

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Lớp học phần: Nhóm 08 – D22

Nhóm bài tập: 08

Tên đề tài nhóm: 17. Hệ thống quản lý cửa hàng cho thuê trang phục biểu diễn

- Họ và tên:
- Đinh Bá Đạt – B21DCCN210
 - Nguyễn Tiến Đạt – B22DCPT052
 - Phạm Đức Hoàn – B22DCAT124
 - Nguyễn Đức Trung – B22DCCN871**

Tên module: Thống kê trang phục được mượn nhiều

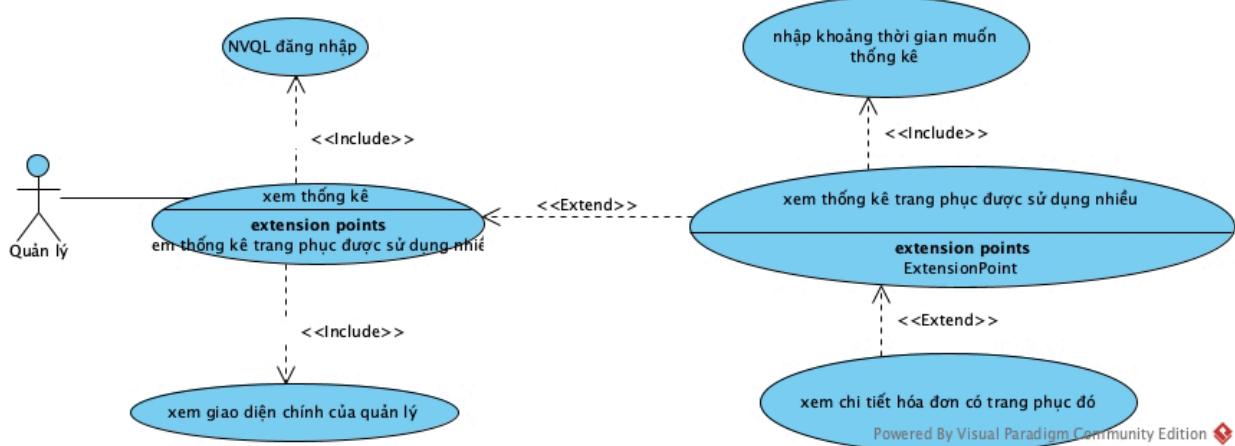
Hà Nội, tháng 05 năm 2025

MỤC LỤC

1. UC chi tiết “Thống kê trang phục được mượn nhiều”:	2
2. Kịch bản chuẩn - Scenario:	3
3. Biểu đồ lớp thực thể pha phân tích module “Thống kê trang phục được mượn nhiều”	5
4. Biểu đồ lớp đầy đủ pha phân tích module “Thống kê trang phục được mượn nhiều”	7
5. Biểu đồ tuần tự pha phân tích	9
6. Biểu đồ thiết kế lớp thực thể	9
7. Biểu đồ thiết kế CSDL	11
8. Thiết kế các giao diện và biểu đồ lớp thiết kế chi tiết đầy đủ	12
Thiết kế biểu đồ lớp chi tiết	16
9. Biểu đồ tuần tự thiết kế	18
10. Test plan và test case chuẩn cho test hộp đen	18

1. UC chi tiết “Thống kê trang phục được mượn nhiều”:

- Thống kê trang phục được mượn nhiều:
 - QL chọn menu thống kê trang phục được mượn nhiều
 - Nhập ngày bắt đầu, ngày kết thúc thống kê -> hệ thống hiển thị danh sách các trang phục được mượn nhiều theo dạng bảng, mỗi dòng tương ứng với một trang phục với đầy đủ thông tin: mã, tên, kiểu, thể loại, cột tổng số lượt được mượn, cột tổng số tiền thu được -> Xếp theo thứ tự giảm dần của cột tổng số lượt mượn, tiếp theo là giảm dần của cột tổng số tiền thu được.
 - QL click vào 1 dòng của 1 trang phục -> hệ thống hiện lên chi tiết hóa đơn có trang phục đó đã mượn, mỗi hóa đơn trên 1 dòng: id, tên khách mượn, ngày giờ mượn, ngày giờ trả, tổng số tiền.



Mô tả các UC:

- UC NVQL đăng nhập: cho phép NVQL đăng nhập vào hệ thống
- UC xem giao diện của quản lý: cho phép NVQL xem giao diện của quản lý để chọn menu thống kê
- UC nhập khoảng thời gian muốn thống kê cho phép NVQL nhập khoảng thời gian bắt đầu và kết thúc thống kê

- UC xem danh sách trang phục được mượn nhiều cho phép NVQL xem danh sách trang phục được mượn nhiều theo dạng bảng
- UC xem chi tiết hóa đơn có trang phục đó cho phép NVQL xem chi tiết hóa đơn có trang phục đó sau khi click vào 1 trang phục cụ thể

2. Kịch bản chuẩn - Scenario:

Scenario	Thông kê trang phục được mượn nhiều
Actor	NVQL
Pre-condition	NVQL có account của NVQL
Post-condition	NVQL xem được thông kê trang phục được mượn nhiều
Main event	<ol style="list-style-type: none"> 1. NVQL A login vào hệ thống với username = manager, password = manager để xem thông kê trang phục được sử dụng nhiều 2. HT hiện giao diện chính của NVQL, có lựa chọn xem thông kê trang phục được mượn nhiều 3. NVQL A chọn chức năng xem thông kê trang phục được mượn nhiều 4. HT hiện giao diện xem thông kê: <ul style="list-style-type: none"> - Ô nhập: ngày bắt đầu, kết thúc thông kê - Nút: thông kê 5. NVQL nhập ngày bắt đầu = 1/1/2025, ngày kết thúc = 2/2/2025 6. HT hiển thị danh sách các trang phục được mượn nhiều, xếp theo thứ tự giảm dần của cột tổng số lượt mượn, tiếp theo là giảm dần của cột tổng số tiền thu được:

