình đại diện | Initial | Hiển thị ảnh đại diện của người dùng |
| Text Tên người dùng | Initial | Hiển thị tên người dùng |
| Text Mã nhân viên | Initial | Hiển thị mã nhân viên của người dùng |
| Text Tên phòng ban | Initial | Hiển thị tên phòng ban của người dùng |
| Button Đăng xuất | Click | Đăng xuất người dùng khỏi hệ thống |
| Button Trang chủ | Click | Điều hướng về Trang chủ của Trưởng đơn vị |
| ComboBox Loại | Click | Chọn loại thống kê theo tháng/quý/năm |
| ComboBoxList | Click |  |
| ComboBox Năm | Click | Chọn năm |
| Button Thống kê | Click | Thực hiện chức năng thống kê |
| Thông báo | Click | Hiển thị thông báo |
| Text Tổng số ca làm việc | Initial | Hiển thị Tổng số ca làm việc |
| Text Tổng số giờ đi muộn | Initial | Hiển thị Tổng số giờ đi muộn |
| Text Tổng số giờ về sớm | Initial | Hiển thị Tổng số giờ về sớm |
| PickerDate | Click | Chọn ngày cụ thể |
| Table Chấm công theo từng ngày | Initial | Hiển thị thông tin chấm công theo từng ngày |
| Button Xem chi tiết | Click | Điều hướng đến trang Hiển thị thông tin chấm công chi tiết của nhân viên |
| Text Tổng số bản ghi | Initial | Hiển thị tổng số dòng của bảng |

\* Đặc tả dữ liệu trên từng dòng của bảng chấm công của từng nhân viên

|  |  |  |
| --- | --- | --- |
| **Tên trường dữ liệu** | **Kiểu dữ liệu** | **Kiểu cách** |
| Họ tên | String | Căn trái |
| Mã nhân viên | String | Căn giữa |
| Đơn vị | String | Căn trái |
| Tổng số ca đi làm | Int | Căn giữa |
| Tổng số giờ đi muộn/về sớm | Double | Căn phải |

