

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TPHCM
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN CỬ NHÂN
ĐỀ TÀI: TRIỂN KHAI TÍNH NĂNG BẢO MẬT
TRONG XÂY DỰNG ỨNG DỤNG SỬA CHỮA VÀ BẢO
HÀNH ĐIỆN THOẠI

Giảng viên hướng dẫn: Nguyễn Phương Hạc

Sinh viên thực hiện:

1. 2033221472 – Mai Ngọc Hoàn
2. 2033220622 – Nguyễn Trần Dinh
3. 2033220450 – Trần Quốc Cường

TP HỒ CHÍ MINH, tháng 12 năm 2025

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TPHCM
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN CỬ NHÂN
ĐỀ TÀI: TRIỂN KHAI TÍNH NĂNG BẢO MẬT
TRONG XÂY DỰNG ỨNG DỤNG SỬA CHỮA VÀ BẢO
HÀNH ĐIỆN THOẠI

Giảng viên hướng dẫn: Nguyễn Phương Hạc

Sinh viên thực hiện:

1. 2033221472 – Mai Ngọc Hoàn
2. 2033220622 – Nguyễn Trần Dinh
3. 2033220450 – Trần Quốc Cường

TP HỒ CHÍ MINH, tháng 9 năm 2025

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến cô Nguyễn Phương Hạc, người đã trực tiếp hướng dẫn, chỉ bảo và định hướng cho nhóm trong suốt quá trình thực hiện đề tài “Triển khai tính năng bảo mật trong xây dựng ứng dụng sửa chữa và bảo hành điện thoại”. Sự tận tình, trách nhiệm và những góp ý quý báu của thầy/cô đã giúp nhóm chúng em hoàn thiện đề tài cả về nội dung lẫn phương pháp nghiên cứu.

Mặc dù nhóm đã nỗ lực hết mình, nhưng chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự đóng góp ý kiến từ cô để đề tài được hoàn thiện hơn.

Một lần nữa, chúng em xin chân thành cảm ơn!

TP HCM, ngày 16 tháng 9 năm 2025

Ký tên

Nguyễn Trần Dinh

Trần Quốc Cường

Mai Ngọc Hoàn

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Đồng ý cho bảo vệ.....

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

—yhan

Nguyễn Phương Hạc

MỤC LỤC

LỜI CẢM ƠN.....	i
MỤC LỤC.....	iii
DANH MỤC CÁC KÍ HIỆU VÀ CHỮ VIẾT TẮT	viii
DANH MỤC HÌNH ẢNH.....	x
DANH MỤC BẢNG	xix
PHÂN CÔNG CÔNG VIỆC.....	xxi
LỜI MỞ ĐẦU.....	1
CHƯƠNG 1: TỔNG QUAN.....	2
1.1. Lý do chọn đề tài.....	2
1.2. Mục tiêu nghiên cứu.....	2
1.2.1. Mục tiêu tổng quát.....	2
1.2.2. Mục tiêu cụ thể.....	2
1.3. Đối tượng và phạm vi nghiên cứu.....	3
1.4. Khảo sát cơ cấu tổ chức	3
1.5. Quy trình nghiệp vụ chi tiết.....	4
1.5.1. Quy trình tiếp nhận và tìm hiểu thông tin khách hàng	4
1.5.2. Quy trình kiểm tra tình trạng máy	5
1.5.3. Quy trình thông báo lỗi và báo giá	5
1.5.4. Quy trình kiểm tra kho và chuẩn bị linh kiện.....	6
1.5.5. Quy trình sửa chữa	6
1.5.6. Quy trình bàn giao và thanh toán.....	6
1.5.7. Quy trình quản lý	7
1.6. Các biểu mẫu thu thập trong quá trình khảo sát.....	8
CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG.....	11
2.1. Giới thiệu	11
2.2. Mô hình hóa quy trình nghiệp vụ	11

2.2.1. Sơ đồ usecase nghiệp vụ	11
2.2.2. Mô hình hóa quy trình nghiệp vụ	12
2.3. Mô hình hóa chức năng hệ thống.....	17
2.3.1. Sơ đồ usecase hệ thống	17
2.3.2. Đặc tả usecase hệ thống	17
2.4. Sơ đồ lớp ở mức phân tích	24
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG.....	26
3.1. Giới thiệu	26
3.2. Thiết kế cơ sở dữ liệu	26
3.3. Sơ đồ lớp ở mức thiết kế	36
3.4. Thiết kế chức năng hệ thống.....	36
3.4.1. Chức năng quản lý đơn hàng	36
3.4.2. Chức năng quản lý hóa đơn	38
3.4.3. Chức năng quản lý thông tin khách hàng	39
3.4.4. Chức năng sửa chữa và bảo hành	40
3.4.5. Chức năng quản lý linh kiện.....	41
3.4.6. Chức năng quản lý yêu cầu linh kiện	42
3.4.7. Chức năng sao lưu và phục hồi	44
3.4.8. Giao diện website.....	45
3.4.9. Giao diện mobile.....	67
3.5. Thiết kế API	81
CHƯƠNG 4: CÀI ĐẶT HỆ THỐNG	87
4.1. Thông số cài đặt môi trường.....	87
4.1.1. Môi trường Máy chủ (Server)	87
4.1.2. Cơ sở dữ liệu Oracle 21c.....	88
4.2. Triển khai máy chủ trên nền tảng ảo hóa.....	89
4.3. Triển khai cơ sở dữ liệu Oracle 21c.....	96

4.4. Triển khai VPN ZeroTier	103
4.5. Định danh, xác thực và profile.....	104
4.5.1. Định danh và xác thực	104
4.5.2. Profile	109
4.6. Xác thực đa nền tảng bằng mã QR.....	115
4.6.1. Tổng quan.....	115
4.6.2. Luồng hoạt động	116
4.6.3. Tóm tắt vai trò.....	118
4.7. Đảm bảo chỉ 1 phiên làm việc tồn tại trên cùng 1 nền tảng.....	120
4.8. Đăng ký	122
4.9. Đổi mật khẩu.....	124
4.10. Ký số và xác thực hóa đơn	126
4.10.1. Mục tiêu	126
4.10.2. Mô hình và kiến trúc hệ thống.....	126
4.10.3. Luồng hoạt động.....	127
4.10.4. Xác thực chữ ký số	130
4.11. Tích hợp mã QR trong tra cứu linh kiện.....	133
4.12. Quản lý vai trò (Role).....	135
4.12.1. Mục tiêu	135
4.13. Phân quyền, điều khiển truy cập (VPD).....	137
4.13.1. Mục tiêu	137
4.13.2. Mô tả bài toán	138
4.13.3. Các chức năng chính.....	138
4.13.4. Luồng hoạt động.....	139
4.13.5. Phạm vi áp dụng.....	139
4.13.6. Cài đặt và áp dụng VPD	140
4.14. Mã hóa dữ liệu	141

4.14.1. Mã hóa mật khẩu người dùng cơ sở dữ liệu.....	141
4.14.2. Mã hóa dữ liệu đường truyền.....	142
4.15. Kiểm toán và giải trình (Standard Audit, Trigger).....	145
4.15.1. Cơ sở lý thuyết.....	145
4.15.2. Triển khai Trigger trong Cơ sở dữ liệu	146
4.15.3. Triển khai Standard Audit trong Cơ sở dữ liệu.....	147
4.16. Sao lưu và phục hồi cơ sở dữ liệu.....	149
4.16.1. Mục tiêu	149
4.16.2. Chiến lược sao lưu tự động (Backup Strategy).....	149
CHƯƠNG 5: THỬ NGHIỆM VÀ TRIỂN KHAI.....	152
5.1. Kịch bản kiểm thử.....	152
5.1.1. Thủ nghiệm kết nối.....	152
5.1.2. Đăng nhập đa nền tảng.....	153
5.1.3. Xác thực đa nền tảng bằng mã QR	155
5.1.4. Đăng nhập tranh chấp trên web.....	160
5.1.5. Chính sách profile khóa tài khoản	161
5.1.6. Đăng xuất.....	163
5.1.7. Đăng ký.....	164
5.1.8. Đổi mật khẩu	167
5.1.9. Nhập kho.....	168
5.1.10. Xuất hóa đơn, xuất kho	170
5.1.11. Quét linh kiện bằng mã QR.....	173
5.1.12. Phân quyền và xác thực theo VPD.....	174
5.1.13. Kịch bản kiểm thử Audit	175
5.1.14. Phục hồi dữ liệu theo thời điểm.....	178
5.2. Đóng gói ứng dụng.....	180
KẾT LUẬN	182

TÀI LIỆU THAM KHẢO.....	184
-------------------------	-----

DANH MỤC CÁC KÍ HIỆU VÀ CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
DB	Database	Cơ sở dữ liệu
DBMS	Database Management System	Hệ quản trị cơ sở dữ liệu
ERD	Entity Relationship Diagram	Sơ đồ thực thể - mối quan hệ
GUI	Graphical User Interface	Giao diện người dùng đồ họa
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
CRUD	Create, Read, Update, Delete	Tạo, Đọc, Cập nhật, Xóa
ID	Identifier	Mã định danh
API	Application Programming Interface	Giao diện lập trình ứng dụng
RBAC	Role-Based Access Control	Kiểm soát truy cập dựa trên vai trò
JWT	JSON Web Token	Mã thông báo web dạng JSON
SHA	Secure Hash Algorithm	Thuật toán băm an toàn
PKCS	Public Key Cryptography Standards	Các tiêu chuẩn mật mã khóa công khai
VPD	Virtual Private Database	Cơ sở dữ liệu riêng ảo
RSA	Rivest–Shamir–Adleman	Thuật toán mã hóa bất đối xứng
AES	Advanced Encryption Standard	Tiêu chuẩn mã hóa tiên tiến
CDB	Container Database	Cơ sở dữ liệu vùng chứa
PDB	Pluggable Database	Cơ sở dữ liệu có thể cắm
VPN	Virtual Private Network	Mạng riêng ảo
NAT	Network Address Translation	Biên dịch địa chỉ mạng

Viết tắt	Tiếng Anh	Tiếng Việt
LVM	Logical Volume Manager	Quản lý phân vùng logic
QR	Quick Response	Mã phản hồi nhanh
PDF	Portable Document Format	Định dạng tài liệu di động
MAC	Mandatory Access Control	Kiểm soát truy cập bắt buộc
DAC	Discretionary Access Control	Kiểm soát truy cập tự quyết
OLS	Oracle Label Security	Bảo mật nhãn Oracle

DANH MỤC HÌNH ẢNH

Hình 1.1: Phiếu bảo hành.....	8
Hình 1.2: Phiếu sửa chữa.....	9
Hình 1.3: Phiếu hóa đơn	10
Hình 2.1: Sơ đồ usecase nghiệp vụ	12
Hình 2.2: Sơ đồ nghiệp vụ thanh toán.....	13
Hình 2.3: Sơ đồ nghiệp vụ bảo hành.	14
Hình 2.4: Sơ đồ nghiệp vụ thanh toán.....	15
Hình 2.5: Sơ đồ nghiệp vụ mua linh kiện.	16
Hình 2.6: Sơ đồ usecase hệ thống.....	17
Hình 2.7: Sơ đồ lớp ở mức phân tích.	25
Hình 3.1: Mô hình dữ liệu.	26
Hình 3.2: Sơ đồ lớp ở mức thiết kế.....	36
Hình 3.3: Sơ đồ lớp ở mức thiết kế của quản lý đơn hàng.	37
Hình 3.4: Sơ đồ lớp ở mức thiết kế của quản lý hóa đơn.....	38
Hình 3.5: Sơ đồ lớp ở mức thiết kế của quản lý thông tin khách hàng.	39
Hình 3.6: Sơ đồ lớp ở mức thiết kế của sửa chữa và bảo hành.	40
Hình 3.7: Sơ đồ lớp ở mức thiết kế của quản lý linh kiện.....	41
Hình 3.8: Sơ đồ lớp ở mức thiết kế của quản lý yêu cầu linh kiện.....	42
Hình 3.9: Sơ đồ lớp ở mức thiết kế của quản lý tài khoản người dùng.....	43
Hình 3.10: Sơ đồ lớp ở mức thiết kế của sao lưu và phục hồi.	44
Hình 3.11: Giao diện chức năng đăng nhập Nhân viên.....	45
Hình 3.12: Giao diện chức năng tải Private key nhân viên.	46
Hình 3.13: Giao diện chức năng xem thông tin của tài khoản nhân viên... ..	46
Hình 3.14: Giao diện chức năng đổi mật khẩu cho nhân viên.	47
Hình 3.15: Giao diện chức năng đăng nhập bằng QR cho nhân viên.	48
Hình 3.16: Giao diện chức năng đăng nhập bằng QR cho nhân viên.	49

Hình 3.17: Giao diện chức năng thêm nhân viên mới.....	50
Hình 3.18: Giao diện chức năng quản lý phân quyền của Admin.....	50
Hình 3.19: Giao diện chức năng quản lý profile.....	51
Hình 3.20: Giao diện danh sách khách hàng.....	52
Hình 3.21: Giao diện danh sách đặt lịch hẹn.....	52
Hình 3.22: Giao diện danh sách nhập kho.....	53
Hình 3.23: Giao diện chức năng nhập kho linh kiện.....	53
Hình 3.24: Giao diện chức năng xuất kho.....	54
Hình 3.25: Giao diện chức năng xem chi tiết đơn xuất kho.....	55
Hình 3.26: Giao diện chức năng quản lý hóa đơn.....	56
Hình 3.27: Giao diện chức năng chi tiết hóa đơn.....	56
Hình 3.28: Giao diện danh sách linh kiện.....	57
Hình 3.29: Giao diện danh sách yêu cầu linh kiện.....	58
Hình 3.30: Giao diện danh sách đơn hàng.....	58
Hình 3.31: Giao diện chức năng tạo đơn hàng.....	59
Hình 3.32: Giao diện chi tiết đơn hàng.....	60
Hình 3.33: Giao diện Trigger Audit.....	60
Hình 3.34: Giao diện Standard Audit.....	61
Hình 3.35: Giao diện trang chủ khách hàng.....	61
Hình 3.36: Giao diện chức năng đăng nhập bằng QR.....	62
Hình 3.37: Giao diện chức năng đặt lịch cho khách hàng.....	63
Hình 3.38: Giao diện chức năng xem lịch hẹn của khách hàng.....	63
Hình 3.39: Giao diện chức năng xem đơn hàng cho khách hàng.....	64
Hình 3.40: Giao diện chức năng liên hệ cho khách hàng.....	65
Hình 3.41: Giao diện chức năng đổi mật khẩu cho khách hàng.....	65
Hình 3.42: Giao diện chức năng trợ giúp và FAQ.....	66
Hình 3.43: Giao diện chức năng xác thực hóa đơn chứa chữ ký số.....	66

Hình 3.44: Giao diện quét QR đăng nhập web.....	67
Hình 3.45: Giao diện nhân viên.....	68
Hình 3.46: Giao diện lịch hẹn	69
Hình 3.47: Giao diện quản lí đơn hàng.....	70
Hình 3.48: Giao diện chi tiết đơn hàng.....	71
Hình 3.49: Giao diện Quản lí đơn nhập.....	72
Hình 3.50: Giao diện chi tiết đơn nhập.....	73
Hình 3.51: Giao diện xác thực chữ ký số bên cơ sở dữ liệu.....	74
Hình 3.52: Giao diện Quản lý hóa đơn xuất.....	75
Hình 3.53: Giao diện Quản lý hóa đơn.....	76
Hình 3.54: Giao diện đổi mật khẩu.....	77
Hình 3.55: Giao diện trang chủ khách hàng.....	78
Hình 3.56: Giao diện đơn hàng.....	79
Hình 3.57: Giao diện thông tin khách hàng.....	80
Hình 4.1: Tạo máy ảo.....	89
Hình 4.2: Tạo nhanh bằng cách chọn Typical	90
Hình 4.3: Dung lượng máy ảo tối thiểu 50GB.....	91
Hình 4.4: Thông số chi tiết của máy chủ	91
Hình 4.5: Customize để cấu hình chi tiết cho máy	92
Hình 4.6: Cấu hình processors core và memory.....	92
Hình 4.7: Màn hình boot Oracle Linux.....	93
Hình 4.8: Thiết lập cài đặt cuối cùng.	93
Hình 4.9: Phân vùng bộ nhớ.....	94
Hình 4.10: Hoàn tất phân vùng bộ nhớ.....	94
Hình 4.11: Cài đặt mật khẩu root.....	95
Hình 4.12: Hoàn tất cài đặt mật khẩu cho root.	95
Hình 4.13: Đòng ý điều khoản sử dụng để tiến hành khởi động lại.	96

Hình 4.14: Lệnh cập nhập hệ thống	96
Hình 4.15: Cài gói meta-package của Oracle	97
Hình 4.16: Cài đặt preinstall tạo sẵn môi trường hệ thống.....	97
Hình 4.17: Trang tải gói Oracle database 21c.....	98
Hình 4.18: Cài đặt Oracle database 21c.....	98
Hình 4.19: Cài đặt thành công Oracle Database 21c	99
Hình 4.20: Lệnh để tạo sẵn thiết lập cơ bản.....	99
Hình 4.21: Mở cổng 1521 trên tường lửa.	100
Hình 4.22: Khởi động listener.....	101
Hình 4.23: Hiển thị tên Container hiện tại đang kết nối.	101
Hình 4.24: tra trạng thái hiện tại của Oracle instance.....	101
Hình 4.25: Chuyển phiên làm việc (session).....	102
Hình 4.26: Tạo người dùng admin.....	102
Hình 4.27: Cấp quyền cơ bản cho người dùng.	102
Hình 4.28: Kết nối thành công.	103
Hình 4.29: Mã nguồn đăng nhập và định danh người dùng	104
Hình 4.30: Mã nguồn xác thực gán ở header ở client trước khi gọi API... <td>105</td>	105
Hình 4.31: Mã nguồn từ điển lưu trữ thông tin người dùng.	105
Hình 4.32: Mã nguồn tạo JWT Token cho client	106
Hình 4.33: Mã nguồn Xác thực người dùng mới khi gọi API.....	107
Hình 4.34: Mã nguồn đăng xuất	108
Hình 4.35: Sơ đồ quy trình đăng nhập, gọi API, đăng xuất.....	109
Hình 4.36: Chi tiết thông số của profile.....	110
Hình 4.37: Tạo profile mới	111
Hình 4.38: Thao tác bằng lệnh SQL của tạo profile.	112
Hình 4.39: Cập nhật profile	112
Hình 4.40: Thao tác bằng lệnh SQL của cập nhật profile.	113

Hình 4.41: Xóa profile.....	114
Hình 4.42: Thao tác bằng lệnh SQL của xóa profile.....	114
Hình 4.43: Gán profile cho người dùng.....	115
Hình 4.44: Thao tác bằng lệnh SQL của gán profile.....	115
Hình 4.45: API cho chức năng đăng nhập QR.....	116
Hình 4.46: Mã nguồn xác thực đa nền tảng đăng nhập web từ mobile.....	117
Hình 4.47: Mã nguồn xác thực đa nền tảng web từ mobile.....	118
Hình 4.48: Sơ đồ xác thực Web từ Mobile bằng mã QR.....	119
Hình 4.49: Sơ đồ xác thực Mobile từ Web bằng chức năng QR Login.....	120
Hình 4.50: Mã nguồn dọn dẹp phiên làm việc cũ.....	121
Hình 4.51: Mã nguồn xóa phiên làm việc.....	121
Hình 4.52: Mã nguồn định danh kết nối mới.....	121
Hình 4.53. Mã nguồn đăng ký nhân viên.....	122
Hình 4.54: Mã nguồn lấy mật khẩu hiện tại và kiểm tra mật khẩu cũ.....	124
Hình 4.55: Mã nguồn băm mật khẩu và cập nhập bảng.....	124
Hình 4.56: Mã nguồn tạo và ký PDF.....	128
Hình 4.57: Hóa đơn bán hàng ở định dạng PDF.....	128
Hình 4.58: Hóa đơn nhập kho ở định dạng PDF.....	129
Hình 4.59: Hóa đơn xuất kho ở định dạng PDF.....	129
Hình 4.60: Mã nguồn xác thực chữ ký số.....	130
Hình 4.61: Kết quả khi xác thực chữ ký hợp lệ.....	131
Hình 4.62: Kết quả với PDF trong bản lưu hệ thống.....	132
Hình 4.63: Kết quả không trùng với PDF trong bản lưu hệ thống.....	132
Hình 4.64: Mã nguồn lấy chi tiết linh kiện.....	133
Hình 4.65: Mã nguồn tạo ảnh QR.....	133
Hình 4.66: Vai trò (Role) của người dùng.....	135
Hình 4.67: Gán vai trò.....	136

Hình 4.68: Thao tác bằng lệnh SQL của gán vai trò.....	136
Hình 4.69: Thu hồi vai trò.....	137
Hình 4.70: Thao tác bằng lệnh SQL của thu hồi vai trò.....	137
Hình 4.71: Sơ đồ luồng phân quyền, truy cập.	138
Hình 4.72: Mã nguồn hàm chính sách kiểm soát truy cập đơn hàng.	141
Hình 4.73: Cấu hình chính sách VPD (Virtual Private Database) cho bảng ORDERS.....	141
Hình 4.74: Mã nguồn mã hóa RSA/AES	144
Hình 4.75: Sơ đồ quy trình mã hóa giải mã dữ liệu đường truyền.	145
Hình 4.734.76: Cấu hình tham số kết nối trong script sao lưu tự động....	150
Hình 5.1: Ip của máy client.....	152
Hình 5.2: Ip ZeroTier của Server Linux.	152
Hình 5.3: Kết quả kiểm tra kết nối mạng VPN thành công.	153
Hình 5.4: Kết nối thành công qua IP ZeroTier.....	153
Hình 5.5: Thông tin phiên làm việc trên web.	154
Hình 5.6: Đăng nhập cùng tài khoản trên mobile.	154
Hình 5.7: Kết quả thực tế khi đăng nhập 2 nền tảng	155
Hình 5.8: Giao diện trang chủ của nhân viên trên mobile.....	155
Hình 5.9: Mở camera để bắt đầu quét QR trên web.....	156
Hình 5.10: Mã QR trên web.....	156
Hình 5.11: Thông tin phiên làm việc trên nền tảng web.....	157
Hình 5.12: Kiểm tra phiên làm việc ở cơ sở dữ liệu.....	157
Hình 5.13: Mã QR để đăng nhập vào ứng dụng mobile	158
Hình 5.14: Chọn đăng nhập bằng tài khoản nhân viên.....	158
Hình 5.15: Mở camera quét mã QR để đăng nhập.....	159
Hình 5.16: Đăng nhập thành công	159
Hình 5.17: Thông tin phiên làm việc đầu tiên.	160

Hình 5.18: Thông tin phiên làm việc thứ hai.....	160
Hình 5.19: Session trước khi thực hiện.	161
Hình 5.20: Kết quả thực tế Session cũ đã bị thay thế.	161
Hình 5.21: Kiểm tra trạng thái người dùng TIEPTAN.....	162
Hình 5.22: Đăng nhập người dùng TIEPTAN với mật khẩu sai.	162
Hình 5.23: Thông báo sau khi nhập mật khẩu sai.	162
Hình 5.24: Trạng thái của người dùng TIEPTAN sau khi nhập sai mật khẩu.	
.....	163
Hình 5.25: SQL kiểm tra phiên làm việc dựa theo p_client_identifier....	163
Hình 5.26: Phiên làm việc hiện tại của người dùng trên nền tảng web....	164
Hình 5.27: Phiên làm việc trên web đã bị xóa.....	164
Hình 5.28: SQL kiểm tra người dùng có trên cơ sở dữ liệu hay không....	164
Hình 5.29: Thực hiện đăng ký khách hàng.	165
Hình 5.30: Dữ liệu khách hàng cập nhật lên cơ sở dữ liệu.....	165
Hình 5.31: Kiểm tra sự tồn tại của username Test.....	166
Hình 5.32: Đăng ký nhân viên với username Test.	166
Hình 5.33: Private key tương ứng cho nhân viên.....	167
Hình 5.34: Dữ liệu nhân viên cập nhật trên cơ sở dữ liệu.	167
Hình 5.35: Mã băm mật khẩu lưu trên cơ sở dữ liệu.	167
Hình 5.36: Đổi mật khẩu của người dùng.....	168
Hình 5.37: Mã hash ban đầu đã bị thay đổi.....	168
Hình 5.38: Thực hiện nhập kho vật phẩm mẫu.....	169
Hình 5.39: Mã nguồn tạo và ký số PDF.....	169
Hình 5.40: Thông tin chi tiết của đơn nhập kho trên cơ sở dữ liệu.	170
Hình 5.41: PDF đã được ký số.	170
Hình 5.42: Tạo đơn hàng mẫu.....	170
Hình 5.43: Tạo yêu cầu linh kiện mẫu.....	171

Hình 5.44: Hệ thống hiển thị yêu cầu.....	171
Hình 5.45: Thông tin chi tiết cho đơn xuất kho trên hệ thống.....	171
Hình 5.46: PDF hóa đơn xuất kho.	172
Hình 5.47: PDF hóa đơn bán hàng.....	172
Hình 5.48: Mở Camera quét mã QR của linh kiện.....	173
Hình 5.49: Thông tin linh kiện được hiển thị sau khi quét.....	173
Hình 5.50: Kết quả truy vấn bảng ORDERS dưới quyền Admin.....	174
Hình 5.51: Kết quả truy vấn bảng ORDERS dưới quyền Khách hàng.	175
Hình 5.52: Thiết lập thao tác kiểm thử Standard Audit.	176
Hình 5.53: Kết quả truy vấn DBA_AUDIT_TRAIL.....	176
Hình 5.54: Thiết lập thao tác kiểm thử Trigger	177
Hình 5.55: Kết quả truy vấn log	177
Hình 5.56: Giá trị cũ của dữ liệu	178
Hình 5.57: Giá trị mới sau khi chỉnh sửa.....	178
Hình 5.58: Trạng thái dữ liệu trước khi sao lưu.....	178
Hình 5.59: Nhật ký thực thi quá trình sao lưu thành công.	178
Hình 5.60: Thực hiện thay đổi dữ liệu sau thời điểm sao lưu.....	179
Hình 5.61: Dữ liệu sau khi thay đổi (Có thêm dòng ID 3).....	179
Hình 5.62: Nhật ký thực thi quá trình phục hồi dữ liệu.	179
Hình 5.63: Dữ liệu sau khi restore.	180
Hình 5.64: Thư mục mã nguồn.....	180
Hình 5.65: Đóng gói WebApp.....	180
Hình 5.66: WebApp sau khi đóng gói.	180
Hình 5.67: Đóng gói WebAPI.	181
Hình 5.68: WebAPI sau khi đóng gói.....	181
Hình 5.69: Đóng gói apk	181

DANH MỤC BẢNG

Bảng 2.1: Bảng đặc tả usecase tra cứu đơn hàng.....	17
Bảng 2.2: Bảng đặc tả usecase tra cứu hóa đơn.....	18
Bảng 2.3: Bảng đặc tả usecase quản lý thông tin khách hàng.....	18
Bảng 2.4: Bảng đặc tả usecase sửa chữa và bảo hành.	20
Bảng 2.5: Bảng đặc tả usecase quản lý linh kiện.	20
Bảng 2.6: Bảng đặc tả usecase quản lý yêu cầu linh kiện.	21
Bảng 2.7: Bảng đặc tả usecase quản lý tài khoản người dùng.....	22
Bảng 2.8: Bảng đặc tả usecase sao lưu và phục hồi dữ liệu.	23
Bảng 3.1: Bảng nhân viên.....	27
Bảng 3.2: Bảng khách hàng.	27
Bảng 3.3: Bảng đặt hàng.....	28
Bảng 3.4: Bảng nhập kho.....	29
Bảng 3.5: Bảng chi tiết nhập kho.....	29
Bảng 3.6: Bảng linh kiện.	30
Bảng 3.7: Bảng xuất kho.	30
Bảng 3.8: Bảng chi tiết xuất kho.....	31
Bảng 3.9: Bảng yêu cầu linh kiện.	31
Bảng 3.10: Bảng chi tiết yêu cầu linh kiện.	32
Bảng 3.11: Bảng log OTP của người dùng.....	32
Bảng 3.12: Bảng ca trực.	33
Bảng 3.13: Bảng đặt lịch hẹn.....	33
Bảng 3.14: Bảng hóa đơn	34
Bảng 3.15: Bảng chi tiết hóa đơn.....	34
Bảng 3.16: Bảng dịch vụ.	35
Bảng 3.17: Bảng dịch vụ của đơn hàng.	35
Bảng 3.18: Common API.....	81

Bảng 3.19: Public API.....	81
Bảng 3.20: Admin API	82
Bảng 4.1: Thông số kĩ thuật của máy chủ.....	87
Bảng 4.2: Cấu hình biến môi trường Oracle.....	88
Bảng .: Bảng tham số của profile.....	110
Bảng .: Bảng tóm tắt vai trò xác thực Web từ Mobile bằng mã QR.....	118
Bảng .: Bảng tóm tắt vai trò xác thực điện thoại từ web bằng QR.....	119
Bảng .: Bảng chi tiết tham số của vai trò	135
Bảng .: Bảng phạm vi áp dụng của VPD.	139
Bảng .: Tham số bảng thông tin kiểm toán.....	146

PHÂN CÔNG CÔNG VIỆC

MSSV	Họ và tên	Công việc được giao
2033221472	Mai Ngọc Hoàn	<p>Khảo sát nghiệp vụ, các tính năng bảo mật trong hệ thống</p> <p>Khảo sát hiện trạng: Cơ cấu tổ chức, Qui trình, biểu mẫu</p> <p>Lập sơ đồ Use-Case nghiệp vụ và đặc tả</p> <p>Lập sơ đồ Use-Case hệ thống và đặc tả.</p> <p>Lập Sơ đồ lớp phân tích</p> <p>Lập Sơ đồ lớp thiết kế, Mô hình dữ liệu</p> <p>Thiết kế giao diện ứng dụng Web</p> <p>Hỗ trợ thiết kế xây dựng các tính năng bảo mật :</p> <ul style="list-style-type: none"> + Xác thực chữ ký số + Mã hóa mật khẩu + Thiết kế profile + Standard Audit
2033220622	Nguyễn Trần Dinh	<p>Xây dựng API hệ thống</p> <p>Triển khai hệ thống trên Linux, cơ sở dữ liệu, kết nối từ xa</p> <p>Xây dựng BackEnd website</p> <p>Hỗ trợ xây dựng Backend Mobile</p> <p>Sơ đồ quy trình hoạt động</p> <p>Thiết kế xây dựng các tính năng bảo mật :</p> <ul style="list-style-type: none"> + Đăng nhập bằng quét QR + Định danh và xác thực + Ký số hóa đơn + Phân quyền VPD + Audit trigger, <p>Thiết kế profile</p>
2033220450	Trần Quốc Cường	<p>Thiết kế giao diện</p> <p>Triển khai hệ thống trên Linux, cơ sở dữ liệu, kết nối từ xa</p>

	<p>Triển khai hệ thống trên Linux, cơ sở dữ liệu, kết nối từ xa</p> <p>Hỗ trợ xây dựng BackEnd website</p> <p>Xây dựng Backend Mobile</p> <p>Xây dựng FrontEnd Mobile</p> <p>Hỗ trợ thiết kế giao diện</p> <p>Hỗ trợ thiết kế xây dựng các tính năng bảo mật :</p> <ul style="list-style-type: none"> + Đổi mật khẩu + Đăng ký người dùng + Truy vấn linh kiện bằng QR + Sao lưu phục hồi dữ liệu <p>Thiết kế profile</p>
--	---

LỜI MỞ ĐẦU

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, điện thoại thông minh đã trở thành một phần tất yếu trong đời sống con người. Kéo theo đó, nhu cầu sửa chữa và bảo hành điện thoại ngày càng tăng cao, tạo điều kiện cho nhiều cửa hàng và trung tâm dịch vụ ra đời. Tuy nhiên, trên thực tế phần lớn các đơn vị này vẫn đang quản lý thông tin khách hàng, thiết bị và quy trình bảo hành theo phương thức thủ công như ghi chép bằng giấy hoặc sử dụng Excel để thống kê. Cách quản lý này tuy đơn giản nhưng bộc lộ nhiều hạn chế, thông tin dễ thất lạc, thiếu tính minh bạch, khó truy xuất khi cần và chưa tạo được sự chặt chẽ trong toàn bộ quy trình.