	<ul style="list-style-type: none"> - Ngày bắt đầu: 1/1/2025, ngày kết thúc: 2/2/2025, nút thống kê <table border="1"> <thead> <tr> <th>Mã</th><th>Tên</th><th>Kiểu</th><th>Tổng số lượt được mượn</th><th>Tổng số tiền thu được</th></tr> </thead> <tbody> <tr> <td>01</td><td>Áo dài</td><td>Màu đỏ size S</td><td>3</td><td>30</td></tr> <tr> <td>02</td><td>Sườn xám</td><td>Màu xanh size M</td><td>2</td><td>20</td></tr> <tr> <td>03</td><td>Kimono</td><td>Màu tím size L</td><td>1</td><td>10</td></tr> </tbody> </table> <ul style="list-style-type: none"> - Nút next <ol style="list-style-type: none"> QL click vào dòng số 1, xem chi tiết thống kê sườn xám HT hiện chi tiết các hóa đơn có trang phục đó đã mượn: <table border="1"> <thead> <tr> <th>id</th><th>Tên khách mượn</th><th>Ngày mượn</th><th>Ngày trả</th><th>Tổng số tiền</th></tr> </thead> <tbody> <tr> <td>1</td><td>Nguyễn A</td><td>4/1/2025</td><td>5/1/2025</td><td>20</td></tr> <tr> <td>2</td><td>Vũ B</td><td>9/1/2025</td><td>10/1/2025</td><td>10</td></tr> </tbody> </table>	Mã	Tên	Kiểu	Tổng số lượt được mượn	Tổng số tiền thu được	01	Áo dài	Màu đỏ size S	3	30	02	Sườn xám	Màu xanh size M	2	20	03	Kimono	Màu tím size L	1	10	id	Tên khách mượn	Ngày mượn	Ngày trả	Tổng số tiền	1	Nguyễn A	4/1/2025	5/1/2025	20	2	Vũ B	9/1/2025	10/1/2025	10
Mã	Tên	Kiểu	Tổng số lượt được mượn	Tổng số tiền thu được																																
01	Áo dài	Màu đỏ size S	3	30																																
02	Sườn xám	Màu xanh size M	2	20																																
03	Kimono	Màu tím size L	1	10																																
id	Tên khách mượn	Ngày mượn	Ngày trả	Tổng số tiền																																
1	Nguyễn A	4/1/2025	5/1/2025	20																																
2	Vũ B	9/1/2025	10/1/2025	10																																
Exception	<ol style="list-style-type: none"> HT hiển thị không có trang phục được mượn trong khoảng thời gian đó QL click reset rồi nhập lại khoảng thời gian 																																			

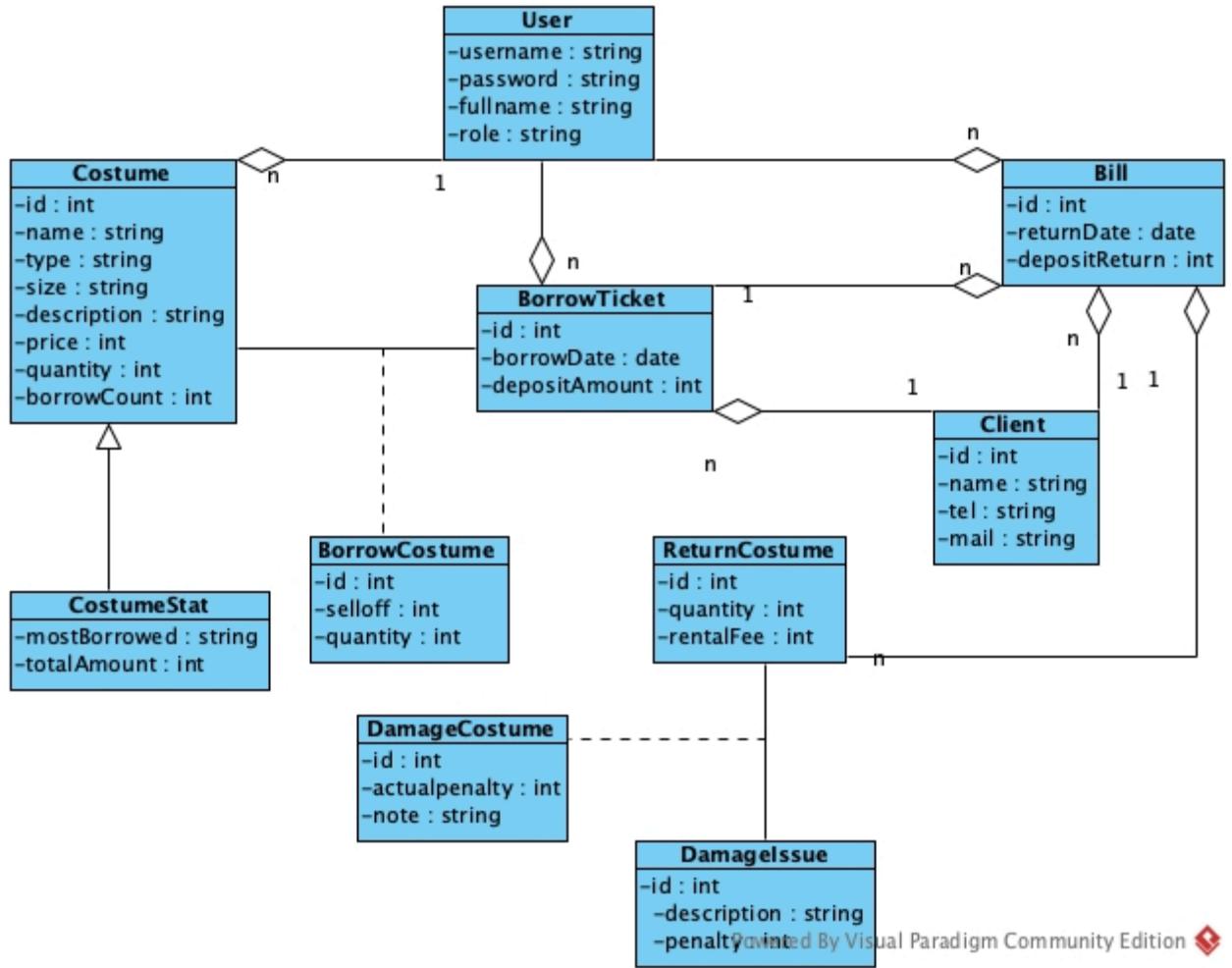
3. Biểu đồ lớp thực thể pha phân tích module “Thống kê trang phục được mượn nhiều”

Trích lớp thực thể:

- User: id, username, password, fullname, role
- Client: id, name, tel, mail
- Costume: id, name, type, size, description, price, quantity
- CostumeStat: mostBorrowed, totalAmount
- BorrowCostume: id, quantity, sellOff
- BorrowTicket: id, borrowDate, depositAmount, costumeList
- Bill: id, clientName, returnDate, costumeList, depositReturn
- ReturnCostume: id, returnDate, quantity, rentalFee
- DamageIssue: id, description, penalty
- DamageCostume: id, note, actualPenalty

Xác định quan hệ giữa các lớp:

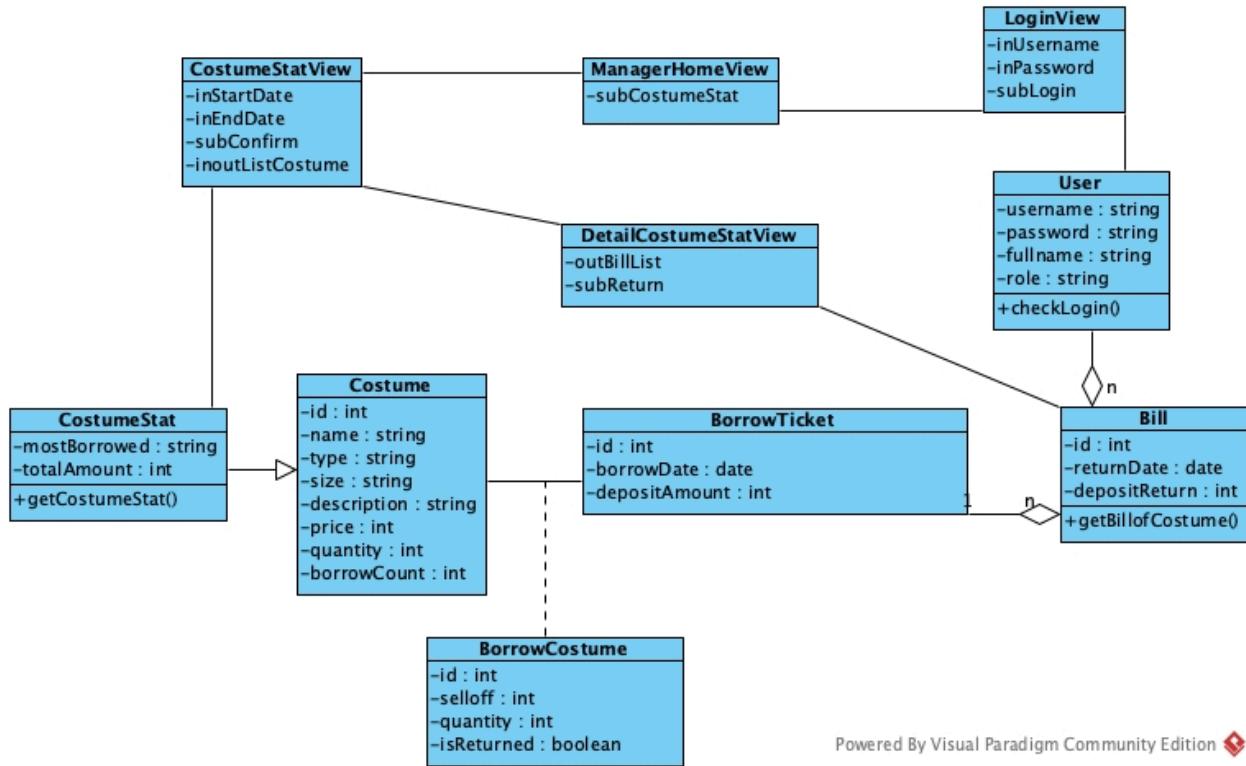
- 1 user quản lý nhiều trang phục, 1 trang phục chỉ được quản lý bởi 1 user chức vụ QL -> User - Costume là 1-n
- 1 user có thể tạo nhiều phiếu mượn, phiếu trả và hóa đơn, 1 phiếu mượn, phiếu trả và hóa đơn chỉ có thể được tạo bởi 1 user -> User - BorrowTicket, và User - Bill đều là 1-n
- 1 phiếu mượn có thể có nhiều hóa đơn do chia thành nhiều đợt trả, 1 hóa đơn chỉ có 1 phiếu mượn gốc -> BorrowTicket - Bill là 1-n
- 1 trang phục có thể nằm trong nhiều phiếu mượn, 1 phiếu mượn cũng có thể có nhiều trang phục, nên quan hệ này là n-n, cần tạo thêm 1 lớp ở giữa là BorrowCostume
- 1 khách hàng có thể mượn nhiều lần, có nhiều phiếu mượn, phiếu trả, hóa đơn, nhưng 1 phiếu mượn/trả hoặc hóa đơn chỉ có thể sở hữu bởi 1 khách hàng -> Client - BorrowTicket và Client - Bill đều là 1-n
- 1 trang phục trả có thể bị nhiều hư hại, 1 hư hại có thể thuộc về nhiều trang phục, quan hệ n-n nên cần 1 lớp trung gian là DamageCostume
- CostumeStat kế thừa 1 số thuộc tính từ lớp Costume



4. Biểu đồ lớp đầy đủ phân tích module “Thông kê trang phục được mượn nhiều”

- Truy cập hệ thống -> cửa sổ đăng nhập hiện ra -> cần lớp LoginView
 - + Nhập input cho username -> inUsername
 - + Nhập input cho password -> inPassword
 - + Submit để login -> subLogin
- Nhập xong username, password -> hệ thống cần check đăng nhập đúng không, cần thêm 1 hàm:
 - + Tên: checkLogin()
 - + Input: username, password (của lớp User)
 - + Output: boolean

- + Gán cho lớp User
- Đăng nhập thành công -> giao diện quản lý hiện lên -> cần lớp ManagerHomeView, có tối thiểu:
 - + Lựa chọn xem thống kê trang phục: subCostumeStat
- Chọn xem thống kê -> giao diện xem thống kê hiện lên -> CostumeStatView
 - + Nhập input ngày bắt đầu thống kê -> inStartDate
 - + Nhập input ngày kết thúc -> inEndDate
 - + Submit để xác nhận -> subConfirm
 - + Trả về danh sách trang phục được mượn nhiều, vừa có thể click vào 1 dòng để xem chi tiết -> inoutListCostume
- Nhập xong start date, end date rồi click confirm -> hệ thống cần truy vấn thống kê trang phục -> cần hàm getCostumeStat():
 - + Input: start date và end date
 - + Output: costume stat list
 - + Gán cho lớp CostumeStat
- Kết quả hiển thị ở lớp CostumeStatView
- Click vào 1 dòng để xem chi tiết thống kê hóa đơn có trang phục đó -> cần lớp DetailCostumeStatView
 - + Output: danh sách hóa đơn có trang phục đó -> outListBill
 - + Submit để quay về màn hình danh sách CostumeStatView -> subReturn
- Để có data để hiển thị cho DetailCostumeStatView -> cần truy vấn đến hóa đơn liên quan đến trang phục trong khoảng thời gian đó -> cần getBillOfCostume():
 - + Input: start date, end date, costume
 - + Output: list bill
 - + Gán cho lớp Bill