### Đặc tả màn hình OfficerDetailView

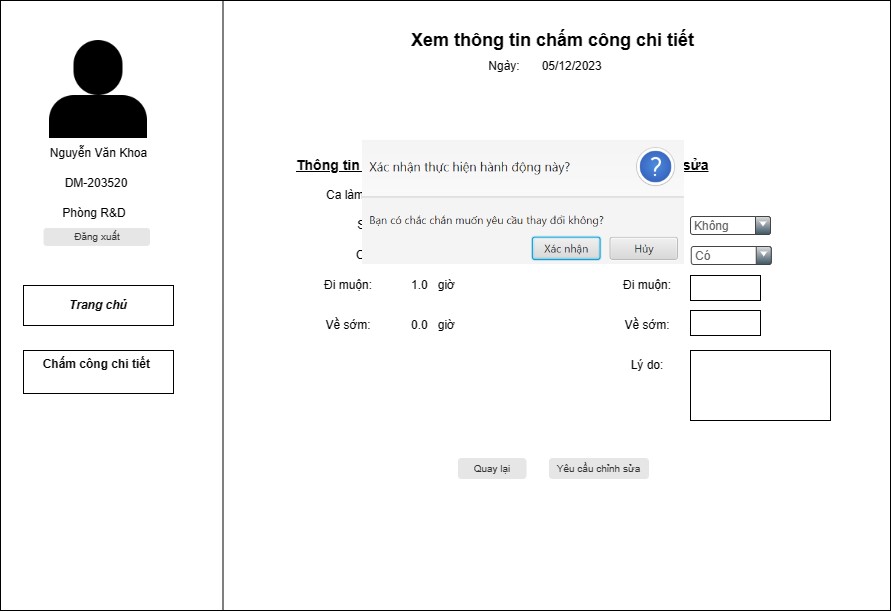
A screenshot of a computer

Description automatically generated

\* Đặc tả điều khiển

|  |  |  |
| --- | --- | --- |
| **Điều khiển** | **Hành vi** | **Chức năng** |
| Hình đại diện | Initial | Hiển thị ảnh đại diện của người dùng |
| Text Tên người dùng | Initial | Hiển thị tên người dùng |
| Text Mã nhân viên | Initial | Hiển thị mã nhân viên của người dùng |
| Text Tên phòng ban | Initial | Hiển thị tên phòng ban của người dùng |
| Button Đăng xuất | Click | Đăng xuất người dùng khỏi hệ thống |
| Button Trang chủ | Click | Điều hướng về Trang chủ của Trưởng đơn vị |
| Text Xem thông tin chấm công chi tiết | Initial | Hiển thị tiêu đề của màn hình |
| Text Ngày | Initial | Hiển thị ngày của bản ghi chấm công |
| Text Sáng | Initial | Hiển thị thông tin chấm công buổi sáng |
| Text Chiều | Initial | Hiển thị thông tin chấm công buổi chiều |
| Text Đi muộn | Initial | Hiển thị số giờ đi muộn |
| Text Về sớm | Initial | Hiển thị số giờ về sớm |
| ComboBox Sáng | Click | Yêu cầu chỉnh sửa thông tin chấm công buổi sáng |
| ComboBox Chiều | Click | Yêu cầu chỉnh sửa thông tin chấm công buổi chiều |
| TextField Đi muộn | Type | Yêu cầu chỉnh sửa số giờ đi muộn |
| TextField Về sớm | Type | Yêu cầu chỉnh sửa số giờ về sớm |
| TextField Lý do | Type | Lý do yêu cầu chỉnh sửa |
| Button Quay lại | Click | Trở về màn hình OfficerHomeView |
| Button Yêu cầu chỉnh sửa | Click | Gửi yêu cầu chỉnh sửa |