Trong khi đó, dữ liệu liên quan đến khách hàng, thiết bị, lịch sử sửa chữa và bảo hành là những thông tin quan trọng, cần được quản lý một cách an toàn và chính xác. Việc thiếu đi một hệ thống quản lý chuyên nghiệp không chỉ gây khó khăn trong vận hành mà còn tiềm ẩn nguy cơ mất mát, rò rỉ hoặc bị sửa đổi trái phép thông tin. Điều này có thể ảnh hưởng trực tiếp đến quyền lợi của khách hàng cũng như uy tín và hiệu quả hoạt động của cửa hàng. Do đó, nhiều đơn vị, cửa hàng thực sự cần một ứng dụng quản lý sửa chữa và bảo hành điện thoại hiện đại, vừa hỗ trợ vận hành hiệu quả vừa đảm bảo yếu tố bảo mật dữ liệu.

Chính vì những lý do nêu trên, em đã quyết định chọn đề tài “Triển khai tính năng bảo mật trong xây dựng ứng dụng sửa chữa và bảo hành điện thoại” làm khóa luận tốt nghiệp. Đề tài vừa mang tính thực tiễn cao khi giải quyết được những khó khăn tồn tại ở nhiều cửa hàng hiện nay, vừa mang ý nghĩa khoa học khi kết hợp giữa lý thuyết và thực hành để tạo ra một giải pháp quản lý hiệu quả, minh bạch và an toàn hơn.

CHƯƠNG 1: TỔNG QUAN

1.1. Lý do chọn đề tài

Trong bối cảnh hiện nay, ngành công nghiệp điện thoại di động phát triển mạnh mẽ, nhu cầu sửa chữa và bảo hành điện thoại ngày càng tăng cao. Khách hàng không chỉ mong muốn thiết bị được sửa chữa nhanh chóng, chính xác mà còn yêu cầu thông tin cá nhân và dữ liệu liên quan phải được bảo mật tuyệt đối.

Tuy nhiên, thực tế cho thấy nhiều cơ sở sửa chữa và bảo hành điện thoại vẫn quản lý thủ công bằng giấy tờ hoặc các công cụ đơn giản như Excel. Theo khảo sát tại cửa hàng sửa chữa điện thoại 24hStore tại TP.HCM, mỗi ngày trung bình tiếp nhận từ 20–30 thiết bị, toàn bộ thông tin khách hàng và tình trạng máy đều được ghi bằng phiếu giấy. Việc lưu trữ thủ công này dẫn đến nhiều rủi ro như thất lạc phiếu, khó kiểm tra lịch sử sửa chữa khi khách hàng quay lại, hoặc mất nhiều thời gian tra cứu.

Ngoài ra, trong một số trường hợp, do không có cơ chế bảo mật và phân quyền rõ ràng, nhân viên có thể truy cập vào toàn bộ thông tin khách hàng, gây lo ngại về tính an toàn dữ liệu. Thực tế đã có những phản ánh trên báo chí về tình trạng rò rỉ thông tin khách hàng từ các cửa hàng sửa chữa điện thoại, làm giảm uy tín doanh nghiệp và ảnh hưởng đến quyền lợi người tiêu dùng.

Chính vì vậy, việc xây dựng một ứng dụng quản lý sửa chữa và bảo hành điện thoại tích hợp tính năng bảo mật là hết sức cần thiết. Hệ thống này không chỉ giúp quản lý minh bạch, hiệu quả mà còn nâng cao tính chuyên nghiệp và tạo sự tin tưởng cho khách hàng.

1.2. Mục tiêu nghiên cứu

1.2.1. Mục tiêu tổng quát

Xây dựng một ứng dụng hỗ trợ quản lý quy trình sửa chữa và bảo hành điện thoại có tích hợp các tính năng bảo mật, nhằm nâng cao hiệu quả quản lý và đảm bảo an toàn dữ liệu khách hàng.

1.2.2. Mục tiêu cụ thể

Khảo sát, phân tích và mô hình hóa quy trình nghiệp vụ sửa chữa – bảo hành điện thoại.

Áp dụng các kỹ thuật bảo mật trong Oracle như: mã hóa dữ liệu, profile, định danh và xác thực, phân quyền truy cập (MAC, DAC, VPD, OLS), kiểm toán và giải trình (Standard Audit, Trigger).

Xây dựng hệ thống có chức năng quản lý khách hàng, dịch vụ, đơn vị cung cấp, kèm theo các tiện ích hiện đại như hóa đơn ký số, quét mã QR để tra cứu dịch vụ.

Thiết kế và triển khai hệ thống trên nền tảng cơ sở dữ liệu Oracle, kết hợp ngôn ngữ lập trình như C# hoặc Java.

Kiểm thử, triển khai và đánh giá hiệu quả của ứng dụng.

1.3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Các quy trình nghiệp vụ liên quan đến hoạt động sửa chữa và bảo hành điện thoại.
- Các kỹ thuật và giải pháp bảo mật trên hệ quản trị cơ sở dữ liệu Oracle.
- Các công cụ phát triển ứng dụng trên nền web hoặc desktop hỗ trợ quản lý nghiệp vụ.

Phạm vi nghiên cứu:

- Tập trung xây dựng một trang web và ứng dụng nhằm Quản lý các nghiệp vụ cơ bản trong sửa chữa và bảo hành điện thoại. Ứng dụng chỉ triển khai các chức năng chính như Quản lý thông tin khách hàng, thiết bị, phiếu sửa chữa, phiếu bảo hành.
- Về mặt bảo mật, đề tài tập trung nghiên cứu và áp dụng một số giải pháp quan trọng trên hệ quản trị cơ sở dữ liệu Oracle, bao gồm cơ chế xác thực người dùng, phân quyền truy cập, mã hóa dữ liệu.

1.4. Khảo sát cơ cấu tổ chức

Một đơn vị sửa chữa và bảo hành điện thoại thông thường có cơ cấu tổ chức như sau:

Admin(Quản trị hệ thống):

- Quản lý tài khoản người dùng.
- Phân quyền sử dụng hệ thống.

- Giám sát toàn bộ hoạt động trong hệ thống (quản lý khách hàng, quản lý linh kiện, quản lý sửa chữa – bảo hành, quản lý ca trực).

Nhân viên tư vấn:

- Tiếp nhận khách hàng, lập hồ sơ khách hàng.
- Ghi nhận thông tin yêu cầu sửa chữa hoặc bảo hành.
- Tra cứu, thống kê, báo cáo thông tin liên quan đến khách hàng.

Kỹ thuật viên:

- Thực hiện quy trình sửa chữa thiết bị theo phiếu yêu cầu.
- Cập nhật tình trạng sửa chữa, bảo hành.
- Có thể lập yêu cầu linh kiện phục vụ sửa chữa.

Thủ kho:

- Quản lý linh kiện và vật tư.
- Thực hiện nhập kho, xuất kho, kiểm tra chất lượng – số lượng.
- Thống kê, báo cáo tồn kho.
- Phê duyệt phiếu xuất kho, tạo danh mục linh kiện mới.

Khách hàng:

- Cung cấp thông tin và thiết bị cần sửa chữa/bảo hành.
- Nhận thông báo và bàn giao thiết bị sau khi xử lý.
- Có thể tra cứu thông tin qua mã QR hoặc yêu cầu từ nhân viên.

1.5. Quy trình nghiệp vụ chi tiết

1.5.1. Quy trình tiếp nhận và tìm hiểu thông tin khách hàng

1. Khách hàng tiếp cận cửa hàng
 - Thông qua website, mạng xã hội, quảng cáo, hoặc giới thiệu từ khách hàng cũ.
 - Có thể liên hệ trước qua điện thoại hoặc chat trực tuyến để đặt lịch kiểm tra.
2. Tiếp nhận thông tin từ khách hàng
 - Khách hàng miêu tả tình trạng máy, các lỗi gặp phải hoặc yêu cầu sửa chữa/bảo trì.
 - Nhân viên tiếp tân ghi nhận thông tin chi tiết: loại thiết bị, thời gian mua, tình trạng bảo hành.

3. Tư vấn dịch vụ

- Tiếp tân giới thiệu các dịch vụ đang được cửa hàng cung cấp: sửa chữa phần cứng, thay linh kiện, nâng cấp phần mềm, vệ sinh thiết bị, bảo hành,..
- Hướng dẫn khách hàng về quy trình tiếp nhận và dự kiến thời gian xử lý.

1.5.2. Quy trình kiểm tra tình trạng máy

1. Kiểm tra thông tin bảo hành

- Tiếp tân tra cứu thông tin bảo hành của máy:
 - + Còn hạn bảo hành sẽ sửa chữa theo chính sách bảo hành.
 - + Hết hạn bảo hành sẽ áp dụng phí dịch vụ và phí linh kiện (nếu cần thay).

2. Chuyển máy cho kỹ thuật viên

- Tiếp tân lập phiếu tiếp nhận máy, ghi rõ lỗi, tình trạng bảo hành, thông tin khách hàng.
- Chuyển máy kèm phiếu tới kỹ thuật viên để kiểm tra chi tiết.

3. Kỹ thuật viên kiểm tra máy

- Kiểm tra chi tiết phần cứng (màn hình, pin, bo mạch, camera, cổng sạc...).
- Kiểm tra phần mềm, dữ liệu và các lỗi hệ thống.
- Xác định linh kiện cần thay hoặc các bước sửa chữa cần thực hiện.

1.5.3. Quy trình thông báo lỗi và báo giá

1. Thông báo tình trạng máy

- Kỹ thuật viên báo cáo lỗi và các linh kiện cần thay cho tiếp tân.
- Tiếp tân thông báo lại với khách hàng:
 - + Nếu thiết bị còn bảo hành sẽ sửa chữa miễn phí.
 - + Nếu thiết bị hết hạn bảo hành sẽ thông báo phí dịch vụ và phí linh kiện.

2. Xác nhận sửa chữa

- Tiếp tân ghi nhận sự đồng ý của khách hàng về giá và các linh kiện cần thay.
- Thông báo ngày dự kiến hoàn thành và nhận thiết bị.

1.5.4. Quy trình kiểm tra kho và chuẩn bị linh kiện

1. Kỹ thuật viên gửi yêu cầu linh kiện
 - Ghi rõ danh sách linh kiện cần thay theo phiếu sửa chữa.
2. Thủ kho kiểm tra tồn kho
 - Kiểm tra số lượng linh kiện trong kho.
 - Nếu đủ sẽ xuất linh kiện và cập nhật tồn kho.
 - Nếu thiếu sẽ thông báo kỹ thuật viên và tiếp tục thời gian nhập linh kiện.
3. Nhập linh kiện (nếu thiếu)
 - Thủ kho tiếp nhận linh kiện từ nhà cung cấp, kiểm tra số lượng, chất lượng, hóa đơn, hạn bảo hành (nếu có).
 - Cập nhật vào kho, sắp xếp hợp lý để thuận tiện cho việc xuất kho.
4. Xác nhận linh kiện sẵn sàng
 - Kỹ thuật viên nhận linh kiện từ thủ kho trước khi tiến hành sửa chữa.

1.5.5. Quy trình sửa chữa

1. Kiểm tra lại trước khi sửa
 - Kỹ thuật viên nhận máy từ tiếp tân.
 - Kiểm tra lại tình trạng máy và xác nhận linh kiện cần thay.
2. Tiến hành sửa chữa
 - Thay linh kiện, sửa lỗi phần cứng và phần mềm theo yêu cầu.
 - Kiểm tra kỹ lưỡng sau sửa chữa:
 - + Chức năng máy hoạt động bình thường.
 - + Không phát sinh lỗi mới.
 - + Kiểm tra pin, camera, kết nối, cảm ứng, loa, mic.....
3. Vệ sinh và đóng gói
 - Vệ sinh thiết bị sau khi sửa.
 - Đóng gói cẩn thận, chuẩn bị cho việc bàn giao.-

1.5.6. Quy trình bàn giao và thanh toán

1. Thông báo hoàn tất sửa chữa
 - Tiếp tân liên hệ khách hàng thông báo nhận máy.
 - Gửi phiếu sửa chữa, danh sách linh kiện thay thế và tổng chi phí.
2. Bàn giao thiết bị và thanh toán

- Khách hàng ký xác nhận hóa đơn.
 - Thanh toán chi phí dịch vụ và linh kiện (nếu hết bảo hành).
3. Chính sách bảo hành dịch vụ
- Thông báo thời gian bảo hành cho dịch vụ sửa chữa và linh kiện.
 - Lưu hồ sơ sửa chữa và thông tin khách hàng để hỗ trợ sau này.

1.5.7. Quy trình quản lý

1. Sao lưu dữ liệu định kỳ
 - Bao gồm thông tin khách hàng, phiếu sửa chữa, phiếu bảo hành, kho linh kiện, thông tin nhân viên, hóa đơn.
2. Khôi phục dữ liệu khi cần
 - Khôi phục dữ liệu khi xảy ra sự cố hoặc mất dữ liệu.
 - Kiểm tra tính toàn vẹn dữ liệu sau khi phục hồi.
3. Thông kê và báo cáo
 - Thông kê doanh thu, số lượng máy sửa chữa, linh kiện xuất/nhập, lỗi thường gặp.
 - Báo cáo định kỳ theo ngày/tuần/tháng/quý.
 - Theo dõi hiệu suất làm việc của kỹ thuật viên, thủ kho và tiếp tân.
4. Quản lý lịch làm việc nhân viên
 - Lập lịch làm việc, theo dõi nghỉ phép, tăng ca, phân công công việc hợp lý.
 - Cập nhật và thông báo thay đổi lịch kịp thời.

1.6. Các biểu mẫu thu thập trong quá trình khảo sát



Minh Tuấn Mobile

Dịch vụ sửa chữa & bảo hành điện thoại

PHIẾU BẢO HÀNH

1. Thông tin khách hàng

Họ tên:

Số điện thoại:

2. Thông tin máy

Hãng máy:

Model:

IMEI:

3. Nội dung bảo hành

Hạng mục:

Ngày tiếp nhận:

Ngày hẹn trả:

4. Ghi chú

KHÁCH HÀNG

(Ký & ghi rõ họ tên)

NHÂN VIÊN

(Ký & ghi rõ họ tên)

Hình 1.1: Phiếu bảo hành.



Minh Tuấn Mobile

Dịch vụ sửa chữa & bảo hành điện thoại

PHIẾU SỬA CHỮA

1. Thông tin khách hàng

Họ tên:

Số điện thoại:

2. Thông tin máy

Hãng máy:

Model:

IMEI:

3. Nội dung sửa chữa

Hạng mục:

Mô tả lỗi:

Linh kiện đã thay:

Ngày tiếp nhận:

Ngày hẹn trả:

4. Ghi chú

.....
.....
.....

KHÁCH HÀNG

(Ký & ghi rõ họ tên)

NHÂN VIÊN

(Ký & ghi rõ họ tên)

Hình 1.2: Phiếu sửa chữa.



Minh Tuấn Mobile

Dịch vụ sửa chữa & bảo hành điện thoại

HÓA ĐƠN

Mã phiếu:

Ngày:

Ghi chú:

Tên linh kiện	Hãng	Serial	Giá (VNĐ)
.....
.....
.....
Tổng cộng:		

Khách hàng ký

(Ký & ghi rõ họ tên)

Nhân viên xác nhận

(Ký & ghi rõ họ tên)

Hình 1.3: Phiếu hóa đơn.

CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG

2.1. Giới thiệu

Giai đoạn phân tích hệ thống đóng vai trò then chốt trong quy trình phát triển hệ thống thông tin, là nền tảng quyết định sự thành công của toàn bộ dự án. Đây là giai đoạn mà nhóm phát triển tiến hành nghiên cứu sâu về đối tượng cần quản lý, hiểu rõ các quy trình nghiệp vụ hiện tại và xác định những yêu cầu cụ thể mà hệ thống cần đáp ứng.

Tầm quan trọng của giai đoạn phân tích thể hiện ở việc nó giúp xác định chính xác phạm vi và mục tiêu của hệ thống, tránh được những hiểu lầm và sai sót có thể dẫn đến thất bại trong các giai đoạn sau. Thông qua việc phân tích nghiệp vụ, thu thập và phân tích yêu cầu, nhóm phát triển có thể đưa ra những quyết định thiết kế phù hợp, đảm bảo hệ thống đáp ứng đúng nhu cầu thực tế của người sử dụng.

Đối với hệ thống quản lý sửa chữa và bảo hành điện thoại, giai đoạn phân tích sẽ tập trung vào việc nghiên cứu các hoạt động quản lý hiện tại, xác định các tác nhân tham gia, phân tích quy trình nghiệp vụ và đặc tả yêu cầu chức năng cũng như phi chức năng. Kết quả của giai đoạn này sẽ là cơ sở vững chắc cho việc thiết kế kiến trúc hệ thống và xây dựng giải pháp tối ưu trong các chương tiếp theo.

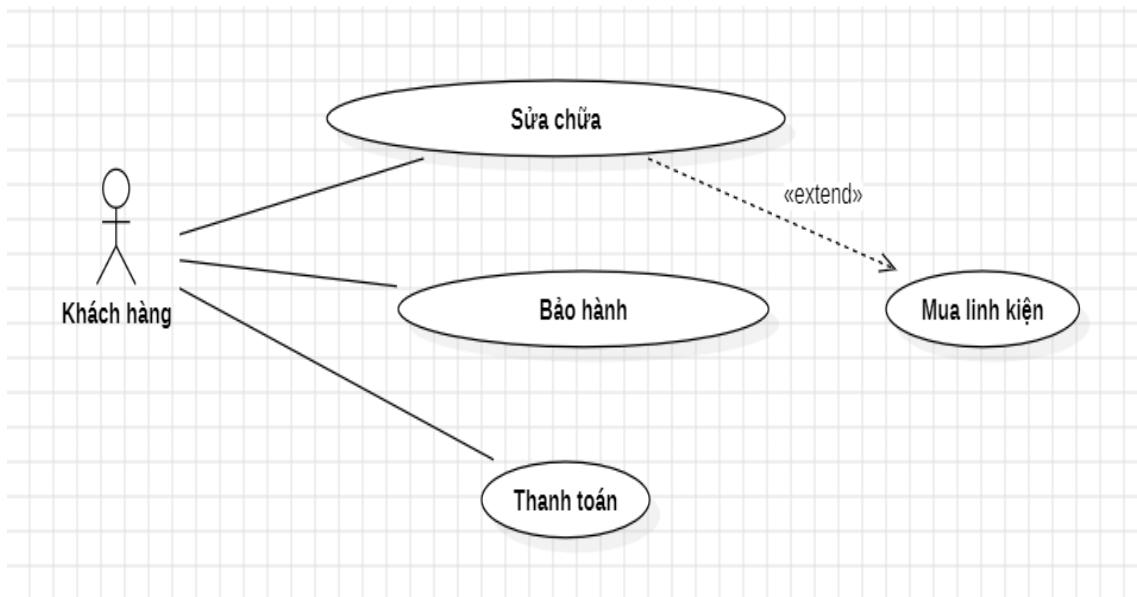
2.2. Mô hình hóa quy trình nghiệp vụ

2.2.1. Sơ đồ usecase nghiệp vụ

Một cửa hàng sửa chữa điện thoại cần xây dựng hệ thống quản lý quy trình sửa chữa và bảo hành cho khách hàng. Khi khách hàng mang thiết bị đến cửa hàng, bộ phận tiếp tân sẽ ghi nhận thông tin chi tiết của thiết bị như loại máy, thời gian mua, và tình trạng bảo hành, đồng thời giới thiệu các dịch vụ đang phục vụ tại cửa hàng. Nhân viên cũng sẽ giải thích quy trình tiếp nhận và dự kiến thời gian xử lý cho khách hàng. Sau khi xác minh thông tin ban đầu, tiếp tân lập phiếu tiếp nhận máy, ghi rõ lỗi và tình trạng hiện tại, rồi chuyển thiết bị kèm phiếu cho kỹ thuật viên để kiểm tra chi tiết. Kỹ thuật viên sẽ đánh giá toàn bộ tình trạng máy, bao gồm phần cứng như màn hình, pin, bo mạch, camera, cổng sạc, và cả phần mềm, dữ liệu hoặc các lỗi hệ thống,... Từ đó, kỹ thuật viên xác định nguyên nhân sự cố, các linh kiện cần thay và các bước sửa chữa cần thực hiện.

Sau khi có kết quả kiểm tra, kỹ thuật viên báo cáo tình trạng máy lại cho tiếp tân để thông báo cho khách hàng. Nếu thiết bị còn bảo hành, khách hàng sẽ được sửa chữa miễn phí; nếu không, hệ thống sẽ cung cấp phí dịch vụ, chi phí linh kiện và các khoản phát sinh nếu có. Khi khách hàng đồng ý với báo giá, tiếp tân ghi nhận xác nhận này và thông báo ngày dự kiến hoàn thành.

Trong trường hợp thiết bị cần thay linh kiện, kỹ thuật viên lập yêu cầu xuất linh kiện theo đúng phiếu sửa chữa. Thủ kho kiểm tra số lượng tồn kho để xác định khả năng đáp ứng. Nếu linh kiện còn đủ, thủ kho sẽ tiến hành xuất linh kiện và cập nhật số lượng tồn. Nếu linh kiện thiếu, thủ kho sẽ thông báo lại cho kỹ thuật viên và tiếp tân về thời gian nhập linh kiện. Sau khi linh kiện được nhập về từ nhà cung cấp, thủ kho kiểm tra số lượng, chất lượng, hóa đơn và thông tin bảo hành của linh kiện, sau đó cập nhật vào kho và sắp xếp theo đúng quy chuẩn. Khi linh kiện đã sẵn sàng, kỹ thuật viên nhận linh kiện từ thủ kho và tiến hành công việc sửa chữa theo quy trình, đảm bảo thiết bị của khách hàng được xử lý đúng kỹ thuật và trong thời gian cam kết.



Hình 2.1: Sơ đồ usecase nghiệp vụ.

2.2.2. Mô hình hóa quy trình nghiệp vụ

a) Mô hình hóa quy trình nghiệp vụ sửa chữa

Use case nghiệp vụ: Sửa chữa

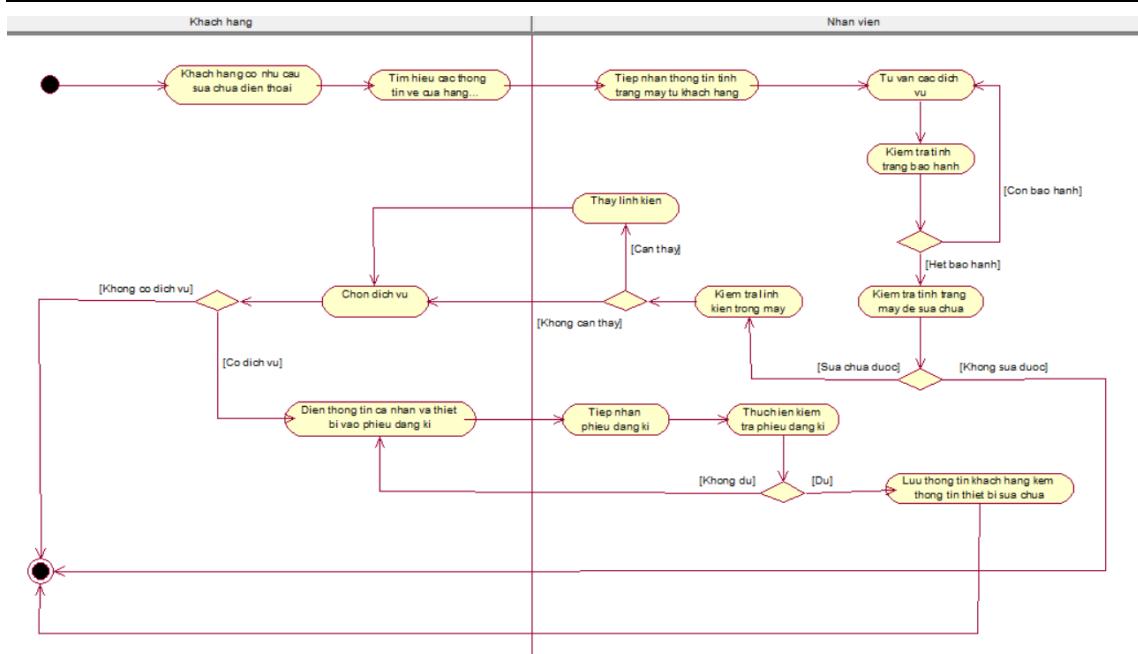
Use case này mô tả quy trình tiếp tân tiếp nhận khách hàng mang thiết bị đến trung tâm để yêu cầu kiểm tra sửa chữa.