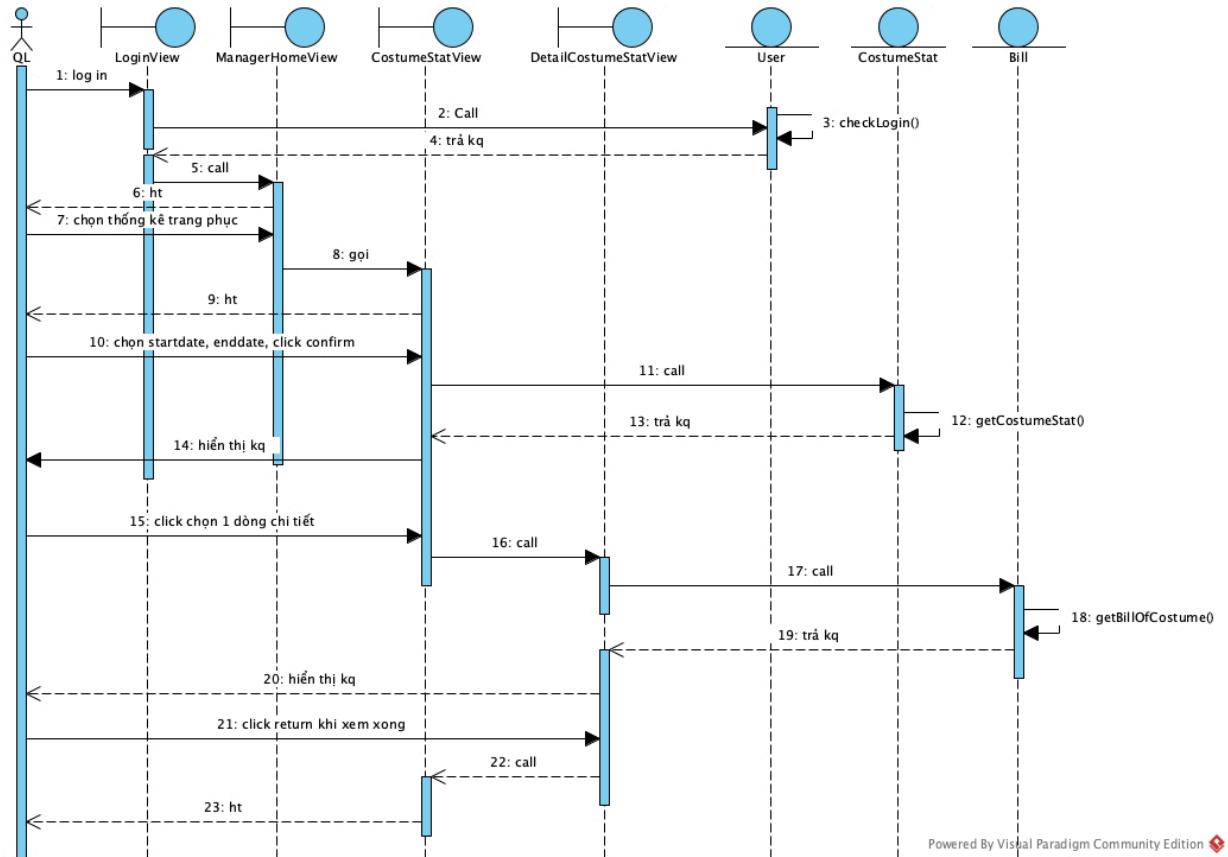


Powered By Visual Paradigm Community Edition

5. Biểu đồ tuần tự phân tích

- QL nhập username và password rồi click đăng nhập
- Lớp LoginView gọi lớp User
- Lớp User thực hiện hàm `checkLogin()`. Login thành công
- Lớp User trả kết quả về cho lớp LoginView
- Lớp LoginView gọi lớp ManagerHomeView
- Lớp ManagerHomeView hiển thị cho QL
- QL chọn thống kê trang phục
- Lớp ManagerHomeView gọi lớp CostumeStatView
- Lớp CostumeStatView hiển thị cho QL
- QL chọn ngày bắt đầu, ngày kết thúc thống kê rồi click nút confirm
- Lớp CostumeStatView gọi lớp CostumeStat
- Lớp CostumeStat thực hiện hàm `getCostumeStat()`
- Lớp CostumeStat trả kết quả về cho lớp CostumeStatView
- Lớp CostumeStatView hiển thị kết quả cho QL

15. QL click chọn 1 dòng có trang phục muốn xem chi tiết hóa đơn
16. Lớp CostumeStatView gọi lớp DetailCostumeStatView
17. Lớp DetailCostumeStatView gọi lớp Bill để lấy data
18. Lớp Bill gọi hàm getBillOfCostume()
19. Lớp Bill trả kết quả về cho lớp DetailCostumeStatView
20. Lớp DetailCostumeStatView hiển thị kết quả cho QL
21. QL click nút Return sau khi xem xong
22. Lớp DetailCostumeStatView gọi lớp CostumeStatView
23. Lớp CostumeStatView hiển thị cho QL