### Đặc tả màn hình Hiển thị thông báo xác nhận

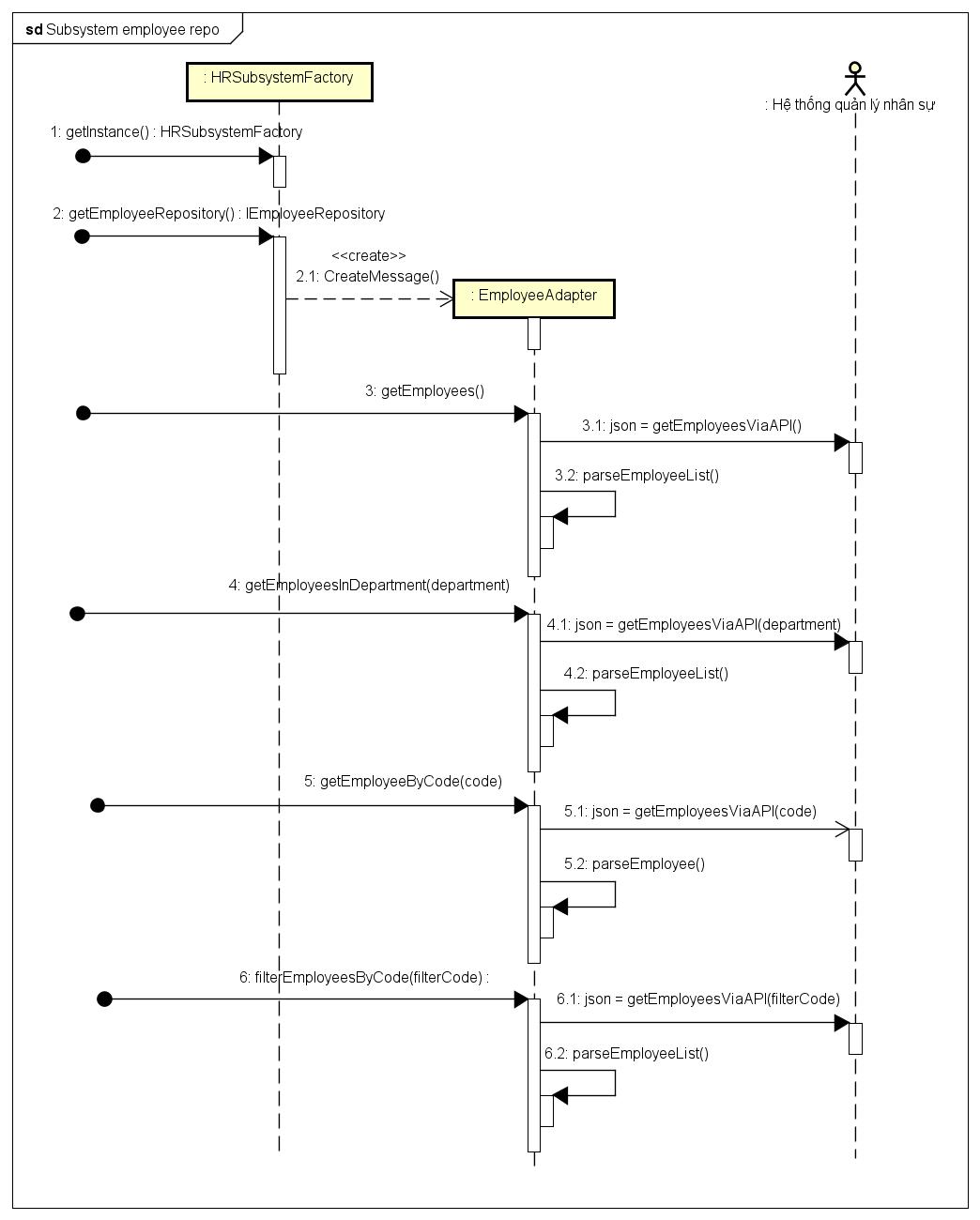


\* Đặc tả điều khiển

|  |  |  |
| --- | --- | --- |
| **Điều khiển** | **Hành vi** | **Chức năng** |
| Button Xác nhận | Click | Xác nhận gửi yêu cầu chỉnh sửa |
| Button Hủy | Click | Hủy gửi yêu cầu chỉnh sửa |