Các dòng cơ bản:

1. Khách hàng mang thiết bị bị hỏng đến trung tâm.
2. Tiếp tân tiếp nhận thiết bị, ghi nhận thông tin khách hàng và tình trạng lỗi ban đầu.
3. Kỹ thuật viên kiểm tra và chẩn đoán nguyên nhân hư hỏng.
4. Tiếp tân thông báo cho khách hàng về tình trạng và chi phí dự kiến.
5. Khách hàng đồng ý tiến hành sửa chữa.
6. Kỹ thuật viên thực hiện sửa chữa, kiểm tra hoạt động sau sửa.
7. Tiếp tân thông báo cho khách hàng đến nhận lại thiết bị.

Các dòng thay thế:

- 2a. Nếu khách hàng không cung cấp đủ thông tin thiết bị, tiếp tân hướng dẫn bổ sung hoặc ghi nhận tạm thời để hoàn thiện sau.
- 4a. Nếu linh kiện trong thiết bị hư hỏng quá nặng, tiếp tân thông báo cho khách hàng và đề xuất mua thêm linh kiện



Hình 2.2: Sơ đồ nghiệp vụ thanh toán.

b) Mô hình hóa quy trình nghiệp vụ bảo hành

Use case nghiệp vụ: Bảo hành

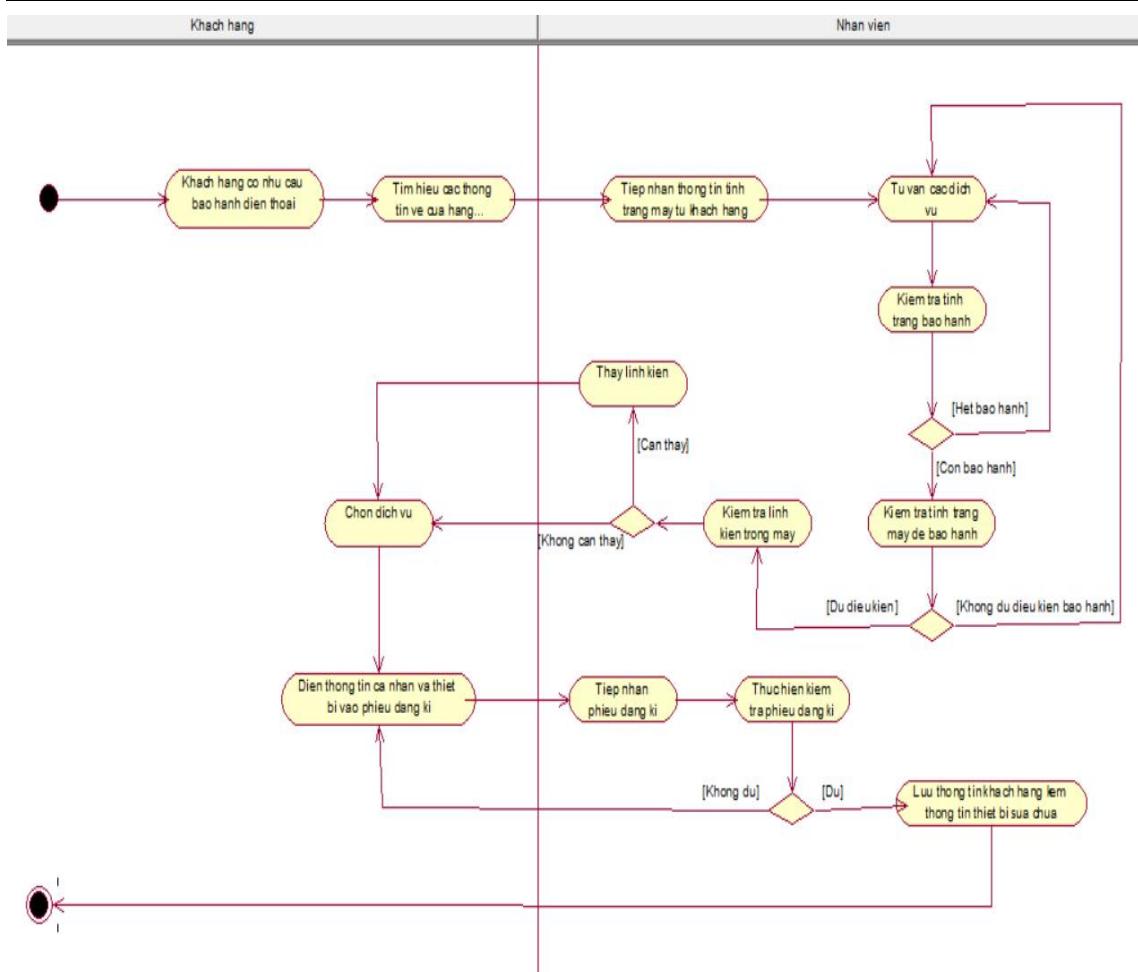
Use case này mô tả quy trình tiếp tân tiếp nhận khách hàng mang thiết bị đến trung tâm để bảo hành khi thiết bị còn trong giai đoạn bảo hành.

Các dòng cơ bản:

1. Khách hàng mang thiết bị đến trung tâm để yêu cầu bảo hành.
2. Tiếp tân kiểm tra thông tin bảo hành (số serial, ngày mua, điều kiện bảo hành).
3. Kỹ thuật viên kiểm tra lỗi và tiến hành sửa chữa/bảo hành miễn phí nếu hợp lệ.
4. Sau khi hoàn tất, tiếp tân thông báo khách hàng đến nhận lại thiết bị.

Các dòng thay thế:

- 2a. Nếu thiết bị hết hạn bảo hành, tiếp tân thông báo và đề xuất chuyển sang usecase sửa chữa.
- 2b. Nếu phát hiện thiết bị không đủ điều kiện bảo hành (mất tem, sửa ngoài), tiếp tân thông báo cho khách và ghi chú tình trạng đặc biệt.



Hình 2.3: Sơ đồ nghiệp vụ bảo hành.

c) Mô hình hóa quy trình nghiệp vụ thanh toán

Use case nghiệp vụ: Thanh toán

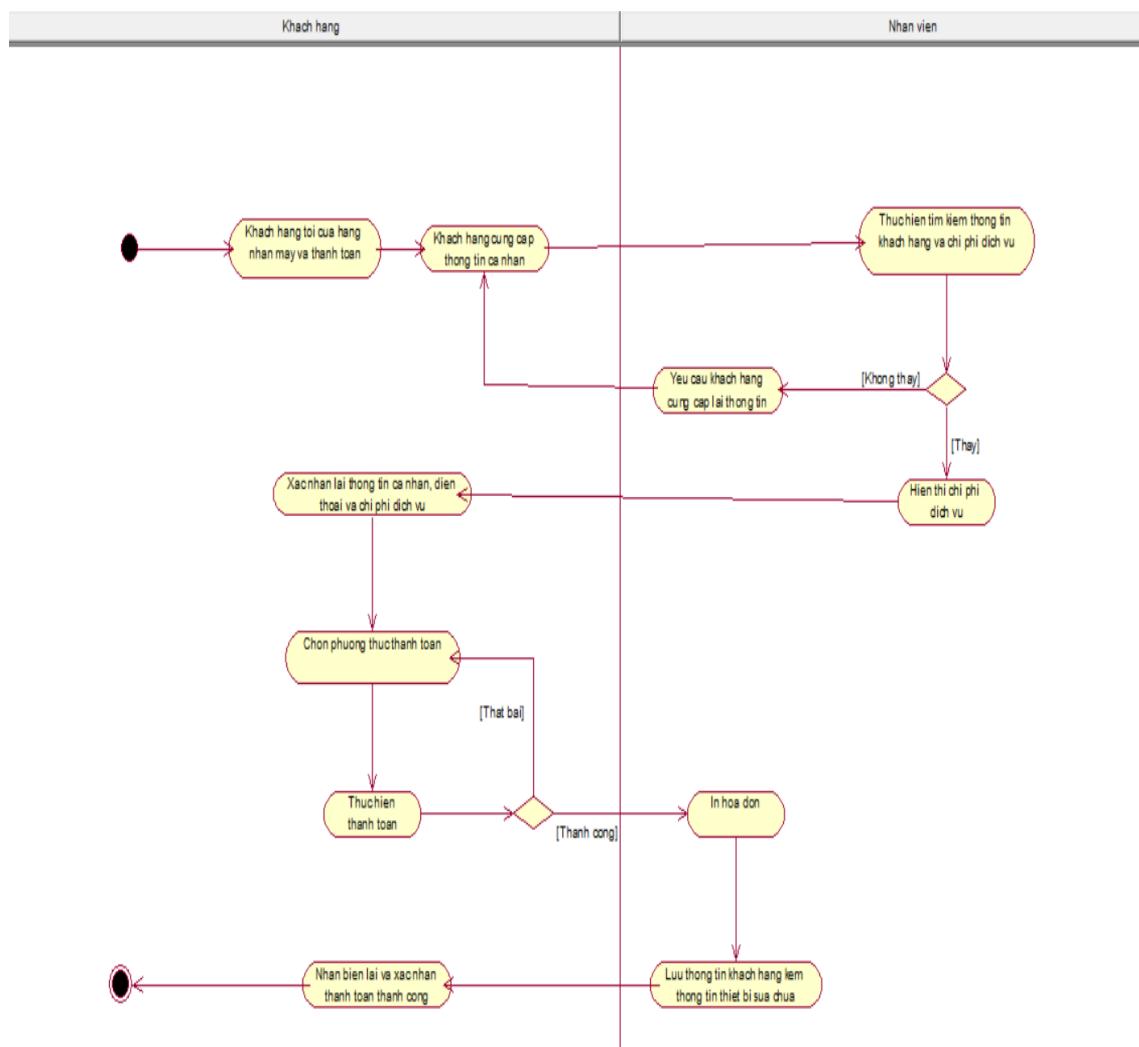
Use case này mô tả quy trình khách hàng thanh toán chi phí sau khi hoàn tất sửa chữa hoặc thay thế linh kiện.

Các dòng cơ bản:

1. Tiếp tân lập hóa đơn sau khi sửa.
2. Khách hàng kiểm tra và xác nhận thanh toán.
3. Tiếp tân nhận tiền và in hóa đơn cho khách hàng.

Các dòng thay thế:

- 2a. Nếu khách hàng không đồng ý với số tiền, tiếp tân kiểm tra lại thông tin hóa đơn và điều chỉnh nếu có sai sót.



Hình 2.4: Sơ đồ nghiệp vụ thanh toán.

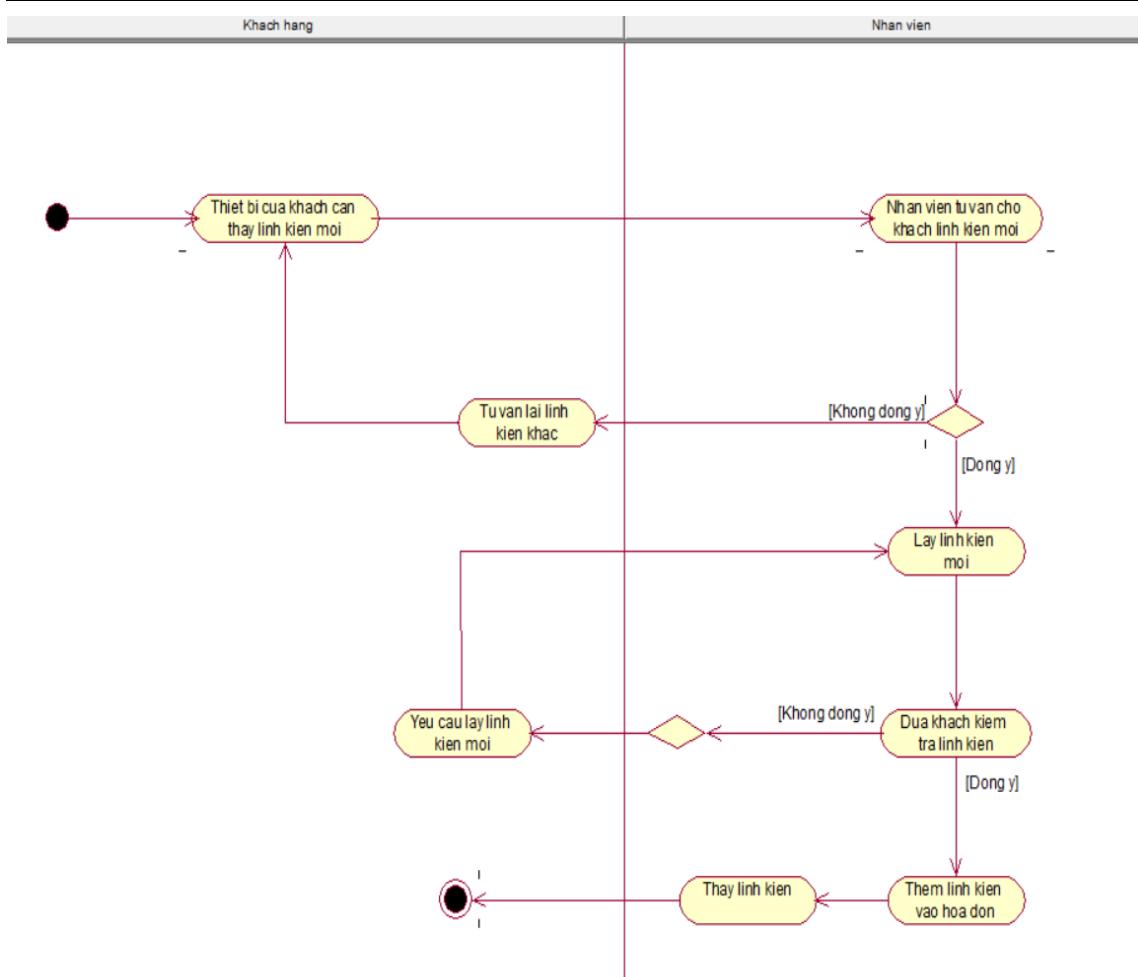
d) Mô hình hóa quy trình nghiệp vụ mua linh kiện

Use case nghiệp vụ: Thanh toán

Use case này mô tả quy trình khách hàng cần phải mua linh kiện mới do linh kiện cũ bị hư.

Các dòng cơ bản:

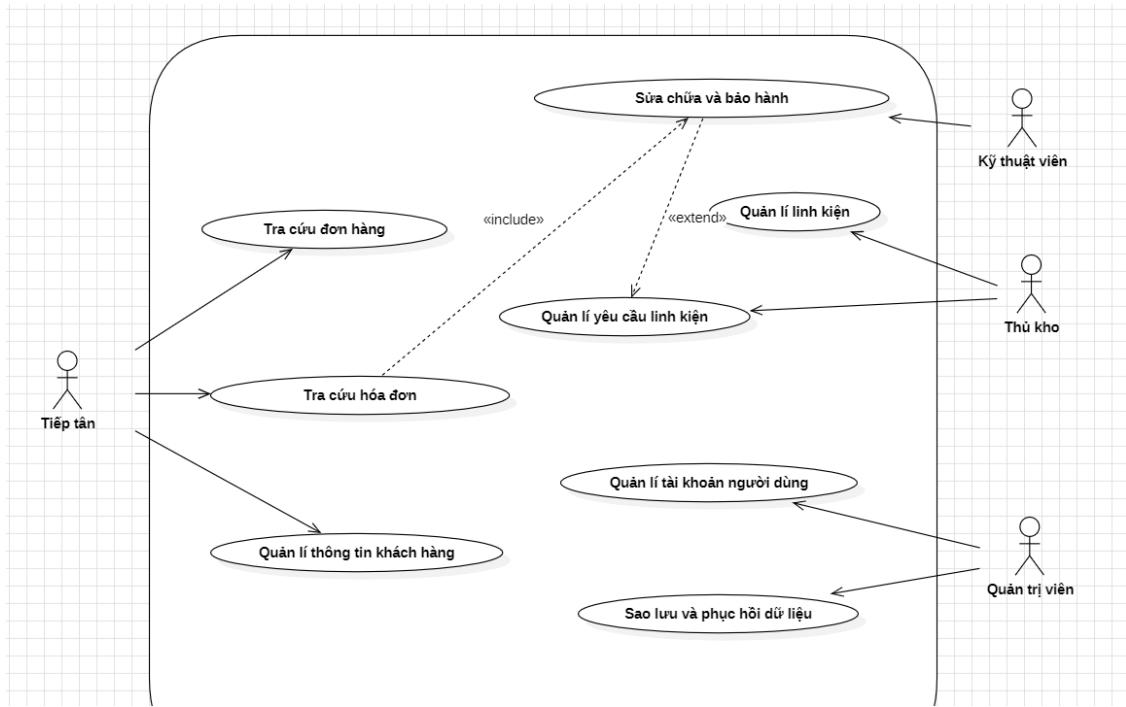
1. Kỹ thuật viên xác định linh kiện trong thiết bị đã bị hư và cần thay thế.
2. Kỹ thuật viên thông báo cho tiếp tân về loại linh kiện cần thay để báo khách hàng.
3. Tiếp tân nhận tiền và in hóa đơn cho khách hàng.
4. Tiếp tân lập hóa đơn cho phần linh kiện và cập nhật vào chi phí sửa chữa của khách hàng.



Hình 2.5: Sơ đồ nghiệp vụ mua linh kiện.

2.3. Mô hình hóa chức năng hệ thống

2.3.1. Sơ đồ usecase hệ thống



Hình 2.6: Sơ đồ usecase hệ thống.

2.3.2. Đặc tả usecase hệ thống

Bảng 2.1: Bảng đặc tả usecase tra cứu đơn hàng.

Tên usecase	Tra cứu đơn hàng.
Tóm tắt	Use case này cho phép nhân viên tiếp tân tra cứu các đơn theo nhiều tiêu chí như mã đơn, tên khách, hoặc ngày tạo, để phục vụ in hóa đơn, kiểm tra tiến độ.
Tác nhân	Tiếp tân.
Use Case liên quan	Tra cứu hóa đơn.
Dòng sự kiện chính	<ul style="list-style-type: none"> Nhân viên tiếp tân đăng nhập hệ thống. Vào giao diện “Tra cứu đơn hàng”. Nhập tiêu chí tìm kiếm: mã đơn hàng, số điện thoại khách hàng, hoặc ngày tiếp nhận. Hệ thống truy vấn và hiển thị danh sách các đơn hàng phù hợp. Tiếp tân chọn một đơn hàng cụ thể để xem chi tiết tình trạng.
Dòng sự kiện phụ	Không nhập thông tin tìm kiếm thì hệ thống yêu cầu “Vui lòng nhập ít nhất một tiêu chí tìm kiếm.”

	Không tìm thấy đơn hàng thì hiển thị “Không có kết quả phù hợp.”
Điều kiện tiên quyết	<ul style="list-style-type: none"> - Tiếp tân đã đăng nhập. - Dữ liệu đơn hàng tồn tại.
Hậu điều kiện	Nhân viên tiếp tân xem được thông tin chi tiết đơn hàng để hỗ trợ khách.

Bảng 2.2: Bảng đặc tả usecase tra cứu hóa đơn.

Tên usecase	Tra cứu hóa đơn.
Tóm tắt	Use case này cho phép nhân viên tiếp tân tìm kiếm, xem và xác minh thông tin đơn hàng sửa chữa hoặc bảo hành của khách hàng để hỗ trợ tra cứu nhanh
Tác nhân	Tiếp tân.
Các usecase liên quan	Tra cứu hóa đơn.
Dòng sự kiện chính	<ul style="list-style-type: none"> – Nhân viên tiếp tân chọn mục “Tra cứu hóa đơn”. – Nhập mã đơn hoặc số điện thoại khách hàng. – Hệ thống tìm kiếm trong cơ sở dữ liệu. – Hiển thị danh sách hóa đơn phù hợp. – Tiếp tân chọn hóa đơn để xem chi tiết. – Hệ thống hiển thị thông tin khách, thiết bị, chi phí, linh kiện.
Dòng sự kiện phụ	<p>Nếu không nhập đủ dữ liệu thì hệ thống sẽ báo “Thiếu thông tin tìm kiếm”.</p> <p>Nếu không tìm thấy thì hệ thống sẽ hiển thị “Không có hóa đơn phù hợp.”</p>
Điều kiện tiên quyết	Nhân viên tiếp tân đã đăng nhập thành công vào hệ thống. Dữ liệu hóa đơn tồn tại.
Hậu điều kiện	<p>Hóa đơn hoặc phiếu giao nhận được in thành công.</p> <p>Thông tin bản in được lưu lại trong hệ thống (ngày giờ, người in, nội dung).</p> <p>Khách hàng nhận được chứng từ giấy xác nhận giao dịch.</p>

Bảng 2.3: Bảng đặc tả usecase quản lý thông tin khách hàng.

Tên usecase	Quản lý thông tin khách hàng.
Tóm tắt	Use case này cho phép nhân viên tiếp tân thực hiện các thao tác quản lý thông tin khách hàng trong hệ thống, bao gồm thêm mới, chỉnh sửa, xóa và tra cứu thông tin khách hàng.

	Mục tiêu nhằm đảm bảo dữ liệu khách hàng luôn chính xác, phục vụ cho quá trình tiếp nhận, sửa chữa và bảo hành thiết bị.
Tác nhân	Tiếp tân.
Usecase liên quan	Tra cứu đơn hàng. Tra cứu hóa đơn. Sửa chữa và bảo hành.
Dòng sự kiện chính	<ul style="list-style-type: none"> – Tác nhân đăng nhập vào hệ thống. – Truy cập mục “Quản lý khách hàng” trên giao diện chính. – Hệ thống hiển thị danh sách khách hàng hiện có (tên, số điện thoại, địa chỉ, số lần sửa chữa, trạng thái). – Tiếp tân chọn thao tác: thêm mới khách hàng, chỉnh sửa thông tin khách hàng, xóa khách hàng, tìm kiếm khách hàng theo tên, số điện thoại hoặc mã khách hàng. – Nếu chọn thêm mới, hệ thống hiển thị form nhập thông tin: họ tên, số điện thoại, địa chỉ, email, ghi chú. – Hệ thống kiểm tra dữ liệu hợp lệ (đủ thông tin bắt buộc, số điện thoại đúng định dạng, không trùng mã khách hàng). – Nếu hợp lệ, hệ thống lưu thông tin vào cơ sở dữ liệu và hiển thị thông báo “Lưu thông tin khách hàng thành công.” – Danh sách khách hàng được cập nhật lại, hiển thị khách hàng vừa được thêm hoặc chỉnh sửa.
Dòng sự kiện phụ	<ul style="list-style-type: none"> – Bỏ trống trường bắt buộc (tên, số điện thoại), hệ thống thông báo “Vui lòng nhập đầy đủ thông tin bắt buộc.” – Số điện thoại đã tồn tại, hệ thống hiển thị “Khách hàng này đã có trong hệ thống.” – Lỗi kết nối cơ sở dữ liệu, hệ thống hiển thị “Không thể lưu thông tin, vui lòng thử lại.” – Khi xóa khách hàng, hệ thống yêu cầu xác nhận. – Khách hàng đang có phiếu tiếp nhận/sửa chữa liên quan, hệ thống từ chối xóa và hiển thị “Không thể xóa khách hàng đang có giao dịch liên quan.”
Điều kiện tiên quyết	<ul style="list-style-type: none"> – Tiếp tân đã đăng nhập thành công vào hệ thống. – Cơ sở dữ liệu khách hàng hoạt động ổn định. – Tác nhân có quyền quản lý (thêm, sửa, xóa) thông tin khách hàng.
Hậu điều kiện	<ul style="list-style-type: none"> – Thông tin khách hàng được thêm mới, chỉnh sửa hoặc xóa thành công.

	<ul style="list-style-type: none"> - Dữ liệu khách hàng trong hệ thống được cập nhật chính xác và đồng bộ với các module liên quan (phiếu tiếp nhận, sửa chữa, bảo hành). - Lịch sử thay đổi thông tin khách hàng được ghi nhận để phục vụ kiểm tra và quản lý.
--	---

Bảng 2.4: Bảng đặc tả usecase sửa chữa và bảo hành.

Tên usecase	Xử lý phiếu sửa chữa và bảo hành.
Tóm tắt	Use case này mô tả quá trình kỹ thuật viên thực hiện kiểm tra, sửa chữa hoặc bảo hành thiết bị và cập nhật kết quả xử lý.
Tác nhân	Kỹ thuật viên.
Usecase liên quan	Tra cứu đơn hàng.
Dòng sự kiện chính	<ul style="list-style-type: none"> - Kỹ thuật viên truy cập danh sách thiết bị đang chờ xử lý. - Chọn thiết bị để xem thông tin chi tiết và lỗi mô tả. - Kiểm tra tình trạng thực tế của thiết bị. - Nếu linh kiện hư chuyển qua usecase “Quản lý yêu cầu linh kiện”. - Sau khi nhận linh kiện, kỹ thuật viên tiến hành sửa chữa. - Kiểm tra lại hoạt động sau sửa. - Cập nhật kết quả: “Đã sửa xong”, “Không sửa được”, “Bảo hành hoàn tất”. - Hệ thống lưu trạng thái và gửi thông báo cho tiếp tân
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Nếu thiết bị không thể sửa cập nhật “Không sửa được”. Nếu lỗi mạng thông báo “Không thể cập nhật trạng thái”.
Điều kiện tiên quyết	<ul style="list-style-type: none"> - Kỹ thuật viên đã đăng nhập và được phân công xử lý thiết bị. - Thiết bị đã được tiếp nhận và có phiếu tiếp nhận trong hệ thống. - Các linh kiện cần thiết được cập nhật trong kho.
Hậu điều kiện	<ul style="list-style-type: none"> - Thiết bị có trạng thái mới trong hệ thống.

Bảng 2.5: Bảng đặc tả usecase quản lý linh kiện.

Tên usecase	Quản lý linh kiện.
Tóm tắt	Use case này cho phép quản lý phụ trách linh kiện thực hiện các thao tác thêm mới, cập nhật, xóa, tra cứu và kiểm soát tồn kho của linh kiện trong hệ thống.
Tác nhân	Thủ kho.
Usecase liên quan	Sửa chữa và bảo hành.
Dòng sự kiện chính	<ul style="list-style-type: none"> - Thủ kho đăng nhập vào hệ thống. - Truy cập mục “Quản lý linh kiện”.

	<ul style="list-style-type: none"> - Hệ thống hiển thị danh sách linh kiện hiện có (mã linh kiện, tên, loại, số lượng tồn, giá nhập, giá xuất). - Thủ kho có thể chọn thao tác: thêm mới linh kiện, cập nhật thông tin linh kiện (giá, số lượng, nhà cung cấp, trạng thái), xóa linh kiện, tìm kiếm linh kiện theo tên, mã hoặc loại linh kiện. - Nếu chọn Thêm mới linh kiện, hệ thống hiển thị form nhập thông tin: Tên linh kiện, mã linh kiện, loại linh kiện, đơn vị tính, giá nhập, giá bán, nhà cung cấp, số lượng ban đầu. - Thủ kho nhập dữ liệu và nhấn “Lưu”. - Hệ thống kiểm tra dữ liệu hợp lệ (mã không trùng, đủ thông tin bắt buộc, giá và số lượng hợp lý). - Nếu hợp lệ, hệ thống lưu dữ liệu vào cơ sở dữ liệu và hiển thị thông báo “Thêm linh kiện thành công.” - Khi có phiếu sửa chữa cần thay linh kiện, thủ kho có thể trừ tồn kho. - Hệ thống cập nhật lại số lượng tồn, đồng thời cho phép xuất báo cáo thống kê linh kiện (tồn kho, đã sử dụng, cần nhập thêm).
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Nhập thiếu thông tin bắt buộc, hệ thống hiển thị “Vui lòng nhập đầy đủ thông tin linh kiện.” - Mã linh kiện bị trùng, hệ thống hiển thị “Mã linh kiện đã tồn tại trong hệ thống.” - Lỗi kết nối cơ sở dữ liệu, hệ thống hiển thị “Không thể lưu dữ liệu, vui lòng thử lại.” - Linh kiện tồn kho < 0, hệ thống cảnh báo “Số lượng linh kiện không đủ để xuất kho.” - Khi xóa linh kiện đang được sử dụng trong phiếu sửa chữa, hệ thống từ chối và hiển thị “Không thể xóa linh kiện đang được sử dụng.”
Điều kiện tiên quyết	<ul style="list-style-type: none"> - Quản lý kho đã đăng nhập thành công vào hệ thống. - Hệ thống có danh sách nhà cung cấp và các loại linh kiện đã định nghĩa. - Cơ sở dữ liệu kho linh kiện hoạt động bình thường.
Hậu điều kiện	<ul style="list-style-type: none"> - Thông tin linh kiện được thêm mới, chỉnh sửa hoặc xóa thành công. - Số lượng tồn kho được cập nhật chính xác sau mỗi lần sửa chữa, nhập hoặc xuất linh kiện. - Dữ liệu có thể được sử dụng cho báo cáo thống kê, kiểm kê và đặt hàng nhà cung cấp.