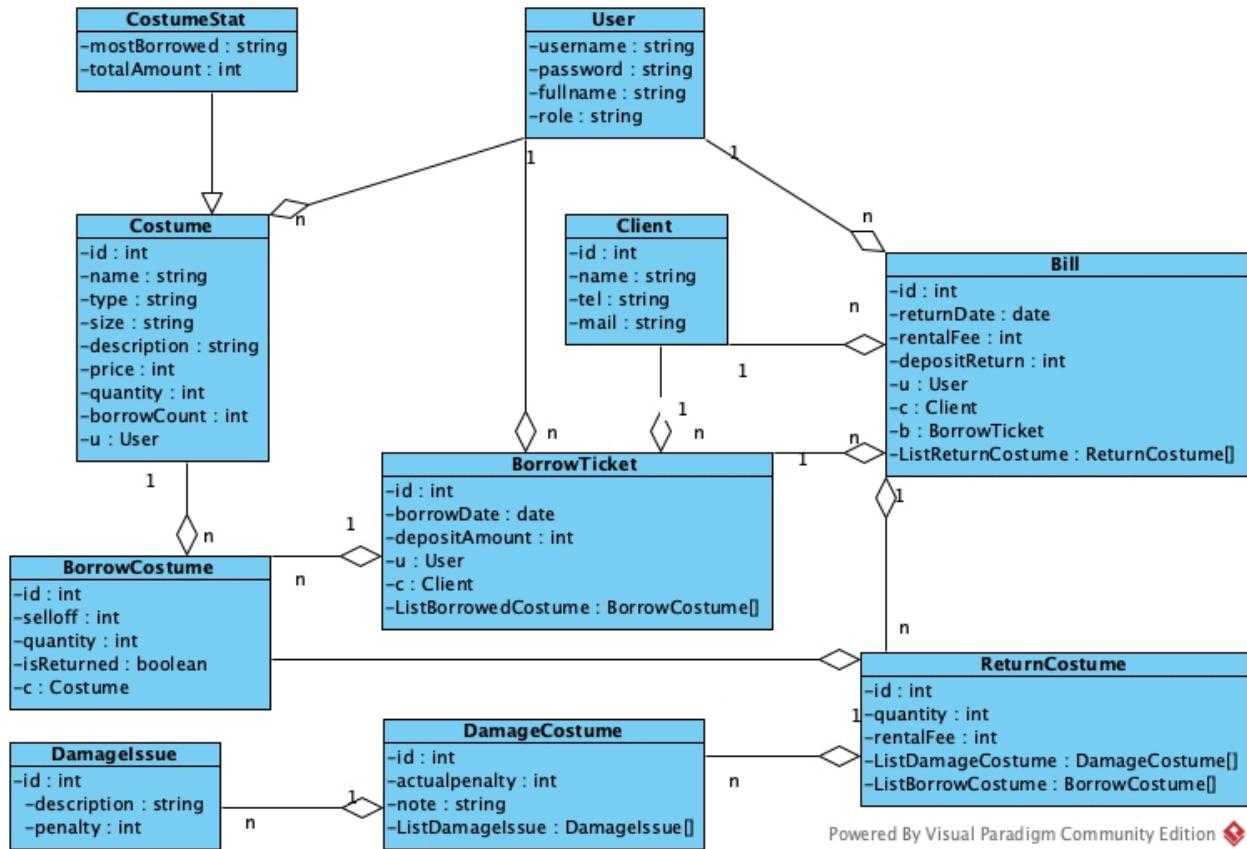


6. Biểu đồ thiết kế lớp thực thể
Chuyển quan hệ association thành aggregation hoặc composition:

- Quan hệ Costume + BorrowTicket -> BorrowCostume chuyển thành:
 - + 1-n giữa BorrowCostume-Costume do trong 1 lần mượn có thể mượn nhiều trang phục
 - + BorrowTicket-BorrowCostume là quan hệ 1-n, do trong 1 phiếu mượn có thể có nhiều trang phục được mượn
- Quan hệ ReturnCostume + DamageIssue -> DamageCostume chuyển thành:
 - + n-1 giữa DamageIssue-DamageCostume do 1 trang phục có thể có nhiều hư hại
 - + 1-n giữa ReturnCostume-DamageCostume do trong 1 nhóm trang phục có thể có nhiều trang phục bị hỏng

Bổ sung các thuộc tính quan hệ tương ứng:

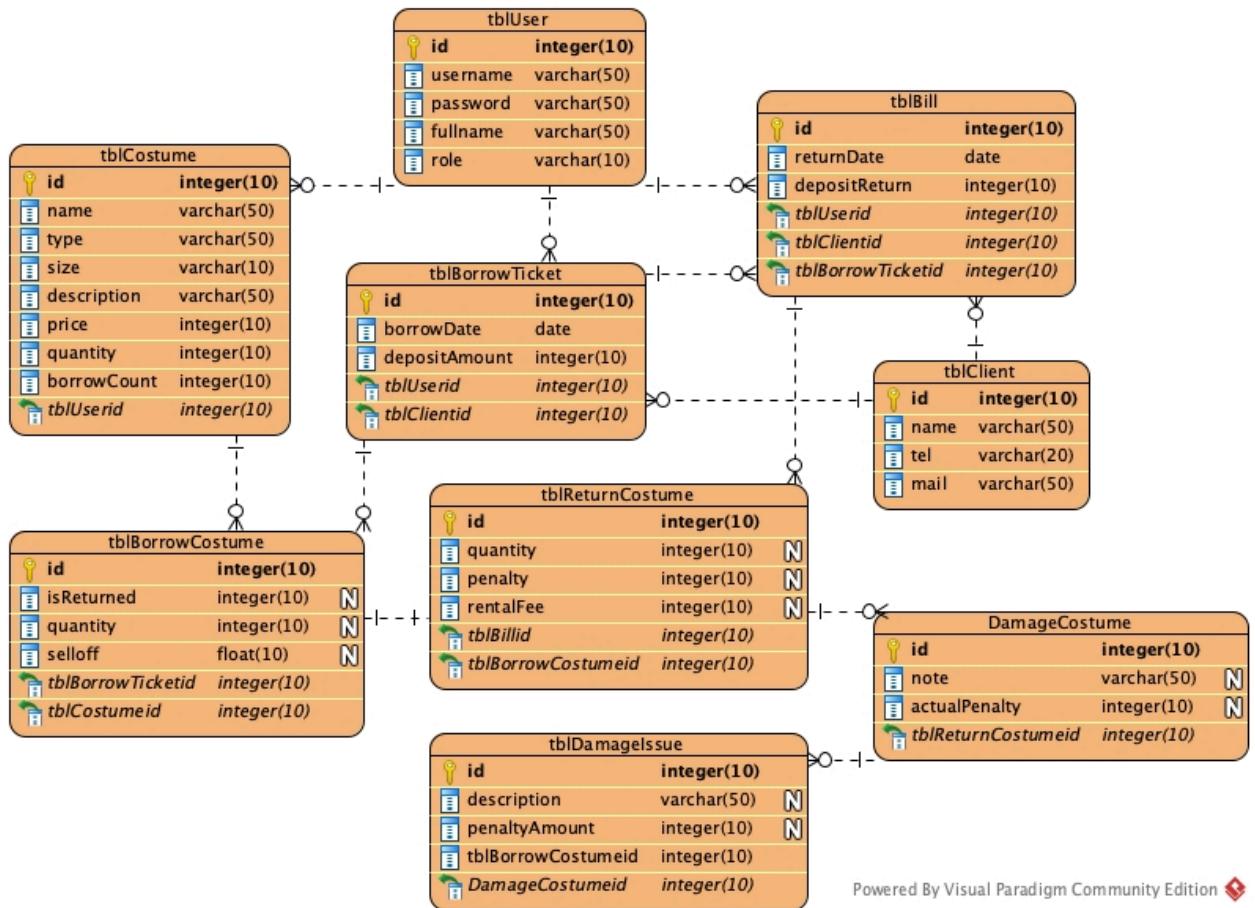
- + User là thành phần của Costume với quan hệ 1-n => 1 Costume có 1 User quản lý
- + User là thành phần của BorrowTicket với quan hệ 1-n => BorrowTicket có 1 User
- + User là thành phần của Bill với quan hệ 1-n => 1 Bill có 1 User
- + BorrowTicket là thành phần của Bill với quan hệ 1-n => 1 Bill có 1 BorrowTicket
- + Costume là thành phần của BorrowCostume với quan hệ 1-n => BorrowCostume có 1 Costume
- + BorrowCostume là thành phần của BorrowTicket với quan hệ n-1 => BorrowTicket có danh sách các BorrowCostume
- + Client là thành phần của BorrowTicket với quan hệ 1-n => BorrowTicket có 1 Client
- + Client là thành phần của Bill với quan hệ 1-n => 1 Bill có 1 Client