## Biểu đồ trình tự cho Subsystem

\*Biểu đồ trình tự liên quan đến EmployeeAdapter



## Biểu đồ lớp cho Subsystem

A diagram of a company

Description automatically generated

# BIỂU ĐỒ TRÌNH TỰ VÀ LỚP MỨC THIẾT KẾ (BÀI 5)

## Thiết kế biểu đồ trình tự mức thiết kế

### Use case “Xem chấm công của bản thân nhân viên văn phòng”

\* Tạo ra các đối tượng IUserRepository, IOfficerAttendanceRepository

A diagram of a company

Description automatically generatedA diagram of a program

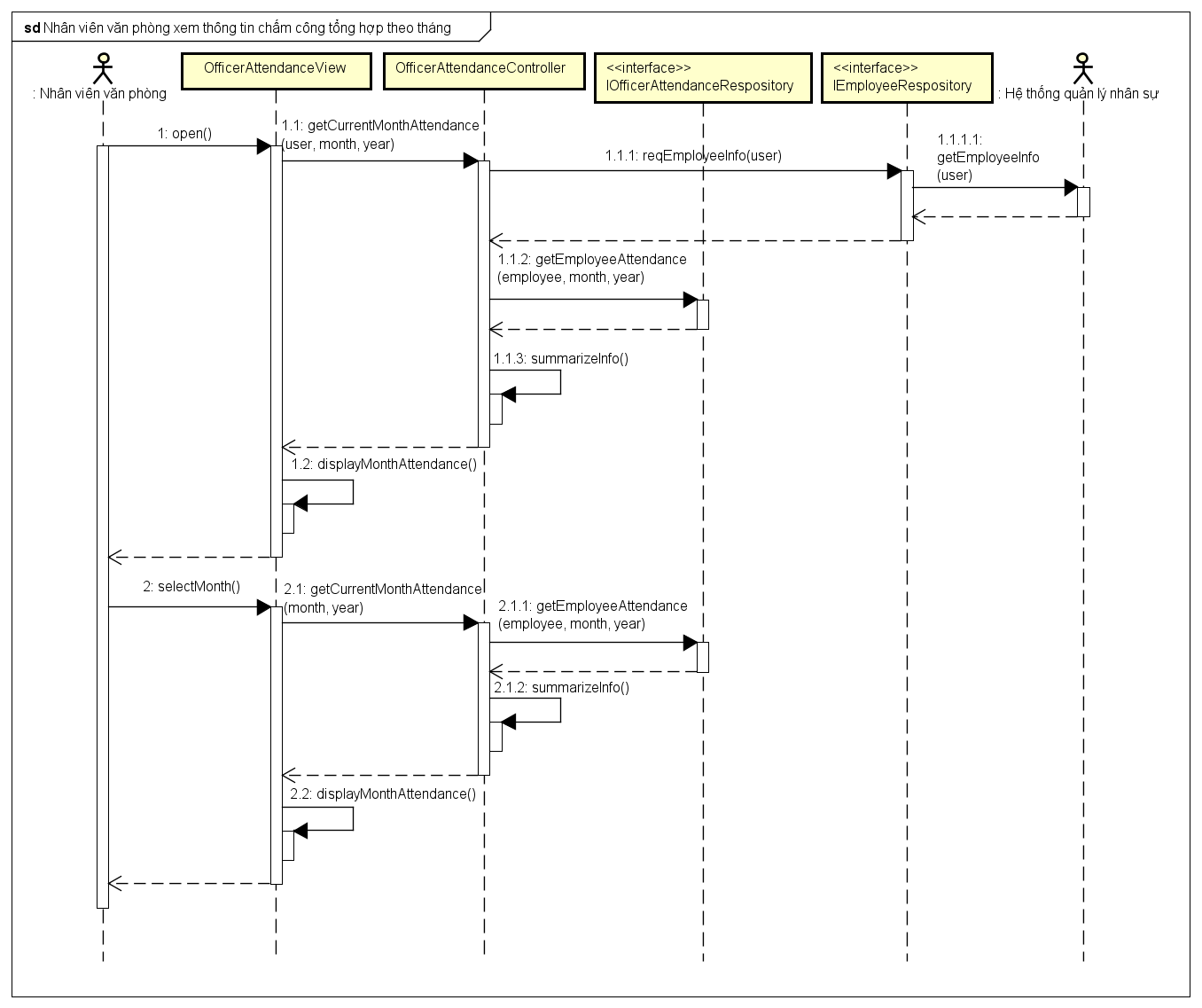
Description automatically generated

\* Main Scenario: Nhân viên văn phòng xem chi tiết ngày chấm công

A screenshot of a computer

Description automatically generated

\* Scenario 2: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo tháng



\* Scenario 3: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo quý

A screenshot of a computer screen

Description automatically generated

\* Scenario 4: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo năm

A diagram of a project

Description automatically generated

## Biểu đồ lớp mức thiết kế

### Use case “Xem chấm công của bản thân nhân viên văn phòng”

A diagram of a computer system

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of a company

Description automatically generated

# CẢI THIỆN BIỂU ĐỒ TRÌNH TỰ VÀ LỚP MỨC THIẾT KẾ (BÀI 6)

## Cách chỉnh sửa và ý nghĩa mang lại

### Áp dụng mẫu thiết kế Singleton

Mẫu thiết kế Singleton là một trong những mẫu thiết kế phổ biến trong lập trình, được sử dụng trong những lớp **UserService, UserFactory, AttendanceFactory, HRSubsystemFactory**

để đảm bảo rằng:

* Một lớp chỉ có duy nhất một thể hiện.
* Có thể dễ dàng truy cập toàn cục đối tượng.
* Không cần phải quan tâm cách khởi tạo đối tượng.