Bảng 2.6: Bảng đặc tả usecase quản lý yêu cầu linh kiện.

Tên usecase	Quản lý yêu cầu linh kiện.
Tóm tắt	Use case này cho phép nhân viên gửi yêu cầu sử dụng linh kiện khi sửa chữa điện thoại, và thủ kho có thể xem, duyệt, hoặc từ chối các yêu cầu đó, nhằm bảo vệ việc cấp phát linh kiện

	đúng quy trình, tránh thất thoát và luôn cập nhật tồn kho chính xác.
Tác nhân	Thủ kho, kĩ thuật viên.
Usecase liên quan	Quản lý linh kiện.
Dòng sự kiện chính	<ul style="list-style-type: none"> – Kỹ thuật viên đăng nhập vào hệ thống. – Chọn chức năng “Yêu cầu linh kiện”. – Hệ thống hiển thị danh sách phiếu sửa chữa mà kỹ thuật viên đang phụ trách. – Kỹ thuật viên chọn phiếu sửa chữa cần bổ sung linh kiện. – Nhập thông tin yêu cầu linh kiện: Mã linh kiện hoặc tên linh kiện, số lượng – Nhấn “Gửi yêu cầu” sau đó hệ thống kiểm tra dữ liệu hợp lệ. – Nếu hợp lệ, hệ thống tạo yêu cầu với trạng thái “Chờ duyệt” và lưu vào cơ sở dữ liệu. – Thủ kho đăng nhập và truy cập mục “Duyệt yêu cầu linh kiện”. – Hệ thống hiển thị danh sách yêu cầu từ kỹ thuật viên. – Thủ kho chọn một yêu cầu để xem chi tiết (thông tin linh kiện, số lượng, phiếu sửa chữa liên quan). – Thủ kho chọn “Duyệt” hoặc “Từ chối”: – Hệ thống gửi thông báo cho kỹ thuật viên về kết quả duyệt.
Dòng sự kiện phụ	<ul style="list-style-type: none"> – Nhập thiếu thông tin thì hệ thống hiển thị “Vui lòng nhập đầy đủ thông tin yêu cầu.” – Linh kiện không tồn tại trong danh mục thì hệ thống hiển thị “Linh kiện chưa có trong kho.” – Số lượng yêu cầu vượt quá tồn kho hệ thống cảnh báo “Không đủ tồn kho để cấp phát.”
Điều kiện tiên quyết	<ul style="list-style-type: none"> – Cả kỹ thuật viên và thủ kho đã đăng nhập hợp lệ vào hệ thống. – Hệ thống có danh sách linh kiện và dữ liệu tồn kho được quản lý. – Các phiếu sửa chữa của kỹ thuật viên đã được tạo trong hệ thống.
Hậu điều kiện	<ul style="list-style-type: none"> – Yêu cầu linh kiện được lưu và có trạng thái tương ứng (Chờ duyệt, Đã duyệt, Từ chối). – Nếu được duyệt, số lượng linh kiện trong kho được trừ tương ứng. – Lịch sử yêu cầu được ghi nhận, phục vụ thống kê và truy vết sau này.

Bảng 2.7: Bảng đặc tả usecase quản lý tài khoản người dùng.

Tên usecase	Quản lý tài khoản người dùng.
-------------	-------------------------------

Tóm tắt	Use case này cho phép quản trị viên thực hiện các chức năng quản lý tài khoản trong hệ thống, bao gồm: thêm xóa sửa, khóa hoặc mở tài khoản.
Tác nhân	Quản trị viên.
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên đăng nhập vào hệ thống. - Từ giao diện chính, chọn mục “Quản lý tài khoản người dùng”. - Hệ thống hiển thị danh sách tài khoản hiện có, bao gồm: tên đăng nhập, họ tên, vai trò (Kỹ thuật viên, Thủ kho, Nhân viên tiếp nhận, Quản lý, v.v.), Trạng thái (Đang hoạt động / Đã khóa). - Quản trị viên chọn thao tác: Thêm mới / Chính sửa / Khóa / Xóa tài khoản. - Hệ thống kiểm tra dữ liệu hợp lệ: tên đăng nhập không trùng, mật khẩu đạt yêu cầu bảo mật. - Hệ thống thực hiện các yêu cầu xóa sửa,... và lưu vào cơ sở dữ liệu. - Hệ thống hiển thị thông báo “Thao tác thành công.”. - Danh sách tài khoản được cập nhật lại trên giao diện.
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Nhập thiếu thông tin khi tạo tài khoản Hệ thống hiển thị “Vui lòng nhập đầy đủ thông tin.”. - Khi xóa tài khoản → Hệ thống hiển thị hộp thoại “Bạn có chắc muốn xóa tài khoản này không?”. - Tên đăng nhập đã tồn tại → Hệ thống cảnh báo “Tên đăng nhập đã được sử dụng.”. - Nếu tài khoản đang được sử dụng trong phiếu / hoạt động thì hệ thống hiển thị “Không thể xóa tài khoản đang hoạt động.”
Điều kiện tiên quyết	<ul style="list-style-type: none"> - Quản trị viên đã đăng nhập thành công vào hệ thống. - Cơ sở dữ liệu người dùng đang hoạt động ổn định. - Hệ thống có bảng phân quyền người dùng (Role, Permission).
Hậu điều kiện	<ul style="list-style-type: none"> - Thông tin tài khoản người dùng được thêm, chỉnh sửa, khóa hoặc xóa theo thao tác của quản trị viên. - Hệ thống ghi nhận lịch sử thay đổi (ai tạo, ai xóa, thời gian thao tác) để phục vụ kiểm tra và bảo mật.

Bảng 2.8: Bảng đặc tả usecase sao lưu và phục hồi dữ liệu.

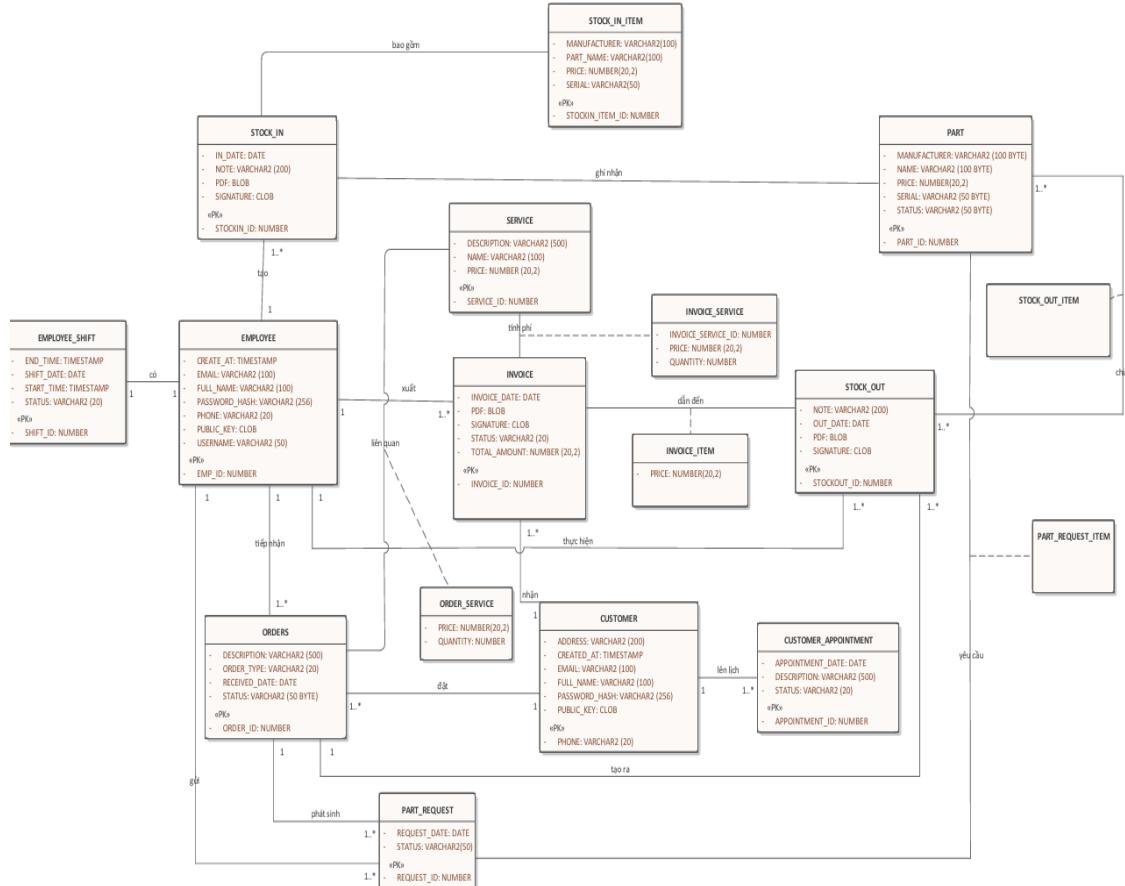
Tên usecase	Sao lưu và phục hồi dữ liệu.
Tóm tắt	Use case này cho phép quản trị viên thực hiện các thao tác sao lưu toàn bộ cơ sở dữ liệu của hệ thống (gồm thông tin khách hàng, phiếu tiếp nhận, phiếu sửa chữa, linh kiện, tài khoản người dùng, v.v.) ra tệp sao lưu định kỳ.
Tác nhân	Quản trị viên.

Usecase liên quan	Quản lý tài khoản người dùng. Quản lý linh kiện. Sửa chữa và bảo hành
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên đăng nhập vào hệ thống với quyền quản trị. - Trên giao diện quản trị, chọn chức năng “Sao lưu & phục hồi dữ liệu”. - Hệ thống hiển thị hai lựa chọn: Sao lưu dữ liệu hoặc phục hồi dữ liệu. - Nếu chọn Sao lưu dữ liệu thì hệ thống yêu cầu chọn vị trí lưu trữ tệp sao lưu, xác nhận thao tác và tiến hành sao chép toàn bộ dữ liệu vào tệp sao lưu. - Nếu chọn Phục hồi dữ liệu thì hệ thống yêu cầu chọn tệp sao lưu hiện có, hiển thị cảnh báo ghi đè dữ liệu và sau khi xác nhận, hệ thống tiến hành phục hồi dữ liệu từ tệp đó. - Sau khi hoàn tất, hệ thống thông báo kết quả thao tác (thành công hoặc lỗi), đồng thời ghi log lại hành động của quản trị viên.
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Quản trị viên hủy thao tác thì hệ thống quay về giao diện chính. - File sao lưu không hợp lệ hoặc bị lỗi thì hệ thống hiển thị “Không thể phục hồi, tệp sao lưu không hợp lệ.”. - Không đủ dung lượng lưu trữ khi sao lưu thì hệ thống báo lỗi “Không đủ dung lượng để lưu tệp sao lưu”. - Lỗi kết nối cơ sở dữ liệu thì hệ thống hiển thị “Thao tác thất bại, vui lòng thử lại.”.
Điều kiện tiên quyết	Quản trị viên đã đăng nhập vào hệ thống. Hệ thống đang hoạt động ổn định và có quyền truy cập vào cơ sở dữ liệu. Có đủ dung lượng để lưu trữ hoặc đọc file sao lưu.
Hậu điều kiện	<ul style="list-style-type: none"> - Nếu sao lưu: Hệ thống tạo tệp sao lưu dữ liệu đầy đủ và an toàn. - Nếu phục hồi: Dữ liệu trong hệ thống được khôi phục về trạng thái tại thời điểm bản sao lưu. - Hệ thống ghi nhận lịch sử thao tác (ngày, giờ, người thực hiện, kết quả).

2.4. Sơ đồ lớp ở mức phân tích

Sơ đồ lớp mức phân tích (Analysis Class Diagram) được sử dụng nhằm mô tả các đối tượng dữ liệu chính trong hệ thống và mối quan hệ giữa chúng ở góc độ nghiệp vụ. Đây là bước cầu nối giữa mô hình hóa chức năng hệ thống (các Use Case) và mô hình thiết kế chi tiết ở giai đoạn tiếp theo.

Sơ đồ lớp phân tích vì vậy đóng vai trò như bản đồ dữ liệu tổng thể của hệ thống, giúp xác định rõ các đối tượng sẽ được quản lý, các mối quan hệ giữa chúng, và là cơ sở quan trọng để xây dựng mô hình cơ sở dữ liệu (ERD) cũng như chuyển sang sơ đồ lớp thiết kế ở chương tiếp theo.



Hình 2.7: Sơ đồ lớp ở mức phân tích.

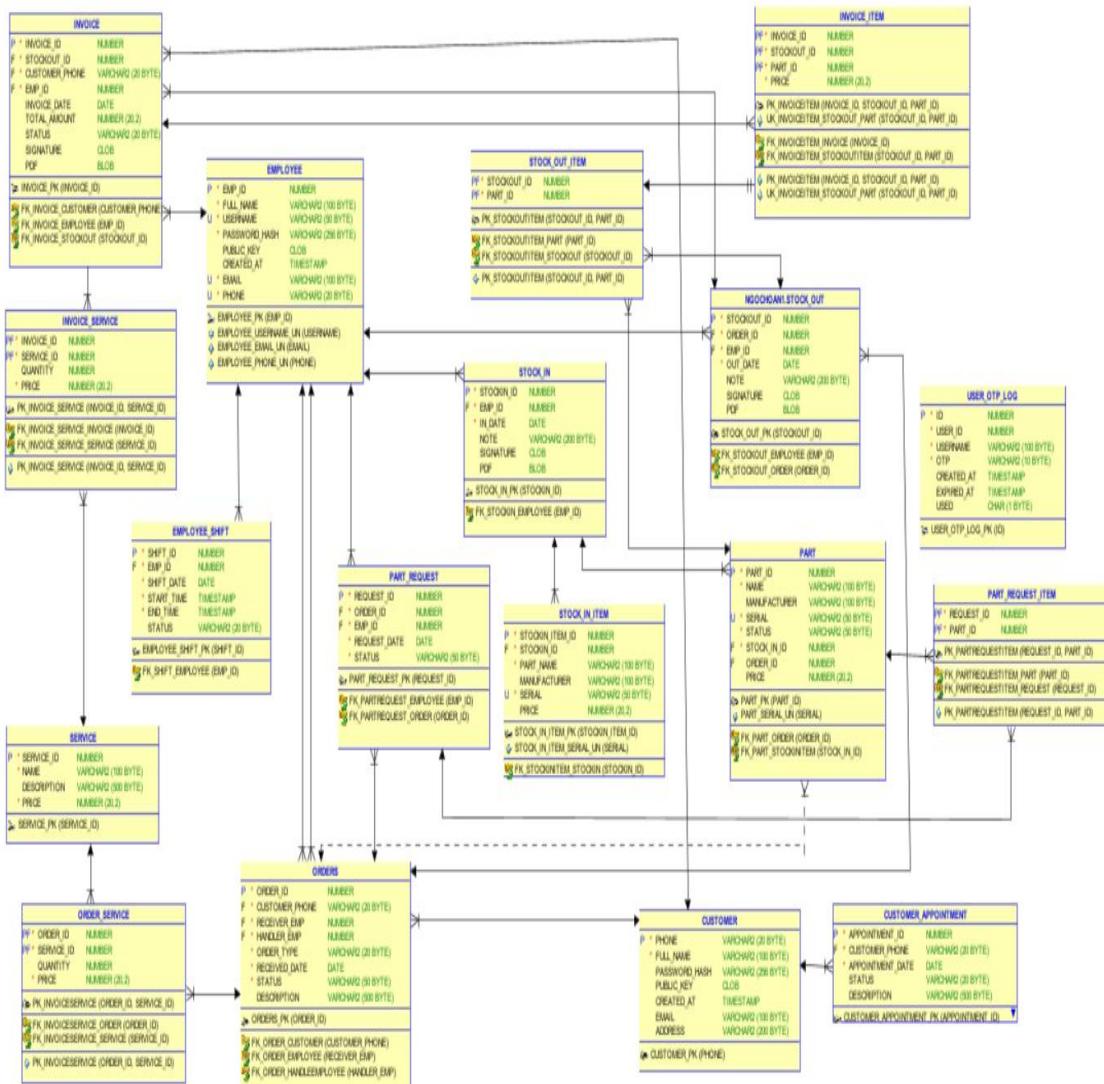
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1. Giới thiệu

Giai đoạn thiết kế hệ thống là bước chuyển đổi quan trọng từ yêu cầu và mô hình phân tích thành giải pháp kỹ thuật cụ thể. Đây là giai đoạn cụ thể hóa các yêu cầu trừu tượng thành kiến trúc hệ thống, cấu trúc dữ liệu và giao diện có thể triển khai được.

Tầm quan trọng của giai đoạn thiết kế thể hiện ở việc quyết định chất lượng, hiệu năng và khả năng mở rộng của hệ thống. Một thiết kế tốt đảm bảo tính ổn định, bảo mật và dễ bảo trì, đồng thời đóng vai trò cầu nối giữa đội phân tích và lập trình, giảm thiểu rủi ro trong quá trình phát triển.

3.2. Thiết kế cơ sở dữ liệu



Hình 3.1: Mô hình dữ liệu.

Bảng 3.1: Bảng nhân viên

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
EMP_ID	NUMBER (IDENTITY)	PK	Mã nhân viên, tự tăng.
FULL_NAME	VARCHAR2(100)	NOT NULL	Họ tên nhân viên.
USERNAME	VARCHAR2(50)	-	Tên đăng nhập duy nhất
PASSWORD_HASH	VARCHAR2(256)	-	Mật khẩu đã băm.
PUBLIC_KEY	CLOB	-	Public key RSA của nhân viên
CREATED_AT	TIMESTAMP	DEFAULT	Thời điểm tạo
EMAIL	VARCHAR2(100)	UNIQUE, NOT NULL	Email duy nhất của nhân viên
PHONE	VARCHAR2(20)	UNIQUE, NOT NULL	Số điện thoại duy nhất

Bảng 3.2: Bảng khách hàng.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
PHONE	VARCHAR2(20)	PK	Số điện thoại khách hàng
FULL_NAME	VARCHAR2(100)	NOT NULL	Họ tên khách hàng.
PASSWORD_HASH	VARCHAR2(256)	-	Mật khẩu đã băm

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
PUBLIC_KEY	CLOB	-	Public key RSA của khách hàng
CREATED_AT	TIMESTAMP	DEFAULT	Thời điểm tạo
EMAIL	VARCHAR2(100)	-	Email khách hàng
PHONE	VARCHAR2(20)	-	Địa chỉ khách hàng

Bảng 3.3: Bảng đặt hàng.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
ORDER_ID	NUMBER	PK	Mã đơn hàng
CUSTOMER_PHONE	VARCHAR2(20)	FK	Khách hàng tạo đơn
RECEIVER_EMP	NUMBER	FK	Nhân viên tiếp nhận đơn
HANDLER_EMP	NUMBER	FK	Nhân viên xử lý đơn
ORDER_TYPE	VARCHAR2(20)	NOT NULL	Loại đơn (sửa chữa, bảo trì...)
RECEIVED_DATE	DATE	NOT NULL	Ngày tiếp nhận đơn
STATUS	VARCHAR2(50)	NOT NULL	Trạng thái hiện tại của đơn hàng

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
DESCRIPTION	VARCHAR2(500)	-	Mô tả thêm về đơn

Bảng 3.4: Bảng nhập kho.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
STOCKIN_ID	NUMBER	PK	Mã phiếu nhập kho
EMP_ID	NUMBER	FK	Nhân viên nhập kho
IN_DATE	DATE	NOT NULL	Ngày nhập kho
NOTE	VARCHAR2(200)	-	Ghi chú
PDF	BLOB	-	File PDF của phiếu nhập

Bảng 3.5: Bảng chi tiết nhập kho.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
STOCKIN_ITEM_ID	NUMBER	PK	Mã chi tiết nhập kho
STOCKIN_ID	NUMBER	FK	Tham chiếu bảng
PART_NAME	VARCHAR2(100)	NOT NULL	Tên linh kiện nhập
MANUFACTURER	VARCHAR2(100)	-	Hãng sản xuất

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
SERIAL	VARCHAR2(50)	UNIQUE, NOT NULL	Số serial duy nhất
PRICE	NUMBER(20,2)	-	Giá linh kiện

Bảng 3.6: Bảng linh kiện.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
PART_ID	NUMBER	PK	Mã linh kiện
NAME	VARCHAR2(100)	NOT NULL	Tên linh kiện
MANUFACTURER	VARCHAR2(100)	-	Hãng sản xuất
SERIAL	VARCHAR2(50)	UNIQUE, NOT NULL	Serial duy nhất
STATUS	VARCHAR2(50)	NOT NULL	Trạng thái linh kiện (kho, đã xuất...)
STOCK_IN_ID	NUMBER	FK	Nhập kho từ phiếu nào
ORDER_ID	NUMBER	FK	Gắn với đơn hàng nào (nếu có)
PRICE	NUMBER(20,2)	-	Giá linh kiện

Bảng 3.7: Bảng xuất kho.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
STOCKOUT_ID	NUMBER	PK	Mã phiếu xuất kho

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
ORDER_ID	NUMBER	FK	Xuất cho đơn hàng nào
EMP_ID	NUMBER	FK	Nhân viên xuất kho
OUT_DATE	DATE	-	Ngày xuất kho
NOTE	VARCHAR2(200)	-	Ghi chú
PDF	BLOB	-	File PDF

Bảng 3.8: Bảng chi tiết xuất kho.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
STOCKOUT_ID	NUMBER	PK, FK	Phiếu xuất kho
PART_ID	NUMBER	PK, FK	Linh kiện được xuất

Bảng 3.9: Bảng yêu cầu linh kiện.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
REQUEST_ID	NUMBER	PK	Mã yêu cầu linh kiện
ORDER_ID	NUMBER	FK	Mã đơn hàng yêu cầu
EMP_ID	NUMBER	FK	Mã nhân viên yêu cầu
REQUEST_DATE	DATE	NOT NULL	Ngày yêu cầu

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
STATUS	VARCHAR2(50)	NOT NULL	Trạng thái yêu cầu

Bảng 3.10: Bảng chi tiết yêu cầu linh kiện.

T cột	Kiểu dữ liệu	Khóa	Diễn giải
REQUEST_ID	NUMBER	PK, FK	Mã yêu cầu
PART_ID	NUMBER	PK, FK	Mã linh kiện

Bảng 3.11: Bảng log OTP của người dùng.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
ID	NUMBER (IDENTITY)	PK	Mã log OTP
USER_ID	NUMBER	-	ID người nhận OTP
USERNAME	VARCHAR2(100)	-	Tài khoản sử dụng OTP
OTP	VARCHAR2(10)	NOT NULL	Mã OTP
CREATED_AT	TIMESTAMP	DEFAULT	Thời điểm tạo OTP
EXPIRED_AT	TIMESTAMP	-	Hết hạn
USED	CHAR(1)	-	Đã dùng hay chưa

Bảng 3.12: Bảng ca trực.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
SHIFT_ID	NUMBER	PK	Mã ca làm việc
EMP_ID	NUMBER	FK	Nhân viên
SHIFT_DATE	DATE	NOT NULL	Ngày làm
START_TIME	TIMESTAMP	NOT NULL	Bắt đầu
END_TIME	TIMESTAMP	NOT NULL	Kết thúc
STATUS	VARCHAR2(20)	-	Trạng thái ca

Bảng 3.13: Bảng đặt lịch hẹn

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
APPOINTMENT_ID	NUMBER	PK	Mã lịch hẹn
CUSTOMER_PHONE	VARCHAR2(20)	FK	Khách hàng
APPOINTMENT_DATE	DATE	NOT NULL	Ngày hẹn
STATUS	VARCHAR2(20)	-	Trạng thái lịch hẹn
DESCRIPTION	VARCHAR2(500)	-	Mô tả

Bảng 3.14: Bảng hóa đơn

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
INVOICE_ID	NUMBER	PK	Mã hóa đơn
STOCKOUT_ID	NUMBER	FK	Mã xuất kho
CUSTOMER_PHONE	VARCHAR2(20)	FK	Số điện thoại khách hàng
EMP_ID	NUMBER	-	Mã nhân viên tạo hóa đơn
INVOICE_DATE	DATE	-	Ngày tạo hóa đơn
TOTAL_AMOUNT	NUMBER(20,2)	-	Tổng tiền hóa đơn
STATUS	VARCHAR2(20)	-	Trạng thái (pending, paid...)
PDF	BLOB	-	File hóa đơn PDF

Bảng 3.15: Bảng chi tiết hóa đơn.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
INVOICE_ID	NUMBER	PK, FK	Mã hóa đơn
STOCKOUT_ID	NUMBER	PK, FK	Mã xuất kho
PART_ID	NUMBER	PK, FK	Linh kiện
PRICE	NUMBER(20,2)	-	Giá bán của linh kiện

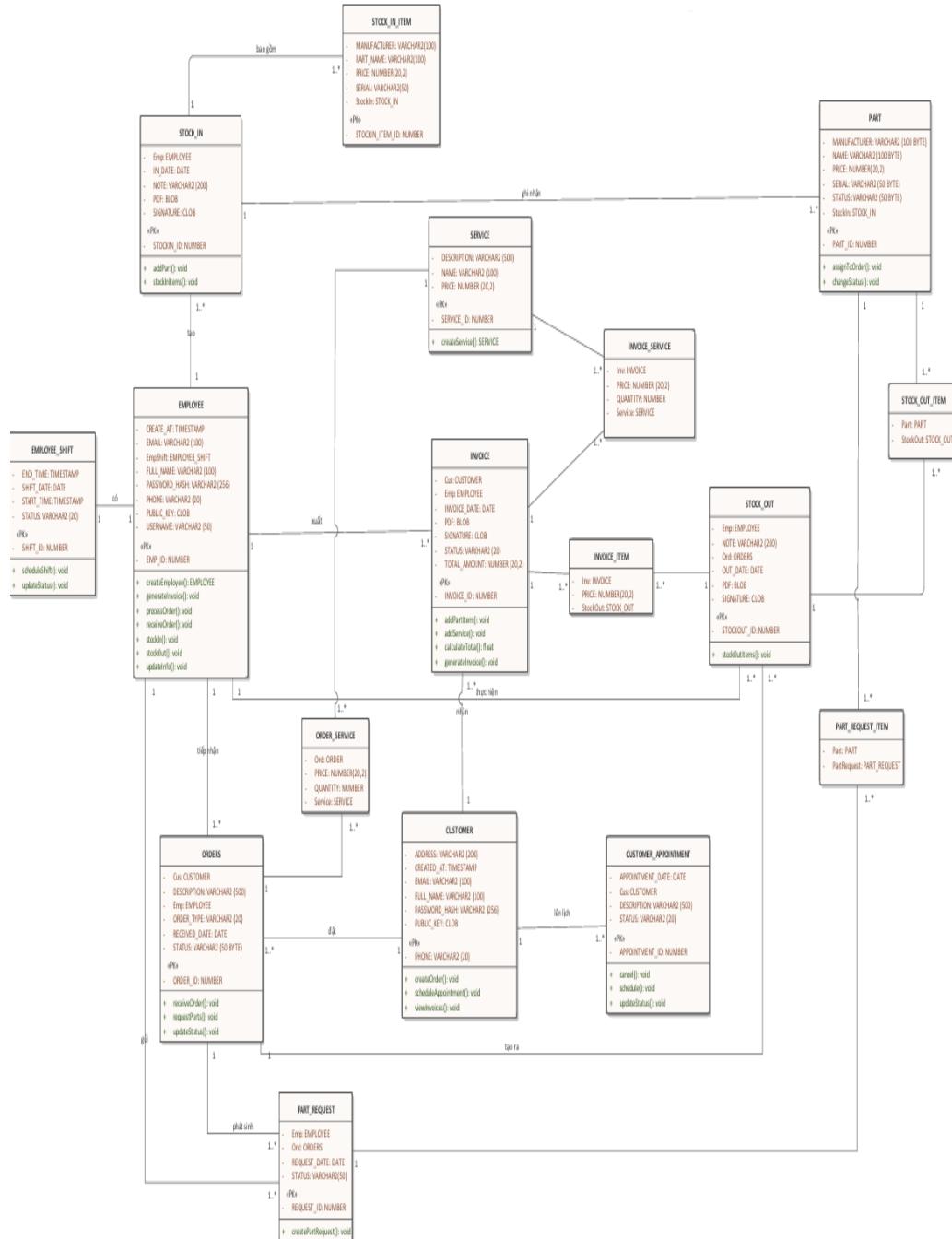
Bảng 3.16: Bảng dịch vụ.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
SERVICE_ID	NUMBER	PK	Mã dịch vụ
NAME	VARCHAR2(100)	-	Tên dịch vụ
DESCRIPTION	VARCHAR2(500)	-	Mô tả dịch vụ
PRICE	NUMBER(20,2)	-	Giá dịch vụ

Bảng 3.17: Bảng dịch vụ của đơn hàng.