7. Biểu đồ thiết kế CSDL

- Mỗi thực thể \rightarrow 1 bảng dữ liệu (tbl<đến>)
- + Lớp Costume \rightarrow bảng tblCostume
- + Lớp User \rightarrow bảng tblUser
- + Lớp Client \rightarrow bảng tblClient
- + ...
- Đưa thuộc tính không phải đối tượng của lớp thực thể thành thuộc tính của bảng tương ứng
- Chuyển quan hệ giữa các lớp thực thể thành quan hệ giữa các bảng:
- + 1 tblUser - n tblBill
- + ...
- Bổ sung các thuộc tính khóa:
- + Khóa chính được thiết lập với thuộc tính id của các bảng tương ứng

- + Khóa ngoại được thiết lập cho các bảng:
 - + 1 tblUser - n tblCostume -> bảng tblCostume có FK tblUserId
 - + ...
- Loại bỏ các thuộc tính dãy xuất và lớp thông kê



8. Thiết kế các giao diện và biểu đồ lớp thiết kế chi tiết đầy đủ

- Giao diện LoginView:

Login

Username

Password

Login

- Giao diện ManagerHomeView:

Manager Home

Costume Stat

Income Stat

- Giao diện CostumeStatView:

Costume Stat

Xem thống kê trang phục được mượn nhiều trong khoảng thời gian:

Start date: End date: Confirm

Mã	Tên	Kiểu	Tổng số lượt được mượn	Tổng số tiền thu được

- Giao diện DetailCostumeStatView:

Abc

Xem thống kê chi tiết của trang phục Abc

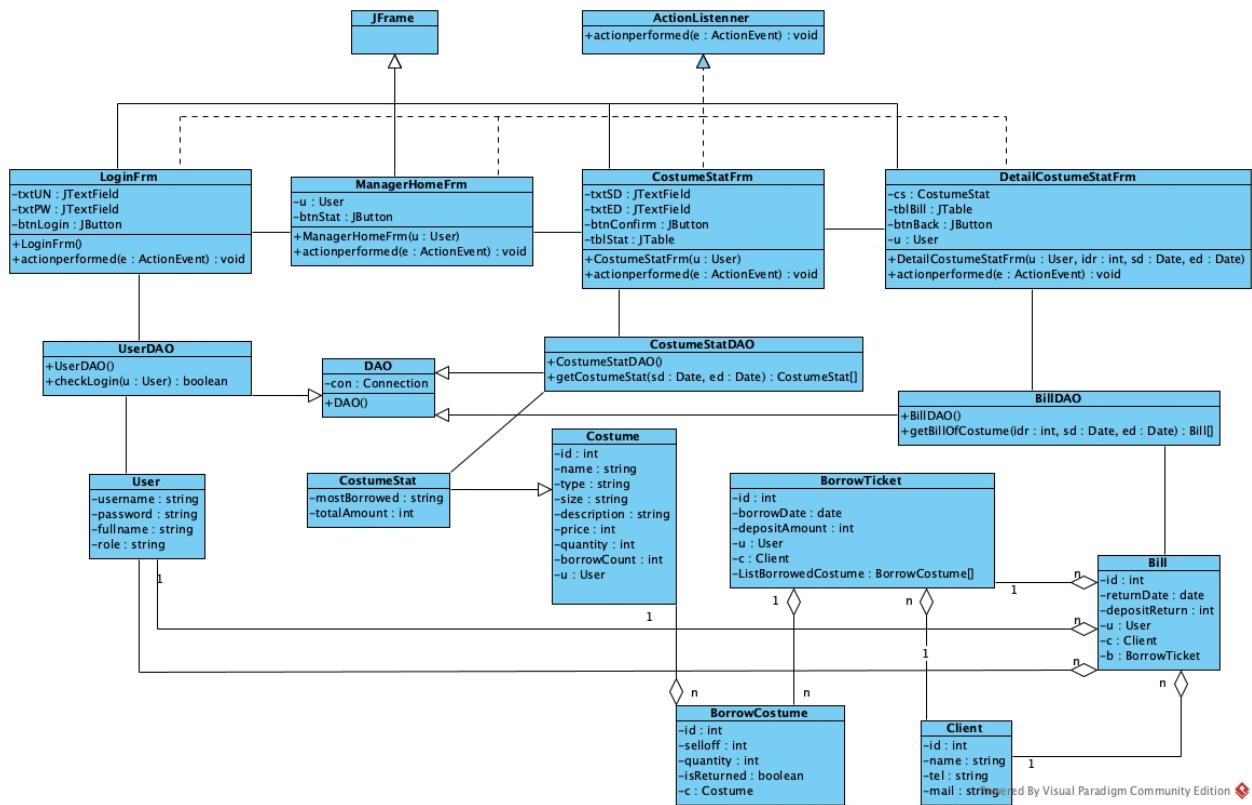
STT	ID	Tên khách	Ngày mượn	Ngày trả	Tổng số tiền

[Return](#)

Thiết kế biểu đồ lớp chi tiết

- Tầng giao diện:
 - + LoginFrm là giao diện để đăng nhập. Cần 1 ô điền để nhập username, 1 ô điền để nhập password, 1 nút để đăng nhập
 - + ManagerHomeFrm là giao diện quản lý. Cần tối thiểu nút để xem thống kê trang phục được mượn nhiều
 - + CostumeStatFrm là giao diện xem thống kê trang phục. Cần 1 ô nhập ngày bắt đầu, 1 ô nhập ngày kết thúc, 1 nút xác nhận và 1 bảng hiện kết quả
 - + DetailCostumeStatFrm là giao diện xem chi tiết thống kê. Cần 1 bảng hiện chi tiết hóa đơn có trang phục đó