### Áp dụng mẫu thiết kế Factory

Mẫu thiết kế Singleton là một trong những mẫu thiết kế phổ biến trong lập trình, được sử dụng trong những lớp **IEmployeeRepository, IDepartmentRepository, IOfficerAttendanceReposito-ry, IUserRepository** với các lớp Factory gồm **UserFactory, AttendanceFactory, HRSubsystem-Factory** được nhằm mục đích:

* **Tạo đối tượng mà không cần biết chi tiết cụ thể**: Mẫu Factory giúp ẩn đi logic cụ thể của việc tạo đối tượng. Khi sử dụng một phương thức tạo đối tượng từ Factory, bạn không cần biết chi tiết cách đối tượng đó được khởi tạo hoặc được xử lý bên trong.
* **Đảm bảo tuân thủ nguyên tắc "Open/Closed"**: Mẫu Factory giúp đảm bảo rằng khi thêm một triển khai mới của interface mà không cần phải sửa đổi mã nguồn hiện tại. Thay vào đó, chỉ cần thêm một lớp mới vào Factory.
* **Tăng sự linh hoạt của hệ thống**: Factory có thể được cấu hình để tạo ra các đối tượng thuộc các lớp con khác nhau dựa trên điều kiện nào đó. Điều này tăng sự linh hoạt và tái sử dụng mã nguồn.
* **Giảm sự phụ thuộc với đối tượng cụ thể**: Khi sử dụng một Factory để tạo đối tượng thì không cần phải biết chi tiết cụ thể cách khởi tạo của đối tượng đó. Điều này giảm sự phụ thuộc và giúp giữ cho mã nguồn linh hoạt hơn.

### Áp dụng nguyên tắc Cohesion and Coupling

Các lớp được phân lại thành các gói **pages, user, report, attendance, hrsubsystem** với mục đính tăng tính kết dính cho các lớp trong một gói, và giảm sự phụ thuộc giữa các gói với nhau. Những lớp có cùng một nhiệm vụ, cùng một nghiệp vụ thì được gom nhóm lại một gói, và mỗi gói lại có một nhiệm vụ khác nhau.

Cohesion giúp tăng tính đóng gói (encapsulation) và giảm sự phụ thuộc giữa các module. Nó làm cho mã nguồn dễ đọc, dễ bảo trì, và dễ kiểm thử. Cohesion cao cũng giúp tái sử dụng mã nguồn dễ dàng hơn.

Coupling giúp tăng tính linh hoạt và tái sử dụng của hệ thống. Khi các thành phần độc lập và không phụ thuộc quá mức lớn vào nhau, sự thay đổi trong một thành phần không ảnh hưởng quá lớn đến các thành phần khác. Điều này làm cho hệ thống linh hoạt và dễ mở rộng, bảo trì.

### Áp dụng nguyên lý SOLID

* *S - Một class chỉ nên giữ một trách nhiệm duy nhất*

Tất cả các lớp đều chỉ thực hiện một trách nhiệm duy nhất, như các lớp Model **OfficerAtten-dance, OfficerAndAttendance, User, Employee, Department** chỉ chứa dữ liệu về đối tượng, trong khi các lớp **EmployeeAdapter, DepartmentAdapter, SqliteOfficerAtte-ndanceRepository, SqliteUserRepository** phụ trách việc thao tác lưu trữ đối tượng trong database hoặc tương tác với hệ thống bên ngoài để lấy đối tượng, …

* *O - Có thể thoải mái mở rộng 1 module, nhưng hạn chế sửa đổi bên trong module đó*

Phần lớn các hàm đều được thiết kế theo dạng subcribe, unsubscribe để tránh việc thay đổi logic hàm, như các hàm **createRepository** và **registerRepository** trong lớp **UserFactory, AttendanceFactory**

* *L - Trong một chương trình, các object của class con có thể thay thế class cha mà không làm thay đổi tính đúng đắn của chương trình*

Không có lớp nào sử dụng kế thừa nên xem như hợp nguyên tắc.

* *I - Thay vì dùng 1 interface lớn, ta nên tách thành nhiều interface nhỏ, với nhiều mục đích cụ thể*

Subsystem được tách thành **IEmployeeRepository, IDepartmentRepository** nhằm để thực hiện truy vấn với từng kiểu dữ liệu lần lượt là **Employee, IDepartment** thay vì dồn vào 1 lớp duy nhất.