Tên cột	Kiểu dữ liệu	Khóa	Diễn giải
ORDER_ID	NUMBER	PK, FK	Đơn hàng
SERVICE_ID	VARCHAR2(100)	PK, FK	Mã dịch vụ
QUANTITY	VARCHAR2(500)	-	Số lượng
PRICE	NUMBER(20,2)	-	Đơn giá dịch vụ

3.3. Sơ đồ lớp ở mức thiết kế

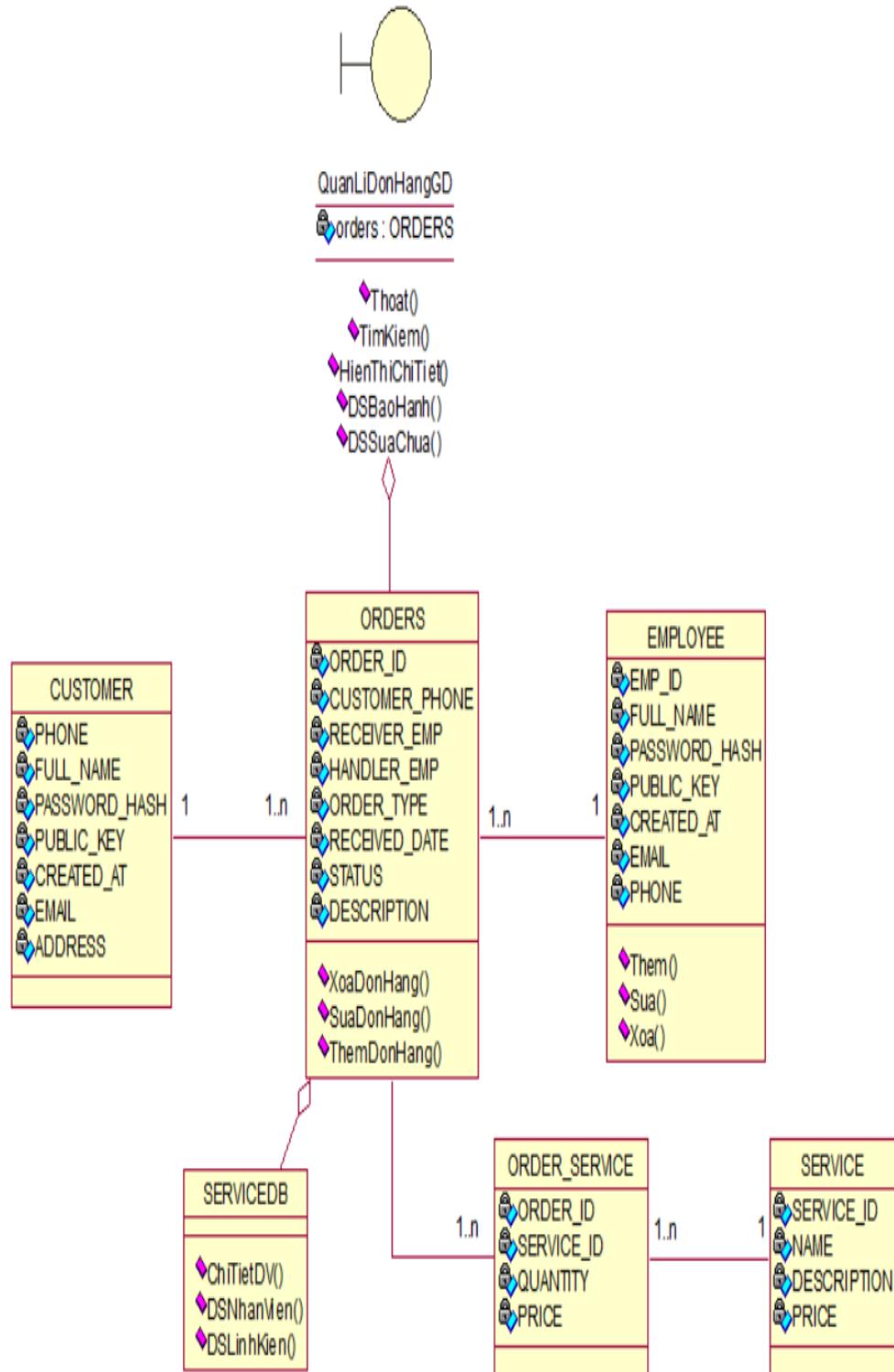


Hình 3.2: Sơ đồ lớp ở mức thiết kế.

3.4. Thiết kế chức năng hệ thống

3.4.1. Chức năng quản lý đơn hàng

Xây dựng sơ đồ lớp ở mức thiết kế theo mô hình 3 lớp:



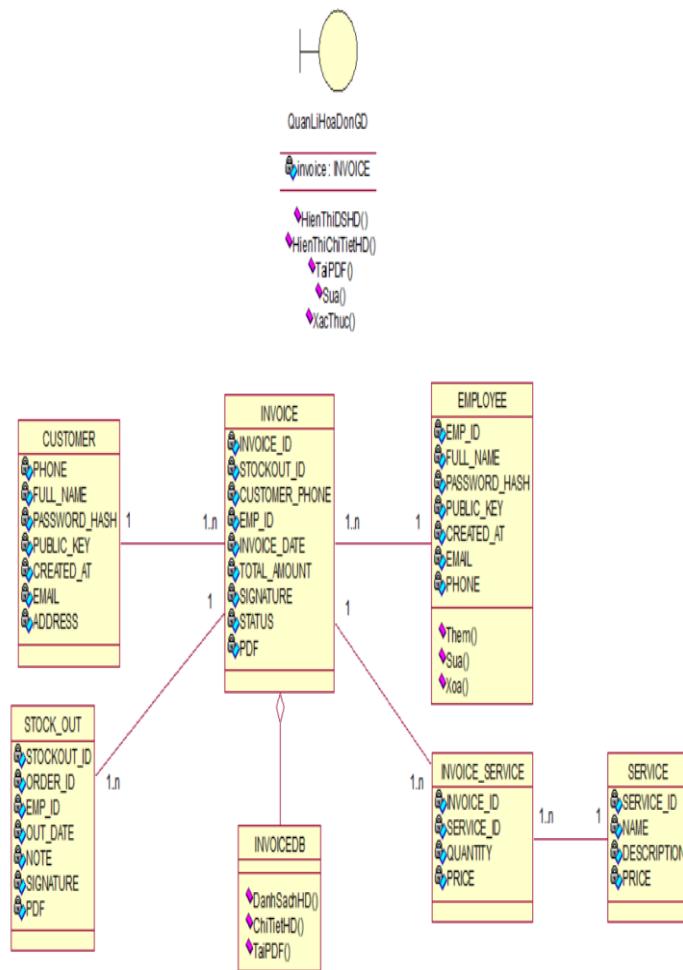
Hình 3.3: Sơ đồ lớp ở mức thiết kế của quản lý đơn hàng.

Mô hình ba lớp này mô tả chức năng quản lý đơn hàng, xoay quanh thực thể trung tâm ORDERS. Thực thể ORDERS này liên kết với CUSTOMER (Khách hàng) và EMPLOYEE (Nhân viên xử lý) theo mối quan hệ một nhiều. Chi tiết từng mặt hàng được quản lý thông qua ORDER_SERVICE, liên kết đơn hàng với các

dịch vụ cụ thể (SERVICE) cùng với số lượng và giá. Mô hình bao gồm các thuộc tính dữ liệu quan trọng như ID, trạng thái, thông tin cá nhân (tên, số điện thoại) và các phương thức nghiệp vụ cơ bản để quản lý các thực thể này, bao gồm thêm, sửa, xóa và tìm kiếm đơn hàng, tạo nên nền tảng dữ liệu và logic cho toàn bộ hệ thống.

3.4.2. Chức năng quản lý hóa đơn

Xây dựng sơ đồ lớp ở mức thiết kế theo mô hình 3 lớp:



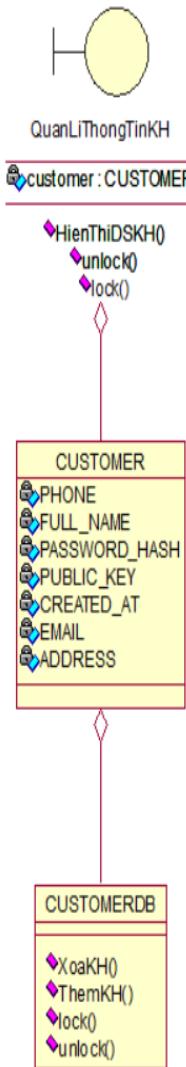
Hình 3.4: Sơ đồ lớp ở mức thiết kế của quản lý hóa đơn.

Mô hình ba lớp này định nghĩa cấu trúc dữ liệu cho chức năng quản lý thông tin hóa đơn, xoay quanh thực thể trung tâm `INVOICE`. Thực thể `INVOICE` này liên kết theo mối quan hệ một nhiều với `CUSTOMER` (Khách hàng đặt hàng), `EMPLOYEE` (Nhân viên lập hóa đơn) và `STOCK_OUT` (Phiếu xuất kho liên quan). Chi tiết các mặt hàng trong hóa đơn được quản lý qua bảng liên kết `INVOICE_SERVICE`, đại diện cho mối quan hệ nhiều nhiều với `SERVICE` (Dịch vụ/Mặt hàng). Toàn bộ hệ thống này cung cấp nền tảng dữ liệu và các thao tác

nghiệp vụ quan trọng (như tạo PDF, xác thực, hiển thị chi tiết) để theo dõi và quản lý chu trình hóa đơn hoàn chỉnh.

3.4.3. Chức năng quản lý thông tin khách hàng

Xây dựng sơ đồ lớp ở mức thiết kế theo mô hình 3 lớp:



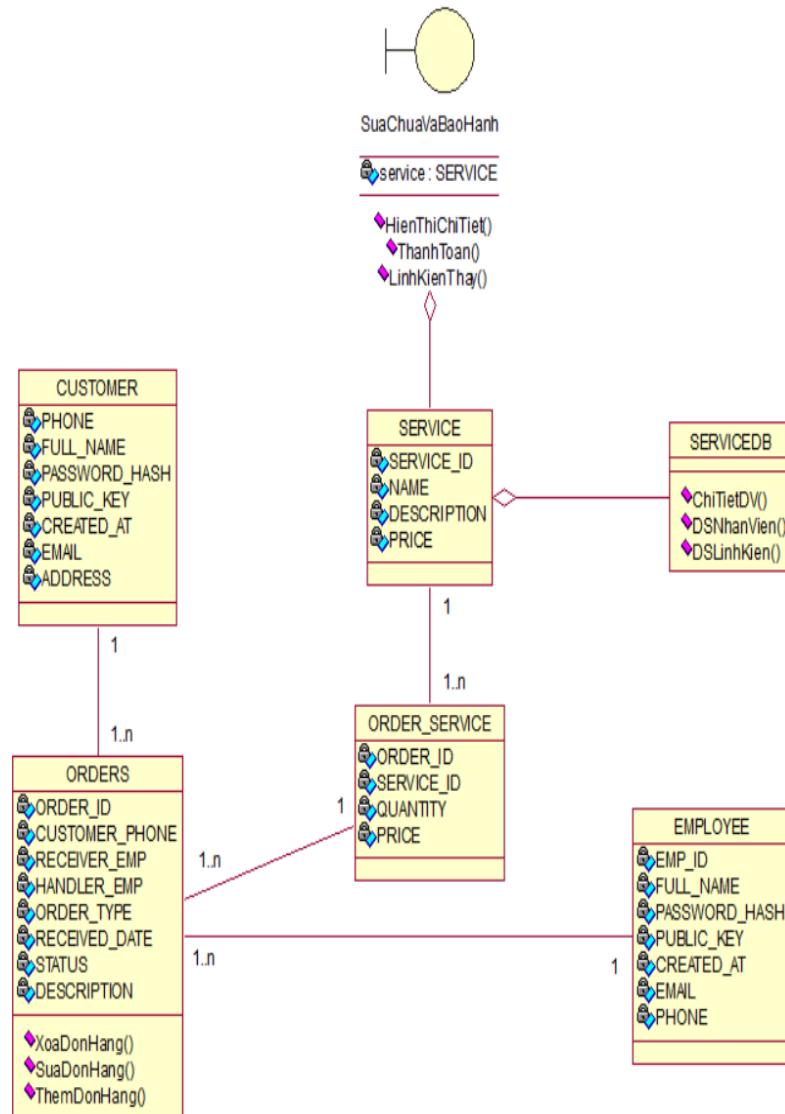
Hình 3.5: Sơ đồ lớp ở mức thiết kế của quản lý thông tin khách hàng.

Mô hình ba lớp định nghĩa cấu trúc dữ liệu cho chức năng quản lý thông tin người dùng, xoay quanh thực thể trung tâm `CUSTOMER`. Thực thể `CUSTOMER` lưu trữ toàn bộ thông tin cá nhân của người dùng, bao gồm các thuộc tính như tên đầy đủ, số điện thoại, địa chỉ, email, và các dữ liệu bảo mật quan trọng như mật khẩu đã được băm và khóa công khai. Lớp điều khiển `QuanLiThongTinKH` tương tác với thực thể này để thực hiện các chức năng nghiệp vụ cấp cao như hiển thị

danh sách khách hàng (HienThiDSKH()), và các thao tác quản lý trạng thái tài khoản như khóa (lock()) và mở khóa (unlock()).

3.4.4. Chức năng sửa chữa và bảo hành

Xây dựng sơ đồ lớp ở mức thiết kế theo mô hình 3 lớp:



Hình 3.6: Sơ đồ lớp ở mức thiết kế của sửa chữa và bảo hành.

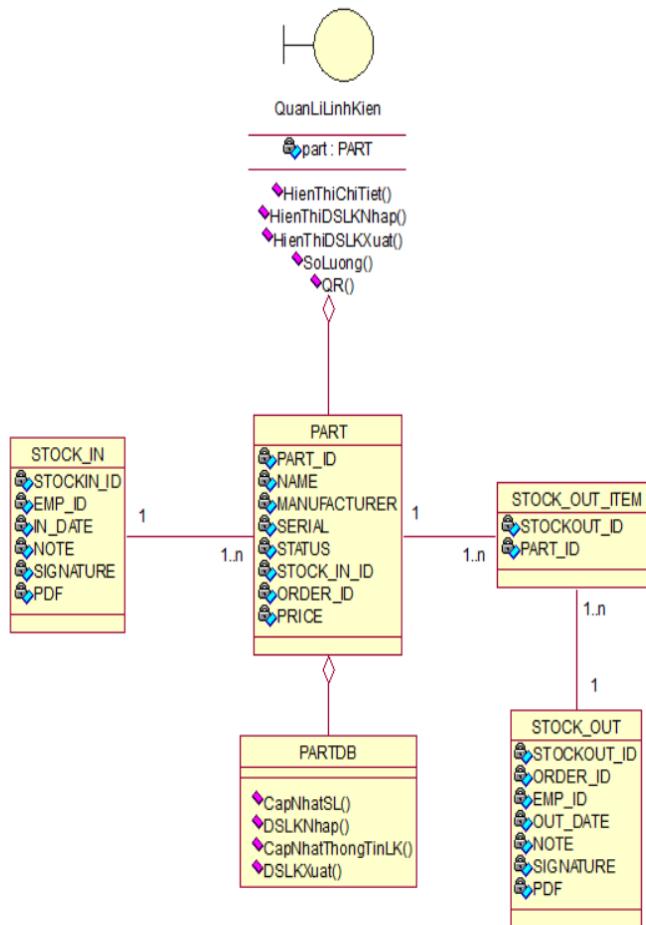
Mô hình ba lớp này định nghĩa cấu trúc dữ liệu cho chức năng quản lý sửa chữa và bảo hành, xoay quanh thực thể trung tâm là SERVICE (dịch vụ).

Thực thể SERVICE chứa các thuộc tính như tên, mô tả và giá, và liên kết với lớp SERVICEDB để thực hiện các thao tác truy cập dữ liệu như lấy chi tiết dịch vụ (ChiTietDV()) và danh sách linh kiện (DLSinhKien()). Hoạt động của dịch vụ được thể hiện thông qua thực thể ORDERS (Đơn hàng/Phiếu sửa chữa),

nơi dịch vụ được đưa vào qua bảng trung gian ORDER_SERVICE cùng với số lượng và giá cụ thể. Các đơn hàng này liên kết với thông tin của CUSTOMER (Khách hàng) và EMPLOYEE (Nhân viên). Lớp điều khiển SuaChuaVaBaoHanh tập trung vào các chức năng nghiệp vụ liên quan đến dịch vụ như hiển thị chi tiết(HienThiChiTiet()), thanh toán (ThanhToan()), và quản lý linh kiện thay thế (LinhKienThay()), tạo nên nền tảng logic và dữ liệu chi tiết cho toàn bộ quy trình cung cấp dịch vụ và bảo hành.

3.4.5. Chức năng quản lý linh kiện

Xây dựng sơ đồ lớp ở mức thiết kế theo mô hình 3 lớp:



Hình 3.7: Sơ đồ lớp ở mức thiết kế của quản lý linh kiện.

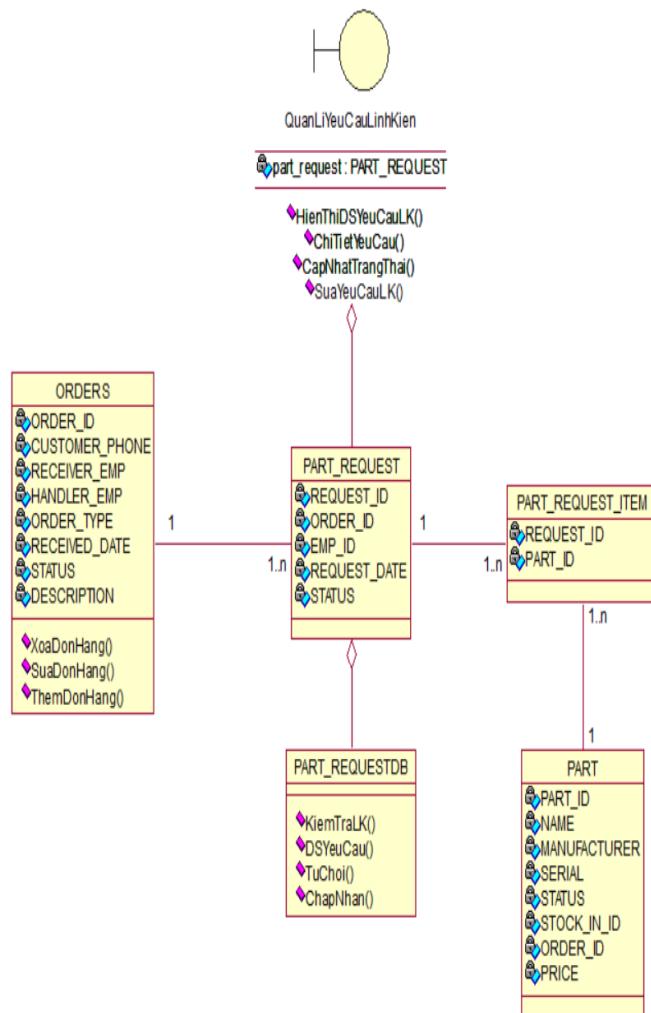
Mô hình ba lớp này định nghĩa cấu trúc dữ liệu cho chức năng quản lý linh kiện, xoay quanh thực thể trung tâm là PART (linh kiện). Thực thể PART lưu trữ các thuộc tính dữ liệu quan trọng như mã linh kiện, tên, nhà sản xuất, số serial, trạng thái và giá. PART liên kết với STOCK_IN (Phiếu nhập kho) theo mối quan hệ một nhiều, cho biết linh kiện được nhập từ phiếu nào. Đồng thời, nó liên kết với

STOCK_OUT_ITEM (Chi tiết phiếu xuất kho), một bảng trung gian giữa PART và STOCK_OUT (Phiếu xuất kho), để quản lý các linh kiện được xuất ra.

Lớp điều khiển QuanLiLinhKien và lớp truy cập dữ liệu PARTDB cung cấp các phương thức nghiệp vụ cơ bản, từ cập nhật số lượng (CapNhatSL()), lấy danh sách nhập/xuất (DSLKNhap(), DSLKXuat()), đến tính toán số lượng (SoLuong()).

3.4.6. Chức năng quản lý yêu cầu linh kiện

Xây dựng sơ đồ lớp ở mức thiết kế theo mô hình 3 lớp:



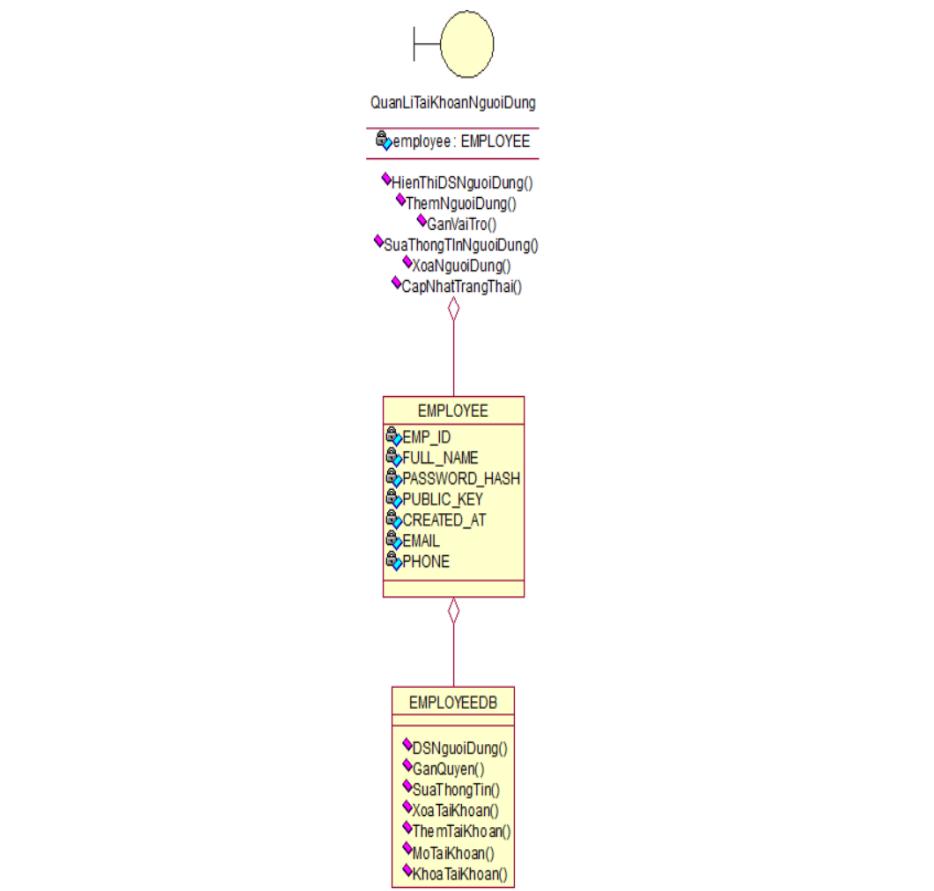
Hình 3.8: Sơ đồ lớp ở mức thiết kế của quản lý yêu cầu linh kiện.

Mô hình ba lớp này định nghĩa cấu trúc dữ liệu cho chức năng quản lý yêu cầu linh kiện, xoay quanh thực thể trung tâm là PART_REQUEST (Yêu cầu Linh kiện). Thực thể PART_REQUEST lưu trữ các thuộc tính dữ liệu quan trọng như mã yêu cầu, ngày yêu cầu, mã hóa đơn và trạng thái. PART_REQUEST liên kết với thực thể ORDERS (Đơn hàng) theo mối quan hệ một nhiều, cho biết phiếu yêu

cầu linh kiện này được tạo ra cho đơn hàng nào. Chi tiết từng linh kiện được yêu cầu được quản lý thông qua bảng trung gian PART_REQUEST_ITEM, liên kết phiếu yêu cầu với các linh kiện cụ thể (PART) theo mối quan hệ một nhiều.

Lớp điều khiển QuanLiYeuCauLinhKien cung cấp các phương thức nghiệp vụ như hiển thị danh sách yêu cầu (HienThiDSYeuCauLK()) và cập nhật trạng thái (CapNhatTrangThai()), trong khi lớp truy cập dữ liệu PART_REQUESTDB thực hiện các thao tác quản lý dữ liệu chuyên biệt như kiểm tra tồn kho (KiemTraLK()) và xử lý phê duyệt (ChapNhan(), TuChoi()). Chức năng quản lý tài khoản người dùng

Xây dựng sơ đồ lớp ở mức thiết kế theo mô hình 3 lớp:



Hình 3.9: Sơ đồ lớp ở mức thiết kế của quản lý tài khoản người dùng.

Mô hình ba lớp này định nghĩa cấu trúc dữ liệu cho chức năng quản lý tài khoản người dùng, xoay quanh thực thể trung tâm là EMPLOYEE (Nhân viên).

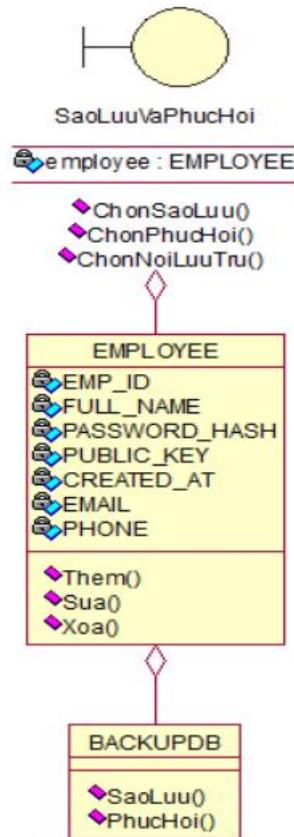
Thực thể EMPLOYEE lưu trữ các thuộc tính dữ liệu quan trọng như mã nhân viên (EMP_ID), tên đầy đủ (FULL_NAME), email, số điện thoại, và các dữ liệu bảo mật như mật khẩu đã được băm (PASSWORD_HASH) và khóa công khai

(PUBLIC_KEY). EMPLOYEE liên kết với lớp EMPLOYEEEDB, lớp này đảm nhận vai trò truy cập dữ liệu trực tiếp, cung cấp các phương thức cơ bản như thêm (ThemTaiKhoan()), xóa (XoaTaiKhoan()), sửa thông tin (SuaThongTin()), và các thao tác quản lý trạng thái bảo mật của tài khoản (KhoaTaiKhoan() và MoTaiKhoan()).