- Tầng DAO:
 - + Cần con để kết nối đến CSDL, cung cấp kết nối cho các lớp DAO kế thừa
 - + UserDao truy cập đến CSDL của lớp User. Cần hàm checkLogin() để kiểm tra thông tin đăng nhập đúng hay sai
 - + CostumeStatDAO có hàm getCostumeStat() để lấy thông tin thống kê
 - + BillDAO cần truy cập đến Bill để lấy hóa đơn của trang phục
- Tầng thực thể: User, Bill, Costume, BorrowCostume, BorrowTicket, CostumeStat, Client. CostumeStat kế thừa từ Costume

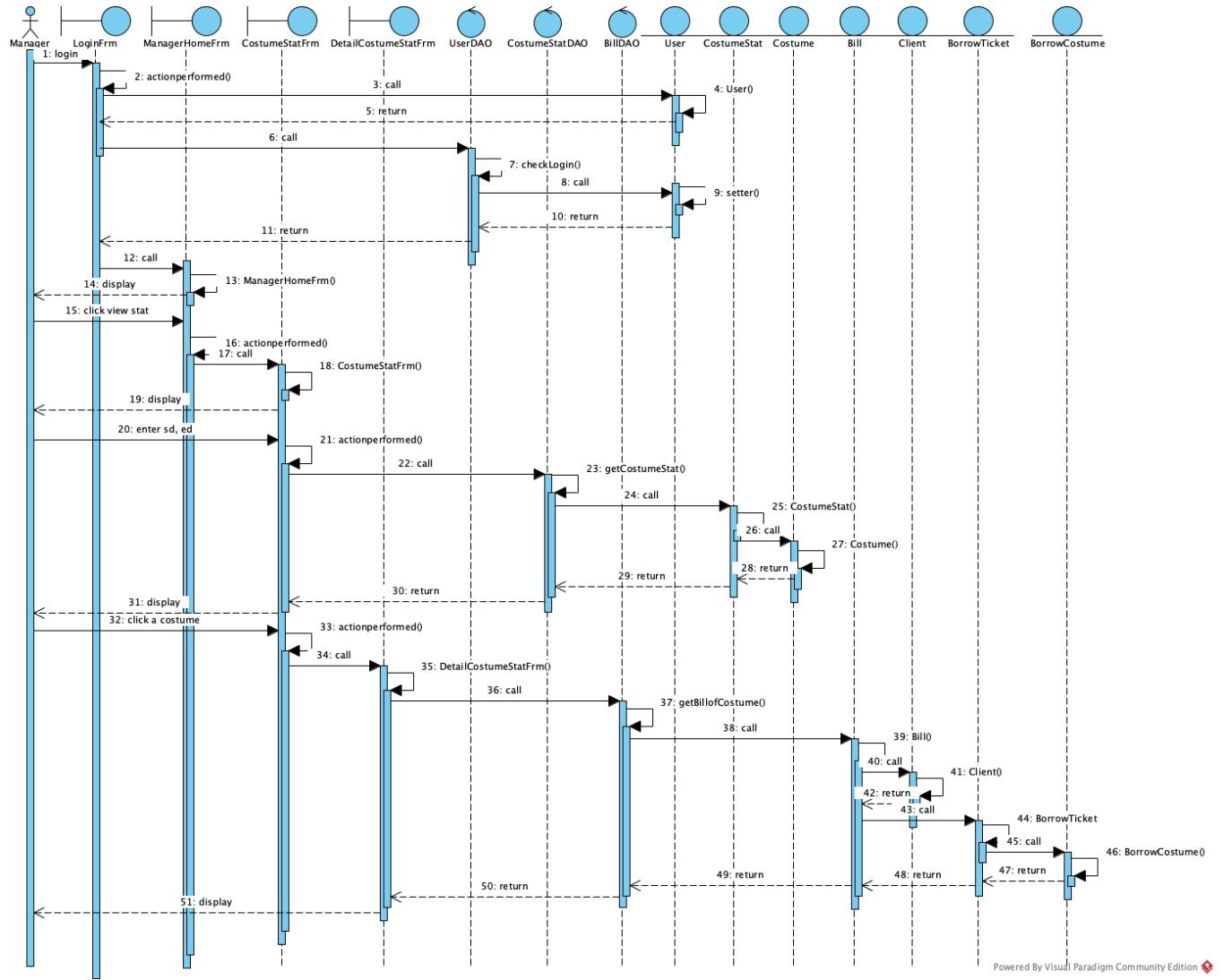


9. Biểu đồ tuần tự thiết kế

1. Quản lý nhập username, password và click vào nút đăng nhập tại LoginFrm
2. Hàm actionPerformed() của LoginFrm được gọi

3. Hàm actionPerformed() gọi lớp User để tạo User object
4. Lớp User đóng gói thông tin thành 1 User object
5. Lớp User trả User object về hàm actionPerformed()
6. Hàm actionPerformed() gọi hàm checkLogin() của lớp UserDAO
7. Hàm checkLogin() kiểm tra thông tin đăng nhập
8. Hàm checkLogin() gọi lớp User lấy thêm 2 thuộc tính name, position
9. Lớp User tự gọi hàm setName(), setPosition()
10. Lớp User trả User object về cho hàm checkLogin()
11. Hàm checkLogin() trả kết quả về actionPerformed()
12. Hàm actionPerformed() gọi lớp ManagerHomeFrm
13. ManagerHomeFrm() được gọi
14. Giao diện ManagerHomeFrm hiển thị cho quản lý
15. Quản lý click nút xem thống kê trang phục được mượn nhiều
16. Hàm actionPerformed() của lớp ManagerHomeFrm được gọi
17. Hàm actionPerformed() gọi lớp CostumeStatFrm
18. CostumeStatFrm() được gọi
19. Giao diện CostumeStatFrm được hiển thị cho quản lý
20. Quản lý nhập ngày bắt đầu, ngày kết thúc thống kê và click vào nút xem
21. actionPerformed() của lớp CostumeStatFrm được gọi
22. actionPerformed() gọi hàm getCostumeStat() của lớp CostumeStatDAO
23. getCostumeStat() thực thi
24. getCostumeStat() gọi lớp CostumeStat để đóng gói kết quả
25. Lớp CostumeStat đóng gói các thuộc tính
26. Lớp CostumeStat gọi hàm setter từ lớp Costume để gói các thuộc tính kế thừa từ lớp Costume
27. Lớp Costume đóng gói các thuộc tính
28. Lớp Costume trả về gói thuộc tính cho lớp CostumeStat