* *D - Các module cấp cao không nên phụ thuộc vào các module cấp thấp. Cả 2 nên phụ thuộc vào abstraction. Interface (abstraction) không nên phụ thuộc vào chi tiết, mà ngược lại. (Các class giao tiếp với nhau thông qua interface, không phải thông qua implementation.)*

Thay vì phải để mã phụ thuộc vào **EmployeeAdapter, DepartmentAdapter, SqliteOfficerAtte-ndanceRepository, SqliteUserRepository** thì cho những đoạn mã phụ thuộc thay thế lần lượt vào **IEmployeeRe-pository, IDepartmentRepository, IOfficerAttendanceRepository, IUserRepository** sẽ mang lại những điều sau:

* **Định hình Cấu trúc Hệ thống**: Nguyên tắc Dependency Convention giúp định hình cấu trúc của hệ thống bằng cách quy ước về cách các thành phần phụ thuộc lẫn nhau. Điều này tạo ra một cấu trúc tự nhiên và dễ hiểu, giúp người đọc mã nguồn dễ dàng theo dõi và hiểu cấu trúc tổng thể của hệ thống.
* **Giảm Độ phức tạp**: Nguyên tắc Dependency Convention giúp giảm độ phức tạp của hệ thống bằng cách hạn chế sự phụ thuộc giữa các thành phần (thay vào đó là phụ thuộc vào interface). Điều này giúp giảm nguy cơ xung đột và khó khăn trong quá trình phát triển và bảo trì.
* **Dễ Bảo trì và Nâng cấp**: Nguyên tắc Dependency Convention giúp việc bảo trì và nâng cấp trở nên dễ dàng hơn. Các thay đổi có thể được thực hiện mà không làm ảnh hưởng đến toàn bộ hệ thống, chỉ bằng cách thay đổi implement này bằng implement khác của interface, và người phát triển có thể tập trung vào các thành phần cụ thể mà họ đang làm việc.
* **Tăng Tính tái sử dụng**: Dependency Convention thúc đẩy việc sử dụng lại mã nguồn bằng cách tạo ra các thành phần độc lập, có thể được tái sử dụng trong các phần khác nhau của hệ thống hoặc trong các dự án khác.
* **Tăng Tính Mô đun hóa**: Nguyên tắc này hỗ trợ tính mô đun hóa của hệ thống, trong đó mỗi mô đun (hoặc thành phần) có trách nhiệm và chức năng cụ thể. Điều này giúp tạo ra các mô đun độc lập, dễ kiểm thử và duy trì.

## Thiết kế biểu đồ trình tự mức thiết kế

### Use case “Xem chấm công của bản thân nhân viên văn phòng”

\* Tạo ra các đối tượng IUserRepository, IOfficerAttendanceRepositoryA diagram of a company