Lớp điều khiển QuanLyTaiKhoanNguoiDung thực hiện các chức năng nghiệp vụ cấp cao như hiển thị danh sách người dùng (HienThiDSNguoiDung()), gán vai trò (GanVaiTro()), và cập nhật trạng thái (CapNhatTrangThai()). Giao diện ứng dụng

3.4.7. Chức năng sao lưu và phục hồi

Xây dựng sơ đồ lớp ở mức thiết kế theo mô hình 3 lớp:

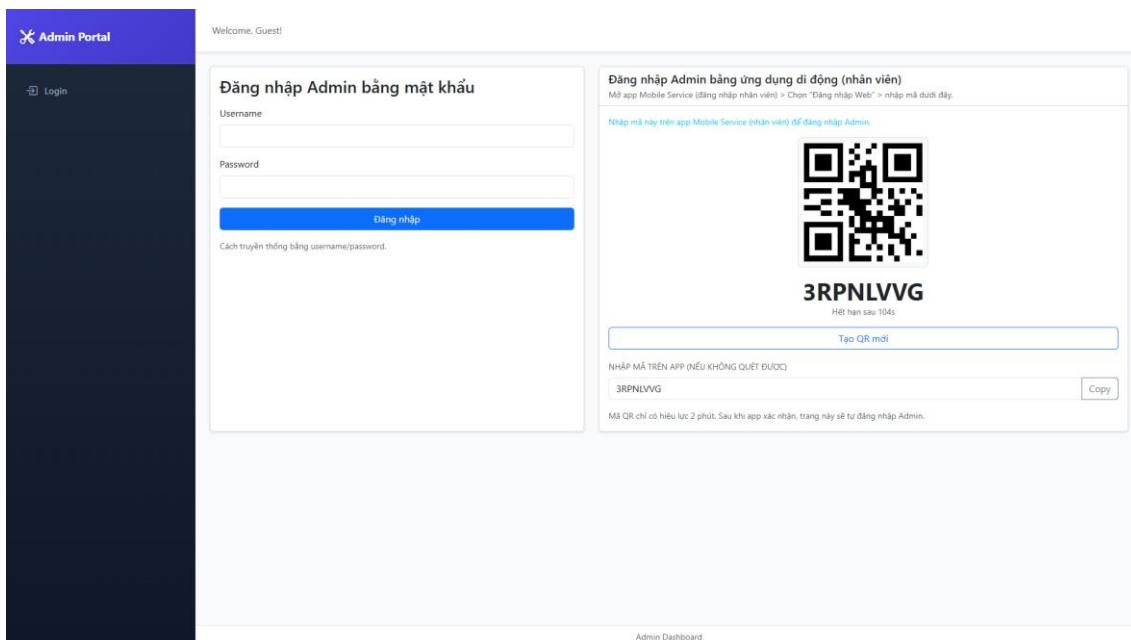


Hình 3.10: Sơ đồ lớp ở mức thiết kế của sao lưu và phục hồi.

Mô hình ba lớp này định nghĩa cấu trúc dữ liệu cho chức năng sao lưu và phục hồi, xoay quanh thực thể EMPLOYEE (Nhân viên) và lớp Truy cập Dữ liệu BACKUPDB. Thực thể EMPLOYEE lưu trữ thông tin của người dùng thực hiện thao tác (bao gồm ID, tên đầy đủ, mật khẩu được băm, khóa công khai, email và số điện thoại) và có các phương thức cơ bản để quản lý hồ sơ nhân viên (Them(), Sua(), Xoa()). EMPLOYEE liên kết với lớp điều khiển SaoLuuVaPhucHoi, nơi thực hiện các chức năng nghiệp vụ cấp cao như chọn loại sao lưu (ChonSaoLuu()), chọn phục hồi (ChonPhucHoi()), và chọn nơi lưu trữ (ChonNoiLuuTru()). Lớp BACKUPDB đóng vai trò là Lớp Dữ liệu chuyên biệt, cung cấp các phương thức cốt lõi để thực hiện việc sao lưu (SaoLuu()) và phục hồi (PhucHoi()) dữ liệu.

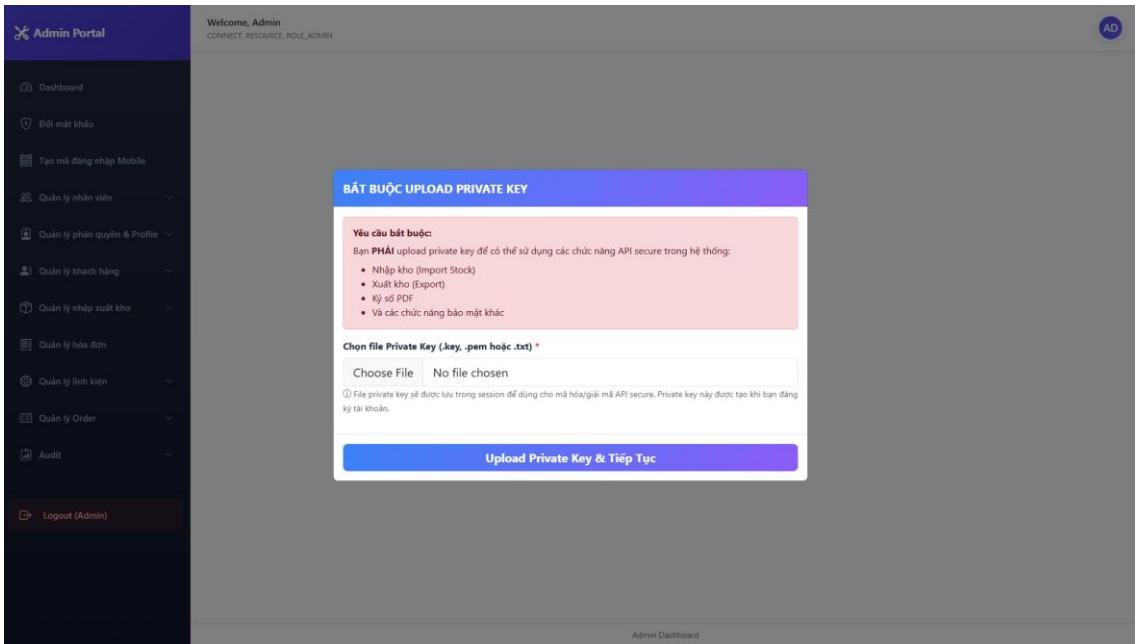
3.4.8. Giao diện website

3.4.8.1 Giao diện nhân viên



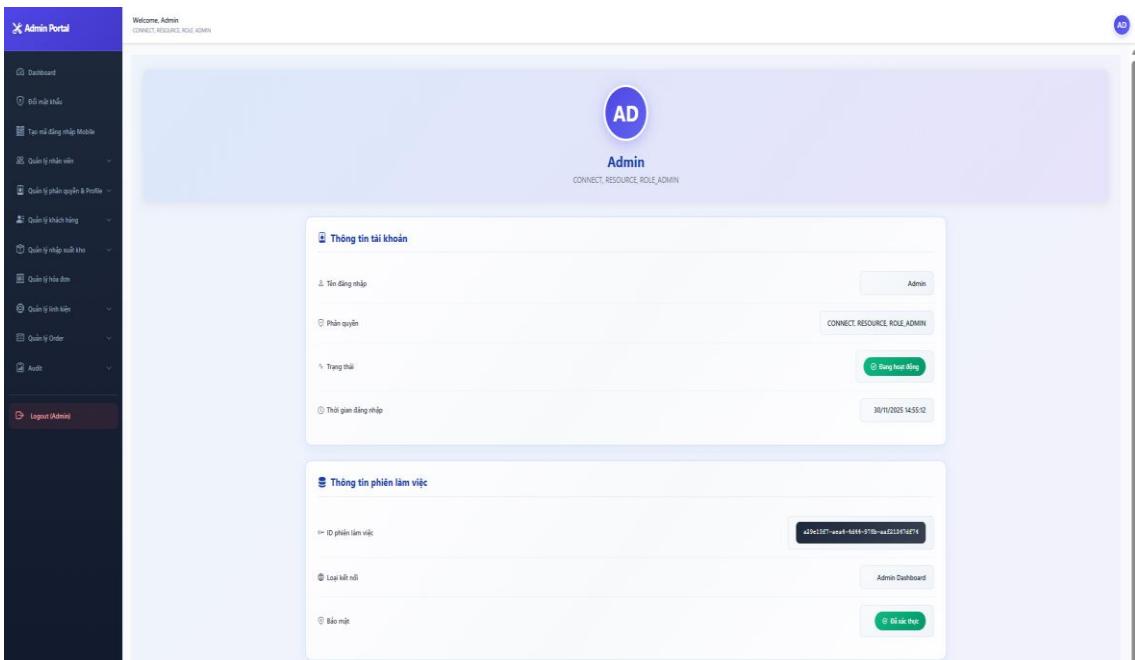
Hình 3.11: Giao diện chức năng đăng nhập Nhân viên.

Ở giao diện đăng nhập nhân viên thì nhân viên có thể lựa chọn nhập tên đăng nhập kèm mật khẩu hoặc đăng nhập vào ứng dụng trên điện thoại và quét QR để đăng nhập trên website.



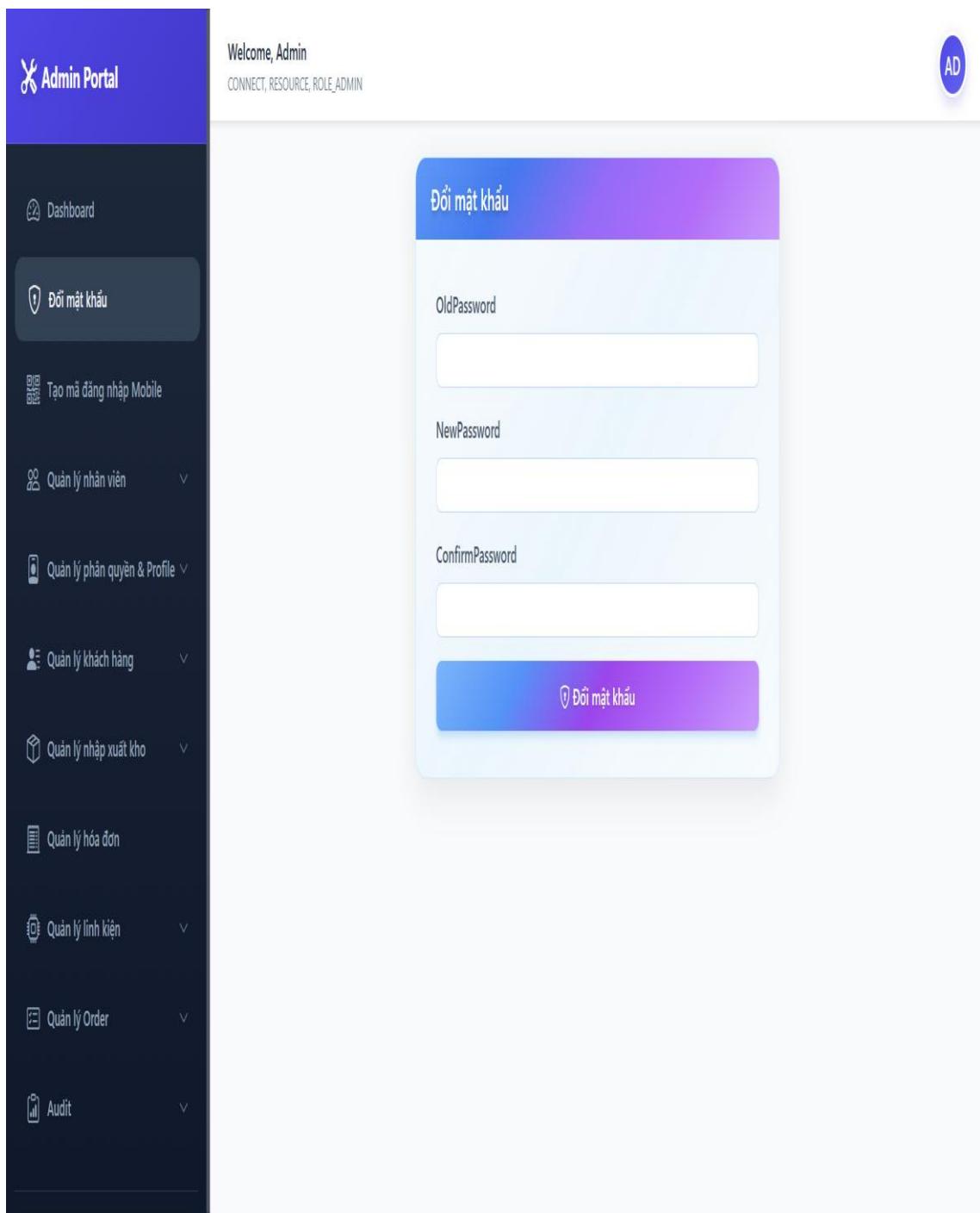
Hình 3.12: Giao diện chức năng tải Private key nhân viên.

Khi thực hiện đăng nhập xong thì một bảng thông báo yêu cầu tải khóa bí mật từ thiết bị của nhân viên để xác thực.



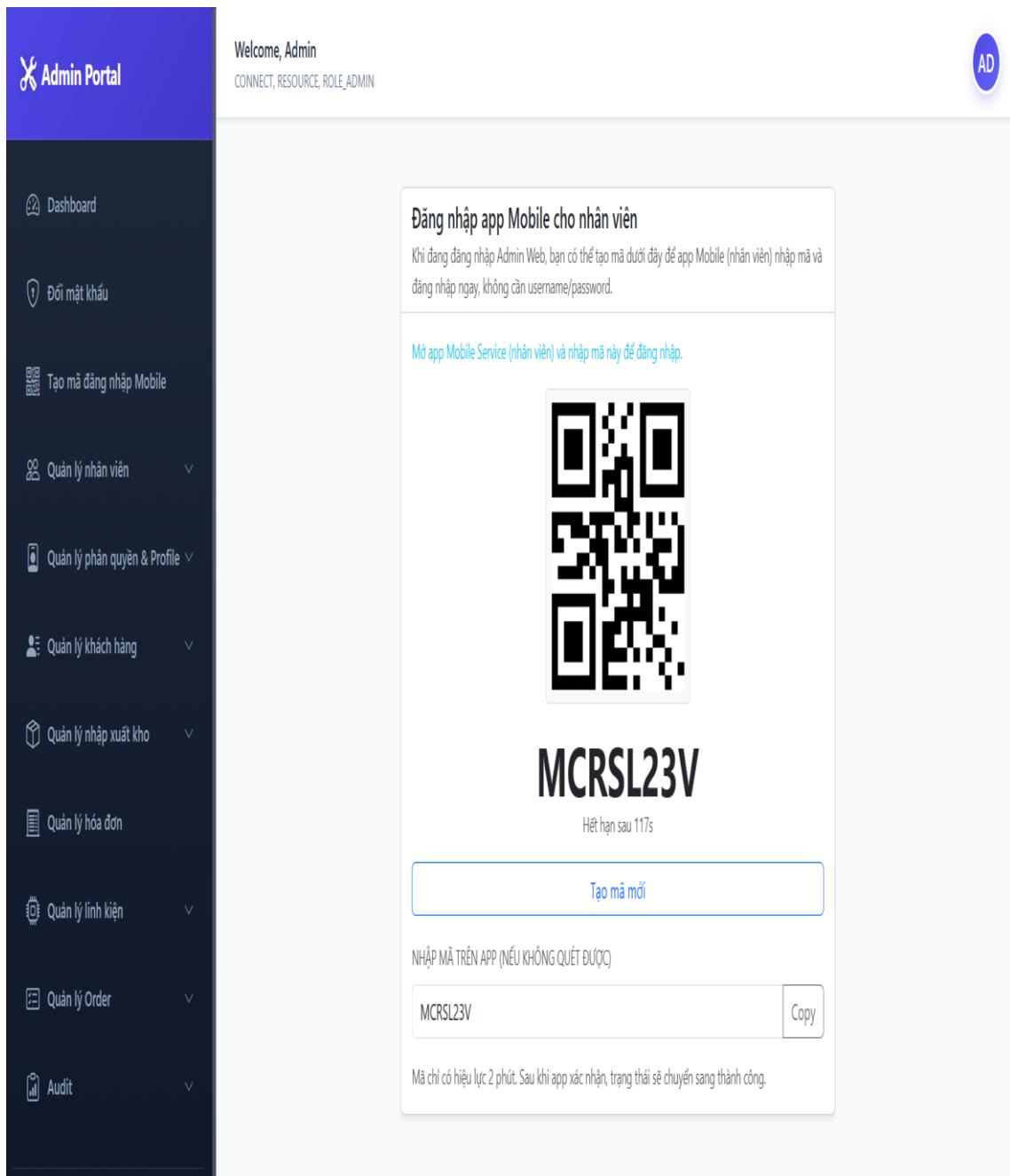
Hình 3.13: Giao diện chức năng xem thông tin của tài khoản nhân viên.

Khi vào giao diện trang chủ của nhân viên thì sẽ có được các thông tin cơ bản của nhân viên đang đăng nhập.



Hình 3.14: Giao diện chức năng đổi mật khẩu cho nhân viên.

Nhân viên có thể thay đổi mật khẩu thường xuyên để cho tài khoản bảo mật hơn.



Hình 3.15: Giao diện chức năng đăng nhập bằng QR cho nhân viên.

Khi nhân viên muốn đăng nhập trên điện thoại một cách nhanh chóng thì trên website có chức năng quét mã QR để đăng nhập, nhân viên thực hiện mở ứng dụng trên điện thoại và thực hiện quét để đăng nhập

Welcome, Admin
CONNECT, RESOURCE, ROLE_ADMIN

Danh sách nhân viên

Thêm mới

Tên nhân viên Username -- Tất cả vai trò -- -- Tất cả trạng thái --

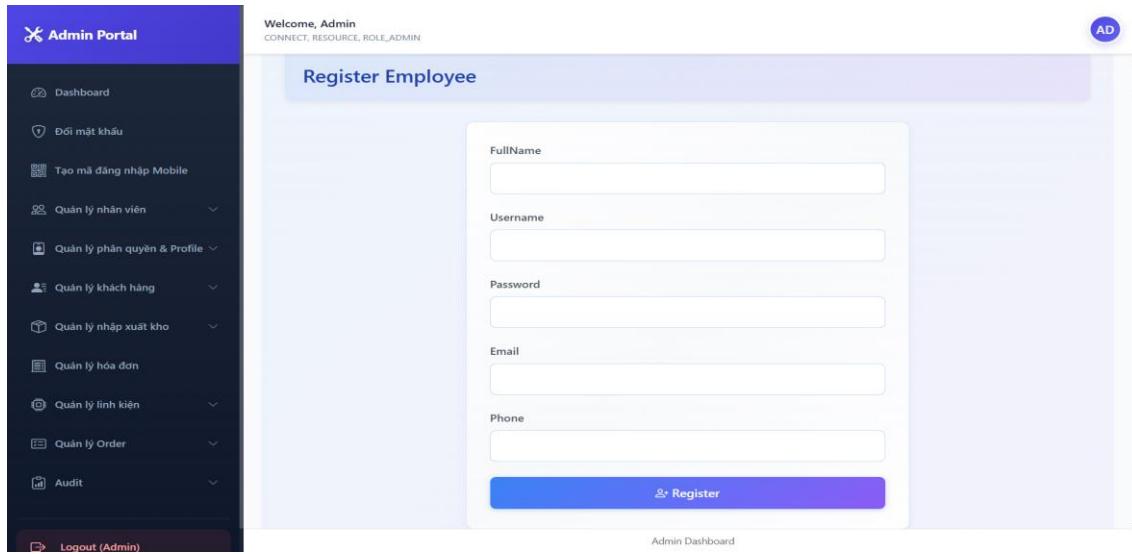
Tim kiếm Đặt lại

#	Full Name	Username	Status	Email	Phone	Role	Actions
1	Admin	Admin	OPEN	Admin@example.com	0987654321	CONNECT, RESOURCE, ROLE_ADMIN	<button>Lock</button>
2	THUKHO	Thukho	OPEN	Thukho@gmail.com	03332917468	CONNECT, RESOURCE, ROLE_THUKHO	<button>Lock</button>
3	Kỹ Thuật Viên	Kithuatvien	OPEN	Kithuatvien@gmail.com	123476898	CONNECT, RESOURCE, ROLE_KITHUATVIEN	<button>Lock</button>
4	Tiếp tân	Tieptan	OPEN	Tieptan@gmail.com	123456789	CONNECT, RESOURCE, ROLE_TIEPTAN	<button>Lock</button>
21	KTV02	Ktv02	OPEN	Ktv02@gmail.com	03328802146	CONNECT, RESOURCE	<button>Lock</button>
22	KTV03	Ktv03	OPEN	Ktv03@gmail.com	0332779313	CONNECT, RESOURCE	<button>Lock</button>

Admin Dashboard

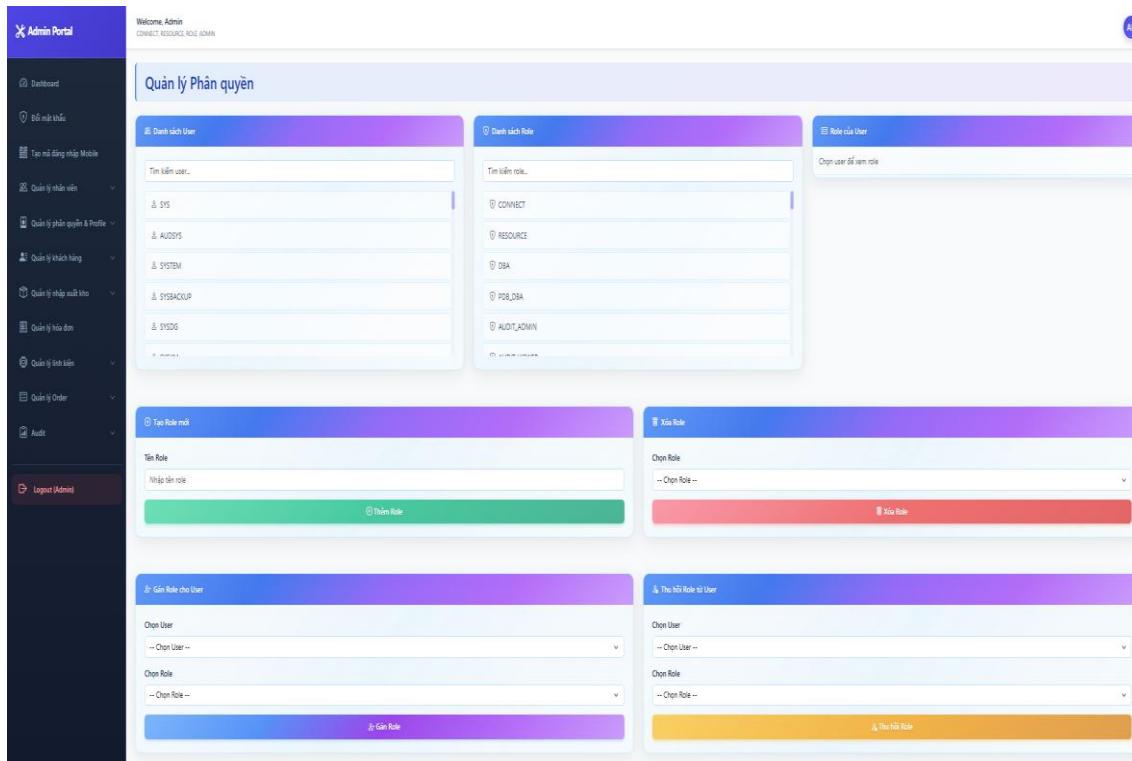
Hình 3.16: Giao diện chức năng đăng nhập bằng QR cho nhân viên.

Chức năng quản lý nhân viên cho phép quản trị viên thực hiện tìm kiếm theo tên, tên đăng nhập. Có thể lọc theo vai trò, trạng thái và khóa tài khoản của nhân viên nếu cần thiết. Ngoài ra thì có chức năng thêm mới nhân viên khi có nhân viên mới vào làm.



Hình 3.17: Giao diện chức năng thêm nhân viên mới.

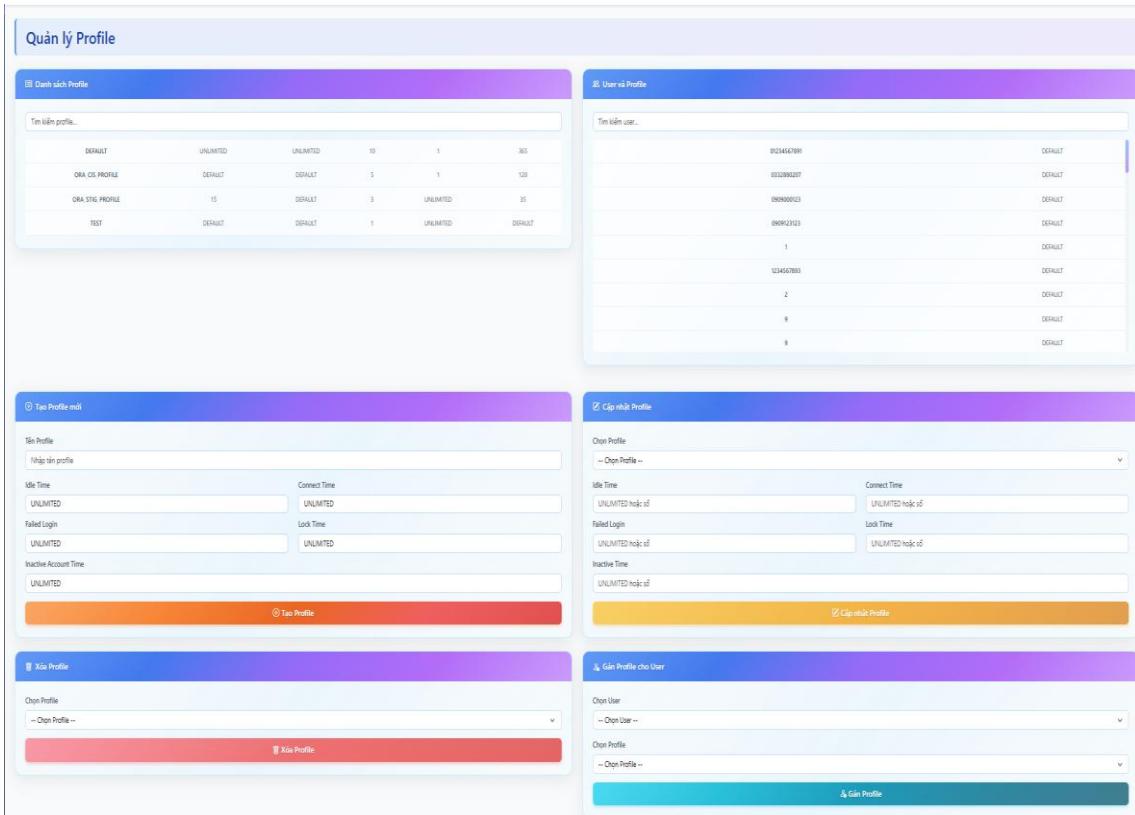
Khi quản trị viên thực hiện chức năng thêm nhân viên mới thì một trang tạo nhân viên mới sẽ xuất hiện và yêu cầu phải nhập đầy đủ các thông tin cần thiết như tên đầy đủ, tên đăng nhập, mật khẩu, email và số điện thoại.



Hình 3.18: Giao diện chức năng quản lý phân quyền của Admin

Chức năng quản lý phân quyền sẽ cho quản trị viên kiểm tra các quyền của một tài khoản người dùng đang có. Còn có các chức năng như tạo một vai trò mới, quản trị viên sẽ nhập tên vai trò mới và nhấn “Thêm vai trò”, có thể nhập tên vai

trò và thực hiện xóa thông qua chức năng “Xóa vai trò”. Ngoài ra thì quản trị viên có thể chọn người dùng và chọn vai trò để gán hoặc thu hồi vai trò thông qua chức năng “Gán vai trò cho người dùng” và “Thu hồi vai trò từ người dùng”.



Hình 3.19: Giao diện chức năng quản lý profile.