29. Lớp CostumeStat trả gói kết quả về cho hàm getCostumeStat()
 30. getCostumeStat() trả kết quả về cho actionPerformed()
 31. actionPerformed() hiển thị kết quả trên giao diện
- CostumeStatFrm
32. Quản lý chọn 1 hàng để xem chi tiết về 1 trang phục
 33. actionPerformed() của lớp CostumeStatFrm được gọi
 34. actionPerformed() gọi lớp DetailCostumeStatFrm
 35. DetailCostumeStatFrm() được gọi
 36. DetailCostumeStatFrm() gọi hàm getBillofCostume() của lớp BillDAO
37. getBillofCostume() thực thi
 38. getBillofCostume() gọi lớp Bill để đóng gói kết quả
 39. Lớp Bill đóng gói các thuộc tính
 40. Lớp Bill gọi lớp Client để lấy các thuộc tính
 41. Lớp Client đóng gói các thuộc tính
 42. Lớp Client trả kết quả về lớp Bill
 43. Lớp Bill gọi lớp BorrowTicket để lấy các thuộc tính
 44. Lớp BorrowTicket đóng gói kết quả
 45. Lớp BorrowTicket gọi tới lớp BorrowCostume để lấy các thuộc tính
 46. Lớp BorrowCostume đóng gói các thuộc tính
 47. Lớp BorrowCostume trả kết quả về cho lớp BorrowTicket
 48. Lớp BorrowTicket trả kết quả về lớp Bill
 49. Lớp Bill trả kết quả về hàm getBillofCostume()
 50. getBillofCostume() trả kết quả về DetailCostumeStatFrm()
 51. DetailCostumeStatFrm() hiển thị chi tiết hóa đơn về trang phục
đã được chọn ở giao diện DetailCostumeStatFrm



10. Test plan và test case chuẩn cho test hộp đen

TT	Module	Test case
1	Thống kê trang phục được mượn nhiều	Có trang phục được mượn trong: start date < borrow date < return date < end date
2		Có trang phục được mượn trong: borrow date < start date < return date
3		Có trang phục được mượn trong: borrow date

		< end date < return date
4		Không có trang phục nào trong khoảng thời gian đó

Test case no 1:

CSDL trước khi test:

tblUser:

id	username	password	fullname	role
1	manager	manager	manager	manager
2	receptionist	receptionist	receptionist	receptionist

tblClient:

id	name	tel	mail
1	Nguyễn A	123456	a@mail.com
2	Vũ B	122333	b@gmail.com
3	Phan C	293682	c@gmail.com
4	D	723462	d@gmail.com
5	E	974735	e@gmail.com

tblCostume:

id	User Id	name	type	size	des	price	quantity	borrowCount
1	2	Áo dài	Truyền thống	s	Màu đỏ	100	5	4

2	2	Sườn xám	Truyền thống	M	Màu xanh	100	5	3
3	2	Kimon o	Truyền thống	L	Màu tím	100	5	1

tblBorrowTicket:

id	UserId	ClientId	borrowDate	depositAmount
1	2	1	4/1/2025	200
2	2	2	9/1/2025	100
3	2	3	3/3/2025	200
4	2	4	15/1/2025	200
5	2	5	21/1/2025	100

tblBorrowCostume:

id	BorrowTicketId	CostumeId	isReturned	quantity	selloff
1	1	1	1	2	
2	2	1	1	1	
3	3	1	1	1	
4	3	2	1	1	
5	4	2	1	2	
6	5	3	1	1	

tblBill:

id	UserId	ClientId	BorrowTicketId	returnDate	depositReturn
1	2	1	1	5/1/2025	180
2	2	2	2	10/1/2025	90
3	2	3	3	4/3/2025	180
4	2	4	4	15/1/2025	180
5	2	5	5	22/1/2025	90

tblReturnCostume:

id	BillId	quantity	BorrowCostumeId	rentalFee
1	1	2	1	20
2	2	1	2	10
3	3	1	3	10
4	3	1	4	10
5	4	2	5	20
6	5	1	6	10

tblDamageIssue:

id	BorrowCostumeId	ReturnCostumeId	description	penaltyAmount

tblDamageCostume

id	note	actualPenalty	returnCostumeid

--	--	--	--

Testing scenario và kết quả mong đợi

scenario	Expected result																		
1. Mở ứng dụng	Giao diện đăng nhập hiện lên với ô nhập username, password và nút đăng nhập																		
2. Nhập username = manager, password = manager, rồi click nút login	Giao diện quản lý hiện lên với nút xem thống kê trang phục được mượn nhiều																		
3. Click nút xem thống kê trang phục được mượn nhiều	Giao diện hiện lên ô nhập start date, end date, nút xem																		
4. Quản lý nhập start date = 1/1/2025, end date = 2/2/2025, click xem	<p>HT hiển thị danh sách các trang phục được mượn nhiều:</p> <table border="1"> <thead> <tr> <th>id</th> <th>name</th> <th>des</th> <th>size</th> <th>Borrow count</th> <th>Total amount</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Áo dài</td> <td>Màu đỏ</td> <td>s</td> <td>3</td> <td>30</td> </tr> <tr> <td>2</td> <td>Sườn xám</td> <td>xanh</td> <td>m</td> <td>2</td> <td>20</td> </tr> </tbody> </table>	id	name	des	size	Borrow count	Total amount	1	Áo dài	Màu đỏ	s	3	30	2	Sườn xám	xanh	m	2	20
id	name	des	size	Borrow count	Total amount														
1	Áo dài	Màu đỏ	s	3	30														
2	Sườn xám	xanh	m	2	20														

	3	kimono	tím	1	1	10															
	HT hiện lên chi tiết các hóa đơn có trang phục đó và nút return ở dưới																				
5. QL click dòng thứ 1 để xem chi tiết các hóa đơn có sườn xám	<table border="1"> <thead> <tr> <th>id</th><th>client</th><th>Borrow date</th><th>Return date</th><th>Total amount</th></tr> </thead> <tbody> <tr> <td>1</td><td>Nguyễn A</td><td>4/1/2025</td><td>5/1/2025</td><td>20</td></tr> <tr> <td>2</td><td>Vũ B</td><td>9/1/2025</td><td>10/1/2025</td><td>10</td></tr> </tbody> </table>						id	client	Borrow date	Return date	Total amount	1	Nguyễn A	4/1/2025	5/1/2025	20	2	Vũ B	9/1/2025	10/1/2025	10
id	client	Borrow date	Return date	Total amount																	
1	Nguyễn A	4/1/2025	5/1/2025	20																	
2	Vũ B	9/1/2025	10/1/2025	10																	
6. QL click return để về trang chủ	HT quay về trang chủ với nút xem thống kê																				

CSDL sau khi test: không thay đổi