Description automatically generatedA diagram of a program

Description automatically generated

\* Main Scenario: Nhân viên văn phòng xem chi tiết ngày chấm công

A screenshot of a computer

Description automatically generated

\* Scenario 2: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo tháng

A screenshot of a computer screen

Description automatically generated

\* Scenario 3: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo quý

A screenshot of a computer screen

Description automatically generated

\* Scenario 4: Nhân viên văn phòng xem thông tin chấm công tổng hợp theo năm

A diagram of a project

Description automatically generated

## Biểu đồ lớp mức thiết kế

### Use case “Xem chấm công của bản thân nhân viên văn phòng”

A diagram of a computer system

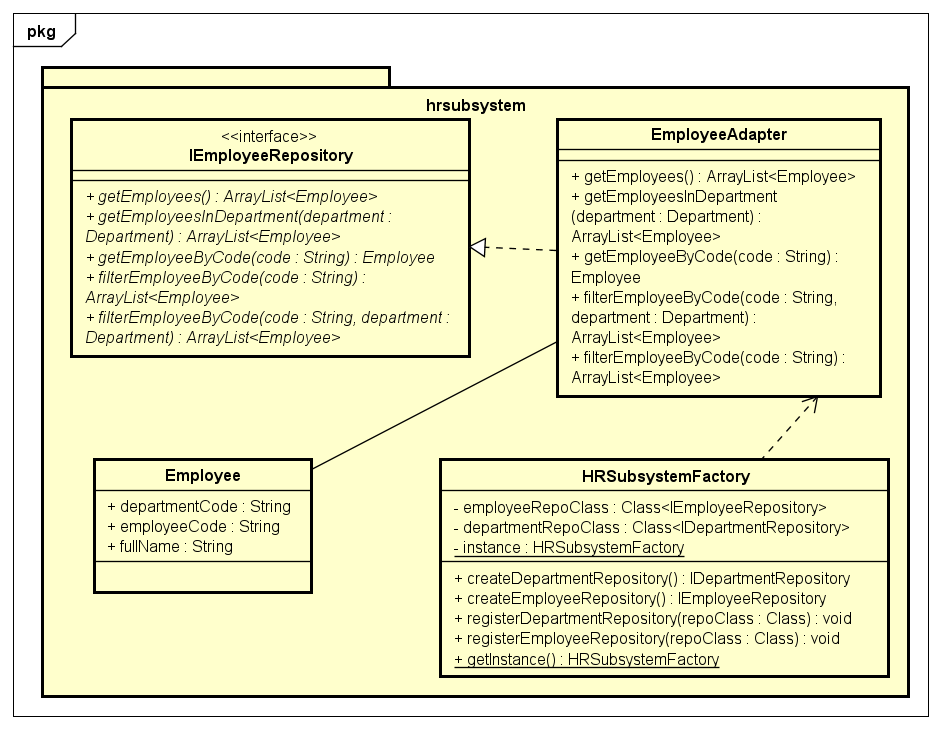
Description automatically generated

A screenshot of a computer

Description automatically generated

A yellow paper with text

Description automatically generated



# Lập trình và kiểm thử đơn vị

## Thiết kế test case cho module: SqliteOfficerAttendanceRespository

### Mô tả module:

Module chứa các hàm liên quan tới việc truy vấn các thông tin về chấm công của nhân viên văn phòng từ cơ sở dữ liệu. Bên trong class này có sử dụng các dependency là các instance của các Class có chức năng làm việc với cơ sở dữ liệu, cung cấp các phương thức để làm các nghiệp vụ của Class SqliteOfficerAttendanceRespository.

### Thiết kế test case cho phương thức getAttendancesOfEmployee():

**Tên phương thức**: getAttendancesOfEmployee

**Kiểu trả về:** ArrayList<OfficerAttendance>

Danh sách tham số: Tài khoản của người dùng (User) user, đối tượng (Employee) employee, tháng bắt đầu (int) month,năm bắt đầu (int) year, số tháng (int) monthCount

**Chi tiết**: Hàm nhận vào các tham số trên và trả về các log chấm công của nhân viên employee bắt đầu từ tháng month năm year và kéo dài trong monthCount tháng.

#### Kỹ thuật sử dụng kiểm thử hộp đen và phân vùng tương đương.

Kiểm thử hộp đen: Không cần biết logic bên trong code, chỉ cần biết các thông số về ngày tháng và nhân viên, kết hợp với giao diện người dùng để kiểm thử.

Phân vùng tương đương cho phép phân chia tập hợp các điều kiện kiểm tra thành một phân vùng nên được coi là giống nhau. Để thiết kế được các test case cho phương thức này, đầu tiên xác định các vùng tương đương của đầu vào.