Chức năng này cho phép quản trị viên thiết lập các chính sách giới hạn chặt chẽ đối với từng nhóm người dùng. Các nghiệp vụ quản lý Profile bao gồm: Tạo Profile mới, nơi quản trị viên định nghĩa các giới hạn như Idle Time (thời gian nhàn rỗi tối đa), Connect Time (tổng thời gian kết nối), Failed Login (số lần đăng nhập thất bại tối đa), và Lock Time (thời gian tài khoản bị khóa); Cập nhật Profile để điều chỉnh các giới hạn này; và Xóa Profile để loại bỏ các cấu hình không còn sử dụng. Quan trọng nhất, hệ thống cho phép quản trị viên Gán Profile cho User để áp dụng chính sách giới hạn tài nguyên tương ứng cho tài khoản người dùng đã

chọn.

Phone	Full Name	Status	Email	Role	Actions
01234567891	Cuong	OPEN	c@gmail.com	ROLE_KHACHHANG	<button>Lock</button>
0332880207	Khách hàng 1	OPEN	vinhdtq123123123@gmail.com	ROLE_KHACHHANG	<button>Lock</button>
0909000123	Nguyen Van A	OPEN	a@gmail.com	ROLE_KHACHHANG	<button>Lock</button>
0909123123	Test User	OPEN	test@example.com	ROLE_KHACHHANG	<button>Lock</button>
1	Dinh	OPEN	Dinh@gmail.com	ROLE_KHACHHANG	<button>Lock</button>
2	2	OPEN	2@gmail.com	ROLE_KHACHHANG	<button>Lock</button>

Hình 3.20: Giao diện danh sách khách hàng.

Chức năng quản lí danh sách khách hàng cho phép quản trị viên thực hiện tìm kiếm theo tên khách hàng, tên khách hàng hoặc lọc theo trạng thái tài khoản. Ngoài ra thì quản trị viên có thể khóa và mở khóa tài khoản nếu khách hàng vi phạm điều khoản của cửa hàng.

ID	Số điện thoại	Ngày hẹn	Trạng thái	Mô tả
16	01234567891	2025-12-03 00:00	Đã lên lịch	Bao hành
11	1	2025-11-29 00:00	Đã lên lịch	A
15	0909000123	2025-11-29 00:00	Đã lên lịch	Hen bao hanh
14	1	2025-11-28 00:00	Đã lên lịch	Alo
12	1	2025-11-28 00:00	Đã lên lịch	ABC
13	1	2025-11-27 00:00	Đã lên lịch	ABC
10	0332880207	2025-11-15 00:00	Đã lên lịch	abc

Hình 3.21: Giao diện danh sách đặt lịch hẹn.

Chức năng quản lí danh sách đặt lịch cho phép quản trị viên chỉ coi, thực hiện tìm kiếm theo số điện thoại, ngày tháng hoặc lọc theo trạng thái đơn hẹn.

Hình 3.22: Giao diện danh sách nhập kho.

Chức năng quản lí linh kiện nhập kho cho phép quản trị viên xem chi tiết thông tin linh kiện và có thể tạo ra một đơn nhập PDF. Ngoài ra khi nhập linh kiện mới về thì nhân viên phải nhập thông tin vào hệ thống bằng cách nhấn “Thêm mới”.

Hình 3.23: Giao diện chức năng nhập kho linh kiện.

Khi nhân viên nhấn nút thêm mới để nhập thông tin linh kiện mới nhập kho thì phải điền đầy đủ các thông tin vào bảng “Thông tin import”, nếu nhập nhiều

linh kiện cùng lúc thì nhập thông tin bên bảng “Tạo nhiều item tự động” sau đó nhấn “Tạo Import”.

Danh sách Export				
StockOut ID	Employee Username	Out Date	Note	Actions
30	Admin	2025-11-23 18:44	Xuất kho tự động cho Order ID 202	Xem chi tiết Tạo hóa đơn xuất (PDF)
29	Admin	2025-11-23 09:20	Xuất kho tự động cho Order ID 201	Xem chi tiết Tạo hóa đơn xuất (PDF)
28	Admin	2025-11-18 14:24	Xuất kho tự động cho Order ID 181	Xem chi tiết Tạo hóa đơn xuất (PDF)
27	Admin	2025-11-17 12:22	Xuất kho tự động cho Order ID 161	Xem chi tiết Tạo hóa đơn xuất (PDF)
26	Admin	2025-11-15 18:49	Xuất kho tự động cho Order ID 144	Xem chi tiết Tạo hóa đơn xuất (PDF)
18	Admin	2025-11-14 02:25	Xuất kho tự động cho Order ID 143	Xem chi tiết Tạo hóa đơn xuất (PDF)
17	Admin	2025-11-14 02:18	Xuất kho tự động cho Order ID 142	Xem chi tiết Tạo hóa đơn xuất (PDF)
16	Admin	2025-11-14 01:57	Xuất kho tự động cho Order ID 141	Xem chi tiết Tạo hóa đơn xuất (PDF)
15	Admin	2025-11-11 09:28	Xuất kho tự động cho Order ID 125	Xem chi tiết Tạo hóa đơn xuất (PDF)
14	Admin	2025-11-11 09:23	Xuất kho tự động cho Order ID 124	Xem chi tiết Tạo hóa đơn xuất (PDF)

Trang 1 / 2 < Trước 1 2 Sau >

Hình 3.24: Giao diện chức năng xuất kho.

Chức năng quản lý danh sách linh kiện xuất kho chỉ cho phép nhân viên xem và không sửa được thông tin, nhân viên có thể chọn xem chi tiết thông tin linh kiện xuất kho hoặc tạo hóa đơn xuất bằng PDF.

Welcome, Admin
CONNECT_RESOURCE_ROLE_ADMIN

Chi tiết Export #30

Employee Name:	Out Date:	Note:
Admin	11/23/2025 6:44:47 PM	Xuất kho tự động cho Order ID 202

Items

Part Name	Manufacturer	Serial	Price
TEST	TEST	TEST-0-18	1000
TEST	TEST	TEST-0-19	1000

Logout (Admin)

Hình 3.25: Giao diện chức năng xem chi tiết đơn xuất kho.

Khi nhân viên nhấn “Xem chi tiết” từ giao diện quản lý danh sách linh kiện xuất kho thì sẽ hiện ra trang chi tiết đơn xuất bao gồm thông tin nhân viên xuất kho, ghi chú, ngày xuất kho cùng các thông tin chi tiết của linh kiện.

Quản lý hóa đơn							
Mã hóa đơn	StockOut ID	SĐT khách hàng	Nhân viên	Ngày lập	Tổng tiền	Trạng thái	Thao tác
8	30	0332880207	Admin	23/11/2025 18:44	0 ₫	PENDING	<button>Chi tiết</button> <button>Tải PDF</button>
7	29	1	Admin	23/11/2025 09:20	0 ₫	PENDING	<button>Chi tiết</button> <button>Tải PDF</button>
6	28	1	Admin	18/11/2025 14:24	0 ₫	PENDING	<button>Chi tiết</button> <button>Tải PDF</button>
5	27	0332880207	Admin	17/11/2025 12:22	0 ₫	PENDING	<button>Chi tiết</button> <button>Tải PDF</button>
4	26	0332880207	Admin	15/11/2025 18:49	0 ₫	PENDING	<button>Chi tiết</button> <button>Tải PDF</button>
3	18	0332880207	Admin	14/11/2025 02:25	0 ₫	PENDING	<button>Chi tiết</button> <button>Tải PDF</button>
1	16	1	Admin	14/11/2025 01:57	0 ₫	PENDING	<button>Chi tiết</button> <button>Tải PDF</button>

Hình 3.26: Giao diện chức năng quản lý hóa đơn.

Chức năng quản lý hóa đơn cho phép nhân viên xem và không được sửa, nhân viên có thể tìm kiếm hóa đơn theo mã hóa đơn, số điện thoại khách hàng, lọc các hóa đơn theo ngày tháng. Ngoài ra thì nhân viên có thể xem chi tiết hóa đơn hoặc tải hóa đơn dạng PDF về máy.

Chi tiết hóa đơn #8							
Thông tin chung				Ghi chú			
Mã hóa đơn:	8			Hóa đơn được sinh tự động từ phiếu xuất kho sau khi hoàn tất đơn hàng.			
StockOut ID:	30						
Khách hàng:	0332880207						
Nhân viên:	Admin						
Ngày lập:	23/11/2025 18:44						
Trạng thái:	PENDING						
Tổng tiền:	0 ₫						

Danh sách linh kiện							
#	Tên linh kiện	Hàng	Serial	Gía			
1	TEST	TEST	TEST-0-10	1,000 ₫			
2	TEST	TEST	TEST-0-19	1,000 ₫			

Danh sách dịch vụ				
#	Tên dịch vụ	Số lượng	Đơn giá	Thành tiền
1	Dịch vụ Bảo hành	1	0 ₫	0 ₫

Hình 3.27: Giao diện chức năng chi tiết hóa đơn.

Khi nhân viên nhấn “Chi tiết” ở giao diện quản lý hóa đơn thì sẽ hiện ra trang chi tiết hóa đơn, gồm các thông tin của nhân viên và khách hàng, ghi chú, thông tin linh kiện được dùng, các dịch vụ sử dụng và tổng tiền hóa đơn.

Danh sách linh kiện									
<input type="button" value="Quét QR Code"/> <input type="text"/> Tên linh kiện <input type="text"/> Serial <input type="text"/> Hàng <input type="button" value="Trong kho"/> <input type="text"/> Giá từ <input type="text"/> Giá đến <input type="button" value="Tim kiếm"/> <input type="button" value="Đặt lại"/>									
#	Name	Manufacturer	Serial	QR	Status	StockId	OrderId	Price	Actions
302	Mặt kính lưng IP XS	Apple	MKLXS-1		Trong kho	36	-	10000	<input type="button" value="Xem chi tiết"/>
303	Mặt kính lưng IP XS	Apple	MKLXS-2		Đã xuất kho	36	85	10000	<input type="button" value="Xem chi tiết"/>
304	Mặt kính lưng IP XS	Apple	MKLXS-3		Trong kho	36	-	10000	<input type="button" value="Xem chi tiết"/>
305	Mặt kính lưng IP XS	Apple	MKLXS-4		Trong kho	36	-	10000	<input type="button" value="Xem chi tiết"/>

Admin Dashboard

Hình 3.28: Giao diện danh sách linh kiện.

Chức năng quản lý danh sách linh kiện cho phép nhân viên xem và không chỉnh sửa, nhân viên có thể tìm kiếm linh kiện theo tên, mã serial hoặc hàng, hoặc lọc theo giá. Ngoài ra thì nhân viên có thể đăng nhập ứng dụng trên điện thoại và thực hiện quét mã QR để xem chi tiết linh kiện hoặc có thể bấm vào “Xem chi tiết” để xem.

Request Id	Employee Username	In Date	Order_Id	Status	Action
48	Admin	2025-11-28 22:58	262	Từ chối	Xem chi tiết
47	Admin	2025-11-28 22:09	243	Đồng ý	Xem chi tiết
46	Admin	2025-11-24 10:29	222	Đồng ý	Xem chi tiết
45	Admin	2025-11-20 21:18	202	Đồng ý	Xem chi tiết
44	Kithuatvien	2025-11-20 21:00	202	Đồng ý	Xem chi tiết
41	Admin	2025-11-20 13:35	201	Đồng ý	Xem chi tiết
40	Admin	2025-11-18 14:24	181	Đồng ý	Xem chi tiết
39	Admin	2025-11-17 12:21	161	Đồng ý	Xem chi tiết
38	Admin	2025-11-14 19:44	144	Đồng ý	Xem chi tiết

Hình 3.29: Giao diện danh sách yêu cầu linh kiện.

Chức năng quản lý danh sách yêu cầu linh kiện cho phép xem và thực hiện hành động từ chối hoặc đồng ý yêu cầu linh kiện.

Danh sách đơn hàng								Thêm mới		
#	Customer Phone	Receiver Emp	Handler Emp	Order Type	Received Date	Status	Description	Actions		
262	0909000123	Admin	Admin	REPAIR	28/11/2025 22:58	Đã tiếp nhận	Sua chua	Chi tiết	Hoàn thành	Hủy đơn hàng
243	0909000123	Admin	Kithuatvien	REPAIR	27/11/2025 19:42	Đã tiếp nhận	Sua chua linh kien	Chi tiết	Hoàn thành	Hủy đơn hàng
242	0909000123	Admin	Admin	WARRANTY	27/11/2025 19:42	Đã tiếp nhận	bao hanh linh kien	Chi tiết	Hoàn thành	Hủy đơn hàng
222	0332880207	Admin	Ktv02	WARRANTY	24/11/2025 10:29	Đã hủy	ABC	Chi tiết		
202	0332880207	Tieptan	Kithuatvien	WARRANTY	20/11/2025 20:05	Đã hoàn thành	ABC	Chi tiết		
201	1	Admin	Ktv02	WARRANTY	20/11/2025 13:35	Đã hoàn thành	AAA	Chi tiết		
181	1	Admin	Admin	WARRANTY	18/11/2025 14:24	Đã hoàn thành	A	Chi tiết		

Hình 3.30: Giao diện danh sách đơn hàng.

Chức năng quản lý danh sách đơn hàng cho phép nhân viên tìm kiếm theo số điện thoại, lọc theo loại đơn, trạng thái và ngày tháng. Ngoài ra nhân viên có thể tạo mới, xem chi tiết đơn hàng và thực hiện các hành động “Hoàn thành” hoặc “Hủy đơn hàng”.

Tạo đơn hàng mới

Thông tin đơn hàng

Customer Phone	Handler Username
<input type="text" value="-- Chọn khách hàng --"/>	<input type="text" value="-- Chọn nhân viên xử lý --"/>
Order Type	
<input type="text" value="-- Chọn loại đơn --"/>	
Description	<input type="text"/>

Dịch vụ

Dịch vụ	Đơn giá	Số lượng	Thành tiền
<input type="text" value="-- Chọn dịch vụ --"/>	0	<input type="text" value="1"/>	0
Thêm dòng			x

Tạo đơn

Hình 3.31: Giao diện chức năng tạo đơn hàng.

Khi nhân viên nhấn “Thêm mới” ở giao diện quản lý đơn hàng thì sẽ hiện trang tạo mới đơn hàng, nhân viên phải nhập đầy đủ thông tin khách hàng bao gồm tên khách hàng, loại đơn và nhân viên xử lý đơn, ngoài ra phải chọn loại dịch vụ sau đó nhấn “Tạo đơn”.

Chi tiết đơn hàng #262

Thông tin đơn hàng

SỐ ĐIỆN THOẠI KH	NHÂN VIÊN TIẾP NHẬN	NHÂN VIÊN XỬ LÝ	LOẠI ĐƠN	NGÀY TIẾP NHẬN	TRẠNG THÁI
0909000123	Admin	Admin	REPAIR	28/11/2025 22:58	Tùy tiếp nhận

MÔ TẢ
Sua chua

Danh sách dịch vụ

Mã dịch vụ	Tên dịch vụ	Mô tả	Số lượng	Đơn giá	Thành tiền
2	Dịch vụ Sửa chữa tổng quát		1	350,000 đ	350,000 đ

Tổng cộng: 350,000 đ

① Đơn hàng này chưa có linh kiện nào được gán.

Danh sách linh kiện được yêu cầu (Part Request)

Mã linh kiện	Tên linh kiện	Nhà sản xuất	Serial	Trạng thái	Giá
526	TEST	TEST	TEST-0-13	Bã được đăng kí	1,000 đ

Hình 3.32: Giao diện chi tiết đơn hàng.

Khi nhân viên viên nhán “Chi tiết” ở giao diện quản lí đơn hàng sẽ hiện trang chi tiết đơn hàng đó bao gồm các thông tin khách hàng, nhân viên xử lí, loại đơn, loại dịch vụ và linh kiện được yêu cầu (nếu có).

Trigger Audit

Trigger Audit: **Đang bật (ENABLED)** Số lượng: 12 / 12 triggers đang enabled

12 / 12

Bật Trigger Audit **Tắt Trigger Audit**

Log ID	Thời gian	DB User	OS User	Machine	Module	App Role	Emp ID	Customer Phone	Object Name	Note	DML Type	Changed Columns	Old Values	New Values
469	30/11/2025 01:24:05	01234567891	ASUS	LAPTOP-AUDVIVGMLAPTOP-AUDVRGM	WebAPI-WEB	ROLE_KHACH_HANG	01234567891	CUSTOMER_APPOINTMENT	INSERT operation on CUSTOMER_APPOINTMENT	INSERT			APPOINTMENT_ID: 16	
													CUSTOMER_PHONE: 01234567891	
													APPOINTMENT_DATE: 2025-12-03	
													STATUS: Đã lên lịch	
													DESCRIPTION: Bảo hành	
													PHONE: 01234567891	PHONE: 01234567891
													FULL_NAME: Cuong	FULL_NAME: Cuong
468	30/11/2025 01:14:05	01234567891	ASUS	LAPTOP-AUDVIVGMLAPTOP-AUDVRGM	WebAPI-MOBILE	ROLE_KHACH_HANG	01234567891	CUSTOMER	UPDATE operation on CUSTOMER - Changed:	UPDATE			EMAIL: c@gmail.com	EMAIL: c@gmail.com
													ADDRESS:	ADDRESS:
													CREATED_AT: 2025-11-30 01:13:01	CREATED_AT: 2025-11-30 01:13:01

Hình 3.33: Giao diện Trigger Audit.

Chức năng Trigger Audit cho phép nhân viên xem được chi tiết thông tin người dùng thực hiện các nghiệp vụ.

Standard Audit						
Standard Audit		Số lượng: 18 / 18 objects đang được audit				
		18 / 18				
Bật Standard Audit	Tắt Standard Audit					
Thời gian	DB User	Schema	Object	Action	SQL Text	
30/11/2025 01:24:05	01234567891	APP	CUSTOMER_APPOINTMENT	INSERT		
30/11/2025 01:14:26	01234567891	APP	CUSTOMER	UPDATE		
30/11/2025 01:13:01	APP	APP	CUSTOMER	INSERT		
28/11/2025 23:36:35	APP	APP	PART	UPDATE		
28/11/2025 23:15:27	APP	APP	ORDERS	DELETE		
28/11/2025 23:11:39	APP	APP	ORDERS	DELETE		

Hình 3.34: Giao diện Standard Audit

Chức năng Standard Audit cho phép nhân viên xem được chi tiết thông tin người dùng thực hiện các nghiệp vụ.

3.4.8.2 Giao diện khách hàng



Hình 3.35: Giao diện trang chủ khách hàng.

Giao diện trang chủ của khách hàng cho phép khách hàng chọn các chức năng “Đặt lịch”, “Xem lịch đặt”, “Xem đơn hàng”, “Trợ giúp” hoặc là “Đổi mật khẩu”.



Home Đặt lịch Xem lịch đặt Xem đơn hàng Trợ giúp

Đổi mật khẩu Logout (01234567891)

Đăng nhập ứng dụng di động bằng QR

Đang đăng nhập Web, bạn có thể mở app Mobile Service (chưa đăng nhập) > chọn "Đăng nhập bằng mã Web" > nhập mã dưới đây để được đăng nhập ngay.

Mở app Mobile Service và nhập mã này để đăng nhập.



DPUBDBEQ

Hết hạn sau 103s

Tạo mã mới

NHẬP MÃ NÀY TRÊN APP (NẾU KHÔNG QUÉT ĐƯỢC)

DPUBDBEQ

Copy

Mã chỉ có hiệu lực trong 2 phút. Khi app xác nhận thành công, trạng thái tại đây sẽ thay đổi.

Hình 3.36: Giao diện chức năng đăng nhập bằng QR.

Khách hàng có thể dùng điện thoại và truy cập vào ứng dụng để đăng nhập thông qua quét mã QR.



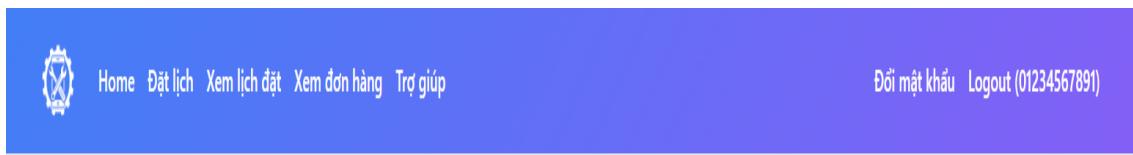
Hình 3.37: Giao diện chức năng đặt lịch cho khách hàng.

Chức năng đặt lịch cho phép khách hàng chọn ngày đặt lịch và phải điền đầy đủ thông tin mô tả sau đó nhấn “Đặt lịch”.

A screenshot of a web application interface titled "Lịch hẹn của bạn" (Your bookings). At the top, there is a navigation bar with icons for Home, Book, View bookings, View orders, and Help, along with links for Change password and Logout. Below the title, a table displays a single booking entry:

Hình 3.38: Giao diện chức năng xem lịch hẹn của khách hàng.

Giao diện xem lịch hẹn giúp khách hàng có thể kiểm tra và theo dõi trạng thái lịch hẹn, thời gian hẹn.



Danh sách đơn hàng của tôi						
Mã đơn	Loại đơn	Ngày tiếp nhận	Trạng thái	Nhân viên tiếp nhận	Nhân viên xử lý	Hành động
 Không có đơn hàng nào						

Hình 3.39: Giao diện chức năng xem đơn hàng cho khách hàng.

Giao diện danh sách đơn hàng giúp khách hàng có thể nắm được danh sách đơn hàng đã được thực hiện.

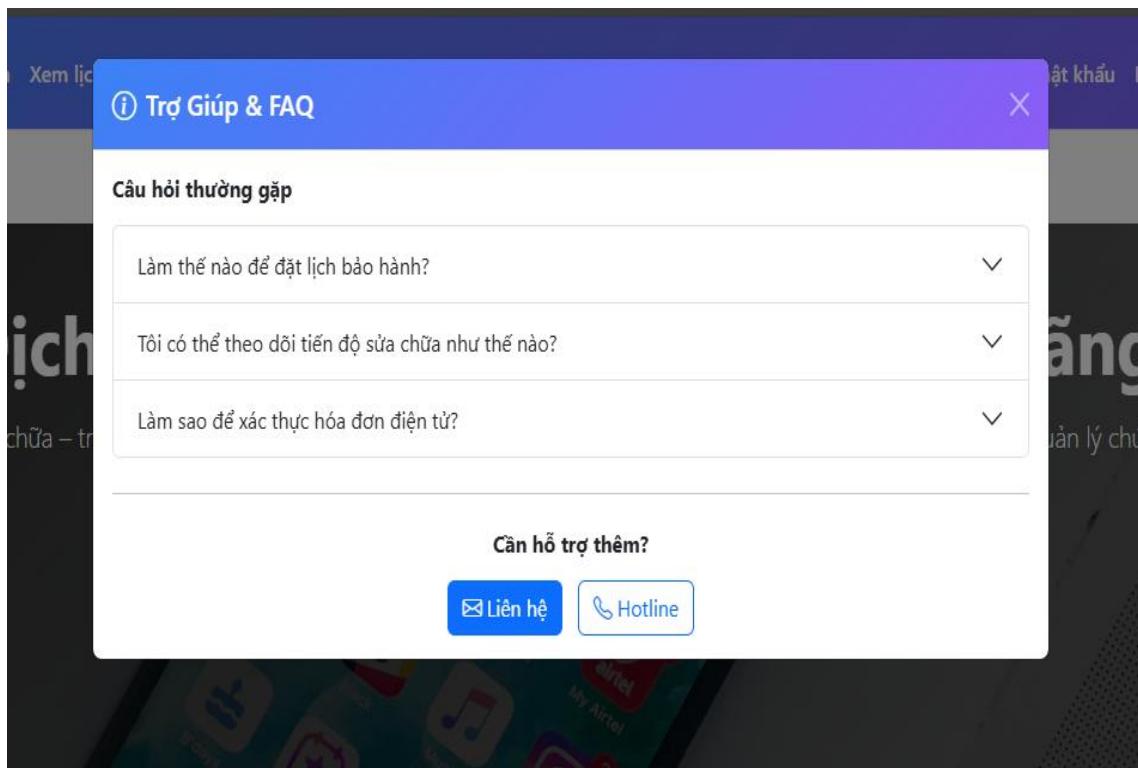
The screenshot shows a contact form titled "Liên hệ với chúng tôi" (Contact us). It includes fields for "Email của bạn" (Your email) containing "your.email@example.com", "Nội dung yêu cầu" (Request content) with placeholder text "Mô tả chi tiết vấn đề bạn gặp phải...", and a blue "Gửi yêu cầu" (Send request) button.

Hình 3.40: Giao diện chức năng liên hệ cho khách hàng.

Giao diện đổi mật khẩu cho khách hàng khả năng thay đổi mật

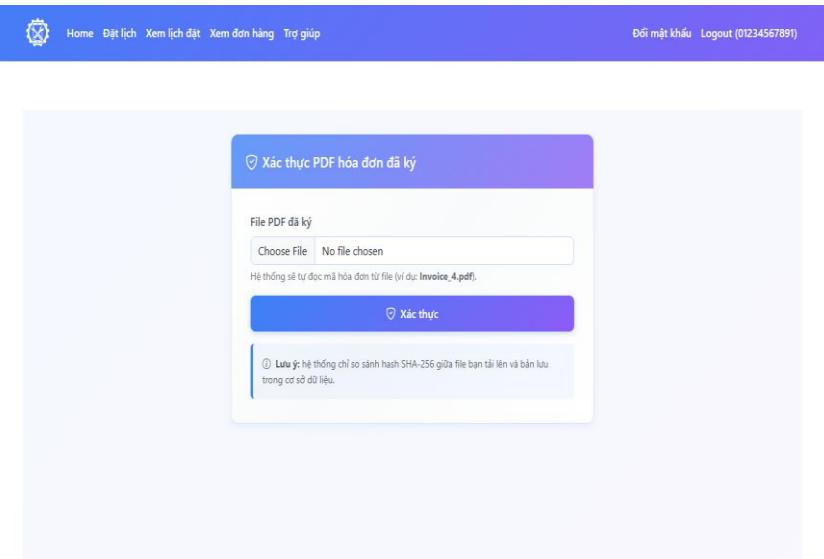
The screenshot shows a password change form titled "Đổi mật khẩu". It has three input fields: "Mật khẩu hiện tại" (Current password), "Mật khẩu mới" (New password), and "Xác nhận mật khẩu mới" (Confirm new password). Below the fields is a blue "Đổi mật khẩu" (Change password) button.

Hình 3.41: Giao diện chức năng đổi mật khẩu cho khách hàng.



Hình 3.42: Giao diện chức năng trợ giúp và FAQ.

Khách hàng có thể nhấp chọn vào biểu tượng dấu chấm hỏi ở dưới góc phải màn hình để có thể được giải đáp các thắc mắc thông thường.

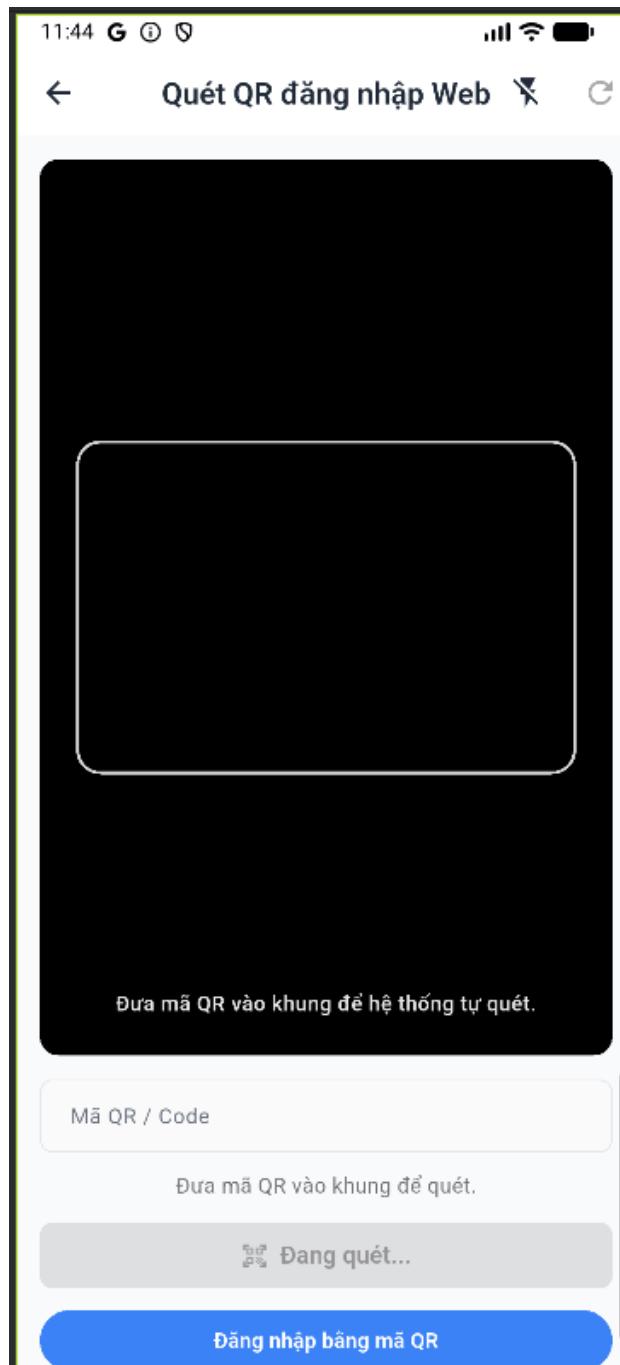


Hình 3.43: Giao diện chức năng xác thực hóa đơn chứa chữ ký số.