**Vùng tương đương:**

+ Vùng 1: employee tồn tại, tháng month hợp lệ (1-12),số tháng monthcount hợp lệ (nguyên không âm), năm và tháng nhỏ hơn thời gian hiện tại

+ Vùng 2: employee không tồn tại

+ Vùng 3: month<1 hoặc month >12, monthcount <0

#### Thiết kế test case.

**Test case Query\_Success**

|  |  |
| --- | --- |
| **Test case #:** 1.1 | **Tên test case:** Query |
| **Hệ thống:** Hệ thống quản lý chấm công | **Hệ thống con:** Hệ thống quản lý nhân sự |
| **Thiết kế bởi:** Bùi Trọng Đức 20200157 | **Ngày thiết kế:** 20/12/2023 |
| **Thực thi bởi:** SqliteOfficerAttendanceRespository | **Ngày thực thi:** 20/12/2023 |
| **Mô tả ngắn gọn:** Kiểm tra phương thức truy vấn cơ sở dữ liệu với kết quả trả về thành công |  |

|  |
| --- |
| **Tiền điều kiện:**  Hệ thống đang hiển thị giao diện Xem thông tin chấm công nhân viên văn phòng  Dữ liệu đầu vào hợp lệ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước | Hành động | Phản hồi mong muốn từ hệ thống | Pass/Fail | Comment |
| 1 | Người dùng lựa chọn loại thống kê là tháng | Hệ thống hiển thị loại thống kê là tháng | Pass |  |
| 2 | Người dùng chọn tháng thống kê (12) | Hệ thống hiển thị tháng thống kê là 12 | Pass |  |
| 3 | Người dùng chọn năm thống kê (2023) | Hệ thống hiển thị năm thống kê là 2023 | Pass |  |
| 4 | Người dùng ấn nút Thống kê | Hệ thống hiển thị các log chấm công trong tháng 12 năm 2023 | Pass |  |

|  |
| --- |
| **Trạng thái của hệ thống sau khi chạy hàm:**  Hệ thống hiển thị các thông tin người dùng đã chọn và hiện thị các log chấm công |

**Test case Query\_InvalidData**

|  |  |
| --- | --- |
| **Test case #:** 1.2 | **Tên test case:** Query |
| **Hệ thống:** Hệ thống quản lý chấm công | **Hệ thống con:** Hệ thống quản lý nhân sự |
| **Thiết kế bởi:** Bùi Trọng Đức 20200157 | **Ngày thiết kế:** 20/12/2023 |
| **Thực thi bởi:** SqliteOfficerAttendanceRespository | **Ngày thực thi:** 20/12/2023 |
| **Mô tả ngắn gọn:** Kiểm tra phương thức truy vấn cơ sở dữ liệu với kết quả trả về không thành công |  |

|  |
| --- |
| **Tiền điều kiện:**  Hệ thống đang hiển thị giao diện Xem thông tin chấm công nhân viên văn phòng  Dữ liệu đầu không hợp lệ (month = 1, year = 2024) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước | Hành động | Phản hồi mong muốn từ hệ thống | Pass/Fail | Comment |
| 1 | Người dùng lựa chọn loại thống kê là tháng | Hệ thống hiển thị loại thống kê là tháng | Pass |  |
| 2 | Người dùng chọn tháng thống kê (12) | Hệ thống hiển thị tháng thống kê là 12 | Pass |  |
| 3 | Người dùng chọn năm thống kê (2024) | Hệ thống hiển thị năm thống kê là 2024 | Pass |  |
| 4 | Người dùng ấn nút Thống kê | Hệ thống thông báo thời điểm thống kê chưa xảy ra | Pass |  |

|  |
| --- |
| **Trạng thái của hệ thống sau khi chạy hàm:**  Hệ thống trở lại trạng thái ban đầu. |

#### Kỹ thuật sử dụng: kiểm thử hộp trắng và bảng quyết định.

Kiểm thử hộp trắng (độ đo c1): bao hàm tất cả các trường hợp xảy ra trong đoạn code.

Sử dụng Control Flow test:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Valid month, year, monthCount | Existing Employee | Existing Attendancelog | Kết quả |
| 1 | Có | Có | Có | Danh sách OfficerAttendances |
| 2 | Không | Có | Có | Exception |
| 3 | Có | Không | Có | Trả về Null |
| 4 | Có | Có | Không | Trả về Null |