Khi khách hàng chọn vào chức năng xác thực ở giao diện trang chủ thì sẽ hiện ra một trang xác thực để đưa hóa đơn vào xác thực chữ ký số.

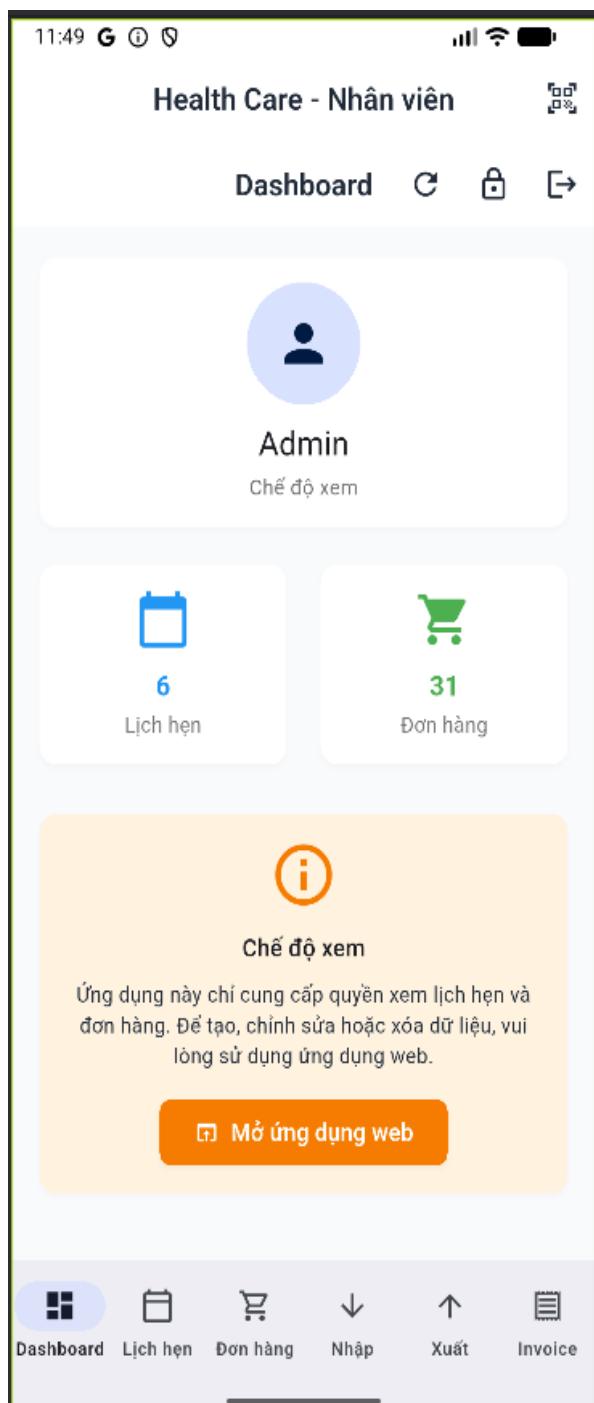
3.4.9. Giao diện mobile

3.4.9.1 Giao diện nhân viên



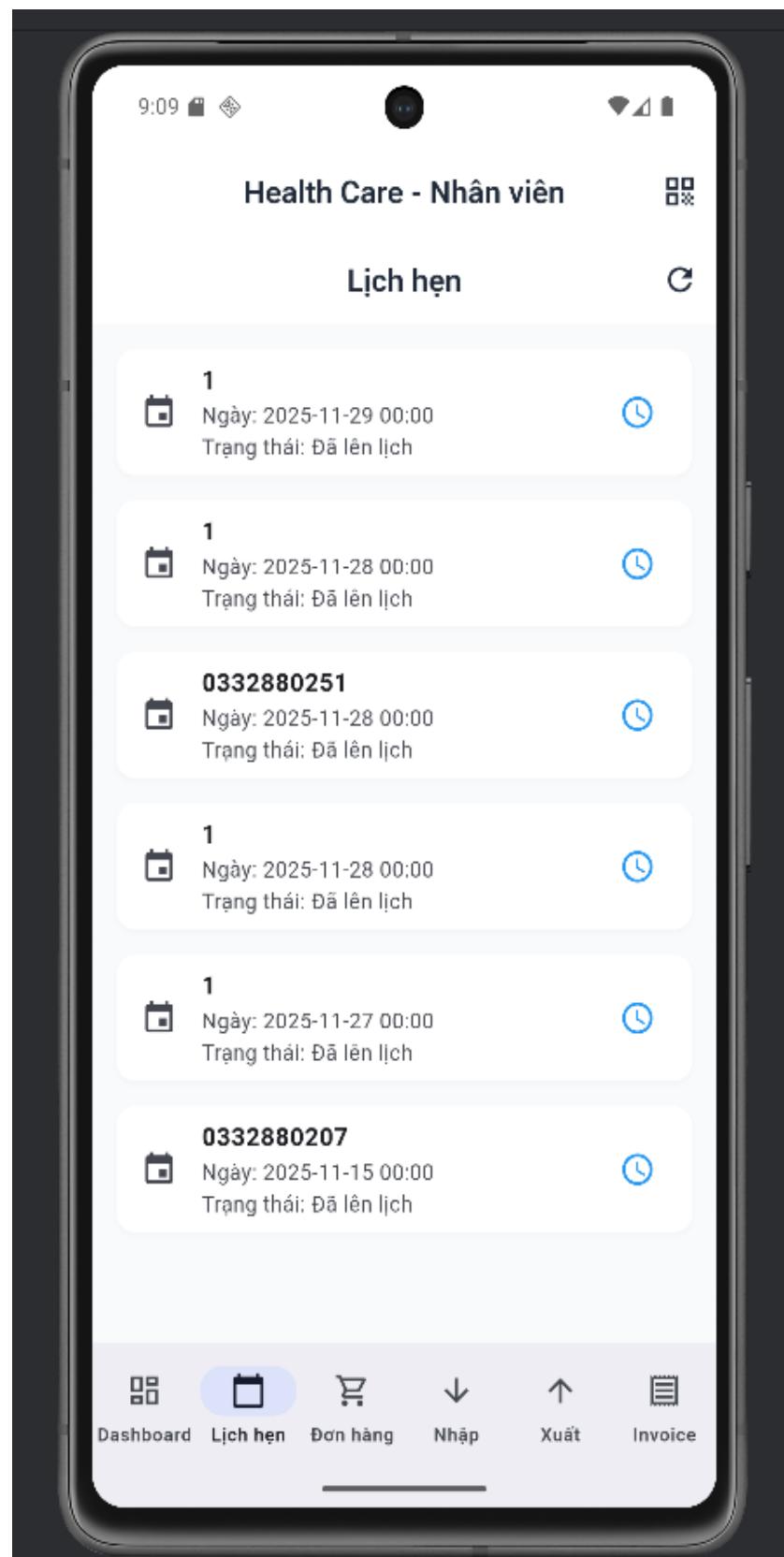
Hình 3.44: Giao diện quét QR đăng nhập web.

Ở giao diện đăng nhập nhân viên thì nhân viên có thể lựa chọn nhập tên đăng nhập kèm mật khẩu hoặc đăng nhập vào ứng dụng trên điện thoại và quét QR để đăng nhập trên website.



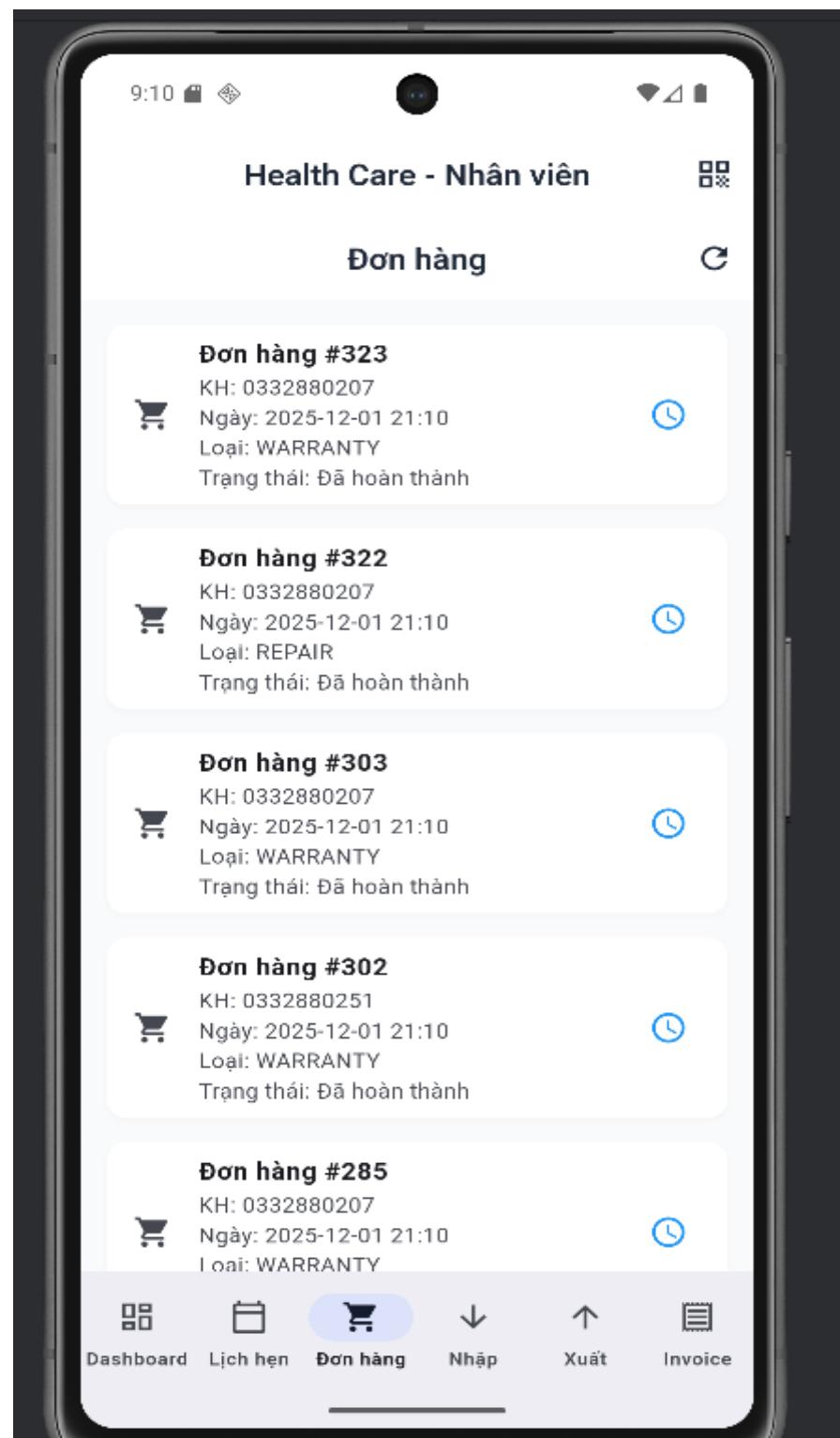
Hình 3.45: Giao diện nhân viên.

Ở giao diện trang chủ của nhân viên cho phép nhân viên xem lịch hẹn, đơn hàng, nhập linh kiện, xuất linh kiện, quản lý hóa đơn và đổi mật khẩu.



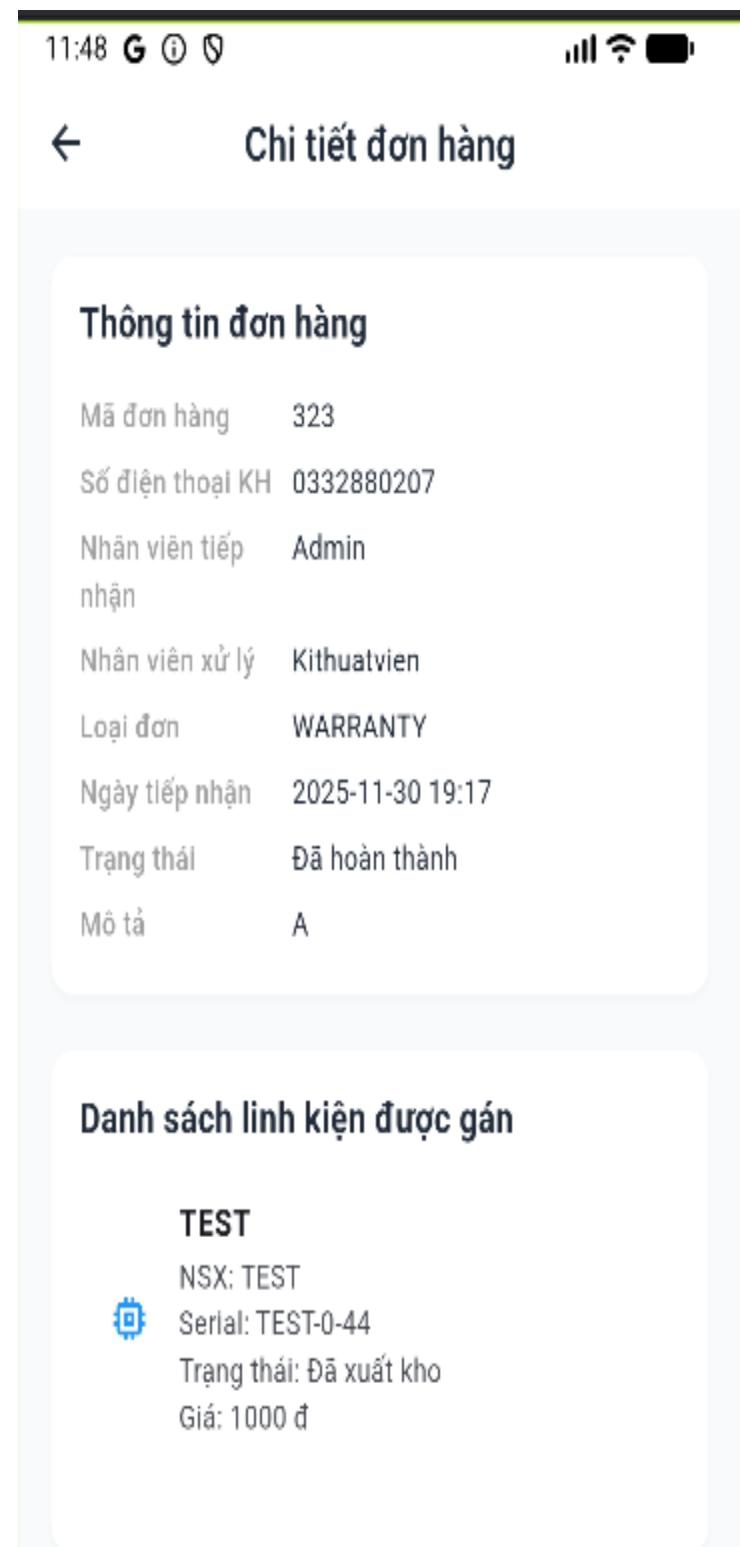
Hình 3.46: Giao diện lịch hẹn

Khi chọn “Lịch hẹn” ở giao diện trang chủ thì sẽ hiện ra danh sách các lịch hẹn đã được khách hàng đặt cùng với thời gian và trạng thái của lịch hẹn.



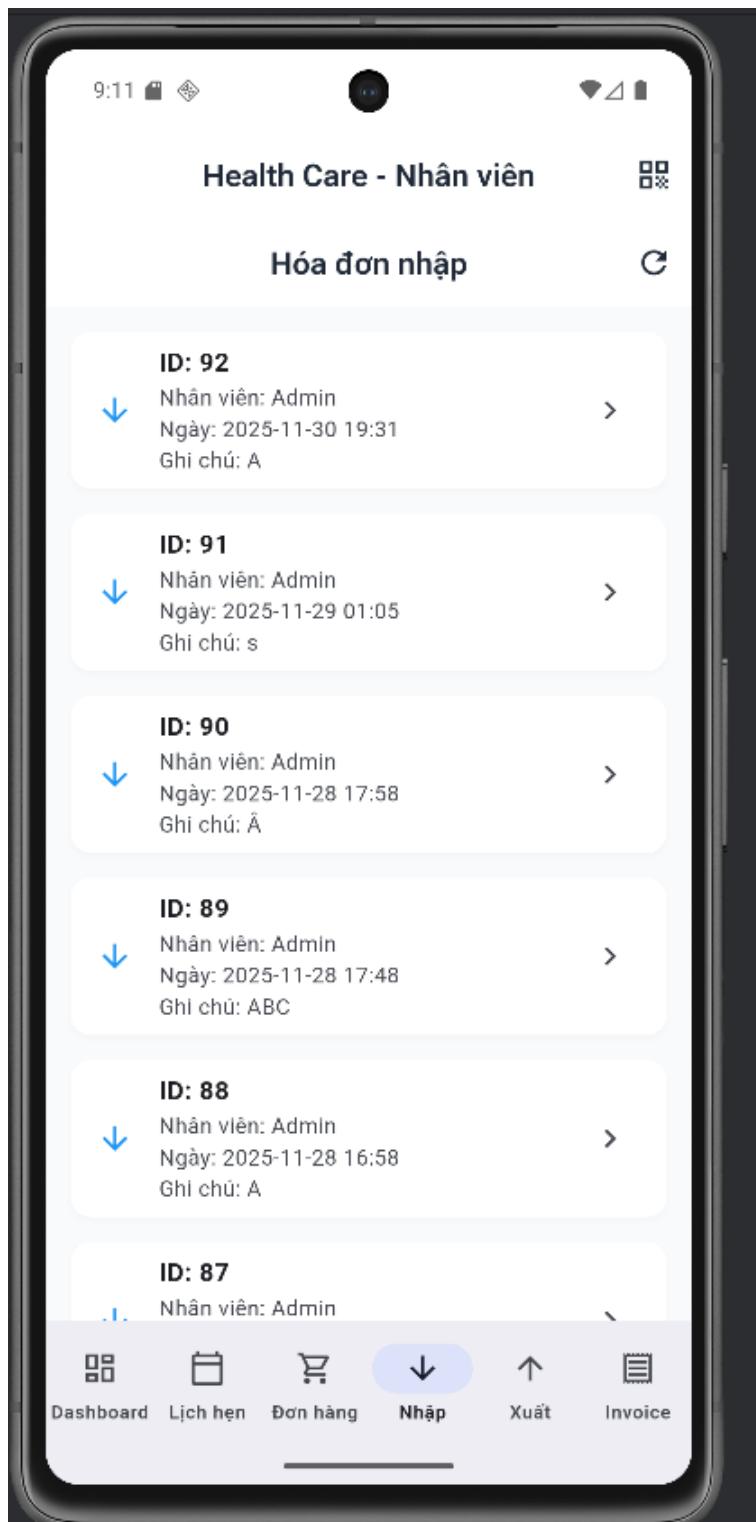
Hình 3.47: Giao diện quản lý đơn hàng

Chọn chức năng “Đơn hàng” ở giao diện trang chủ sẽ hiện thị ra danh sách các đơn hàng đang có.



Hình 3.48: Giao diện chi tiết đơn hàng

Ở giao diện danh sách đơn hàng thì người dùng có thể nhấn vào một đơn hàng bất kì để xem chi tiết thông tin của đơn hàng đó.



Hình 3.49: Giao diện Quản lý đơn nhập

Chọn chức năng “Nhập” ở giao diện trang chủ sẽ hiện ra danh sách các đơn nhập linh kiện.

Thông tin chung

ID: 92

Nhân viên: Admin

Ngày: 2025-11-30 19:31

Ghi chú: A

Chi tiết sản phẩm

AQ

Nhà sản xuất: adáđá

Serial: 13savf

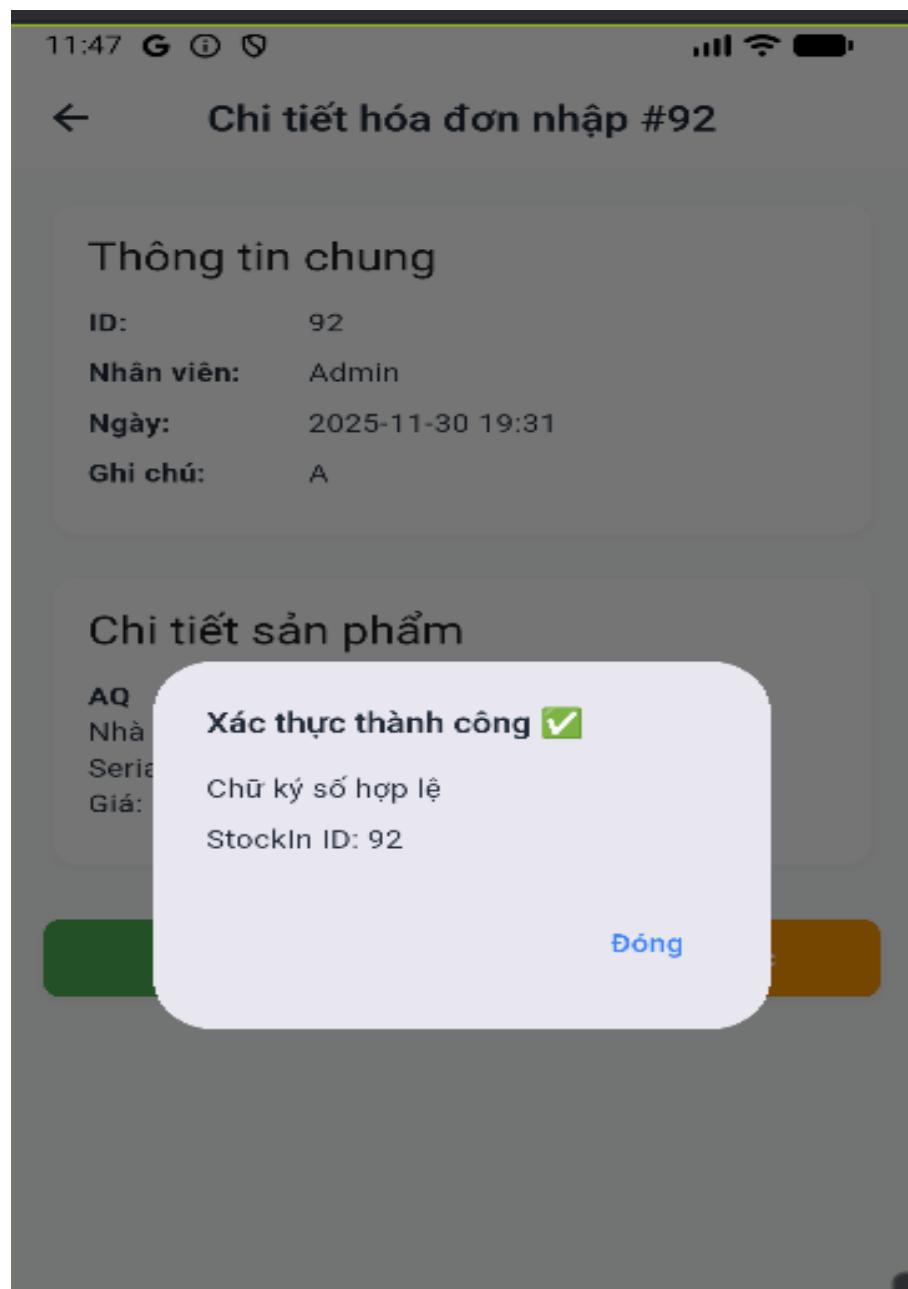
Giá: 10000

 Tải PDF

 Xác thực

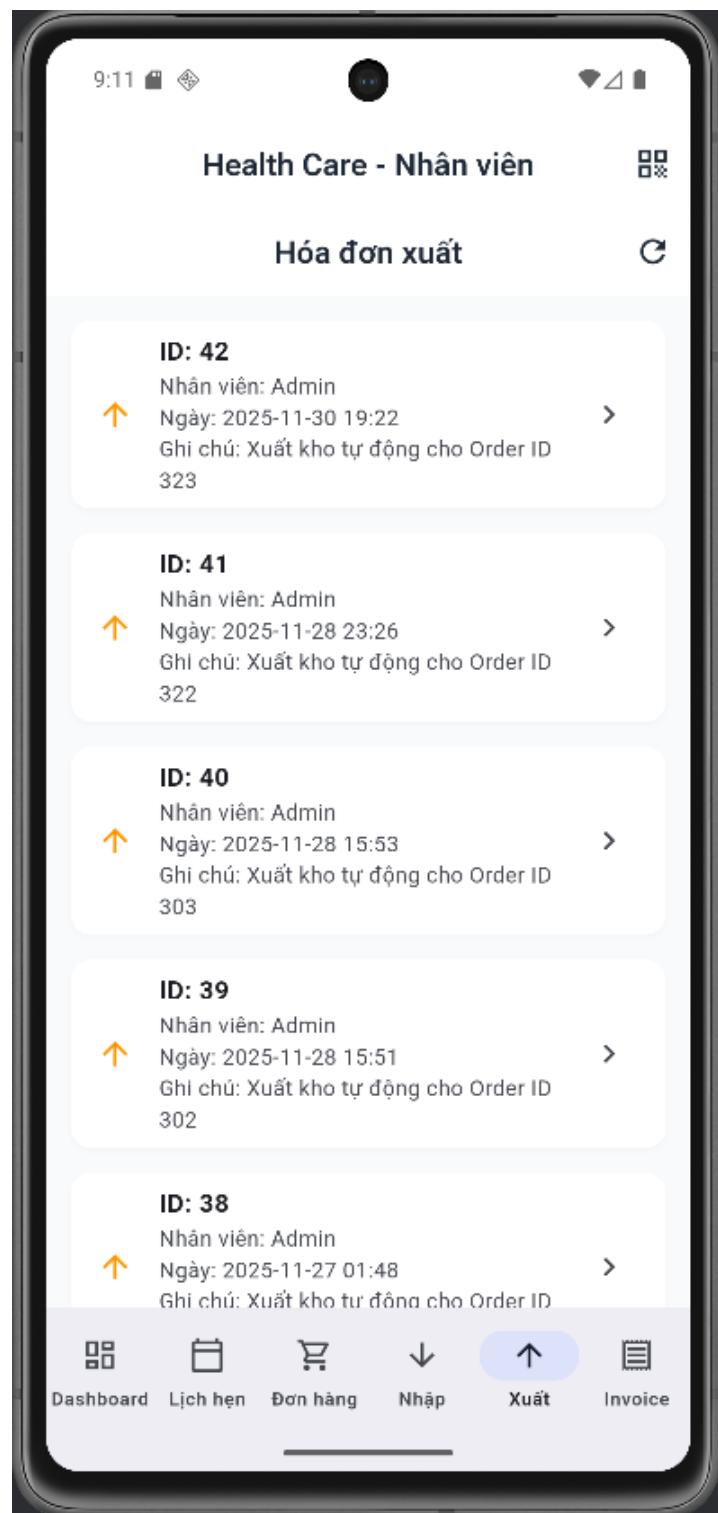
Hình 3.50: Giao diện chi tiết đơn nhập

Khi chọn một đơn nhập bất kỳ trong danh sách đơn nhập thì sẽ hiển thị chi tiết đơn nhập và nhân viên có thể chọn tải đơn nhập về máy với dạng PDF và có thể xác thực chữ ký số của đơn nhập này.



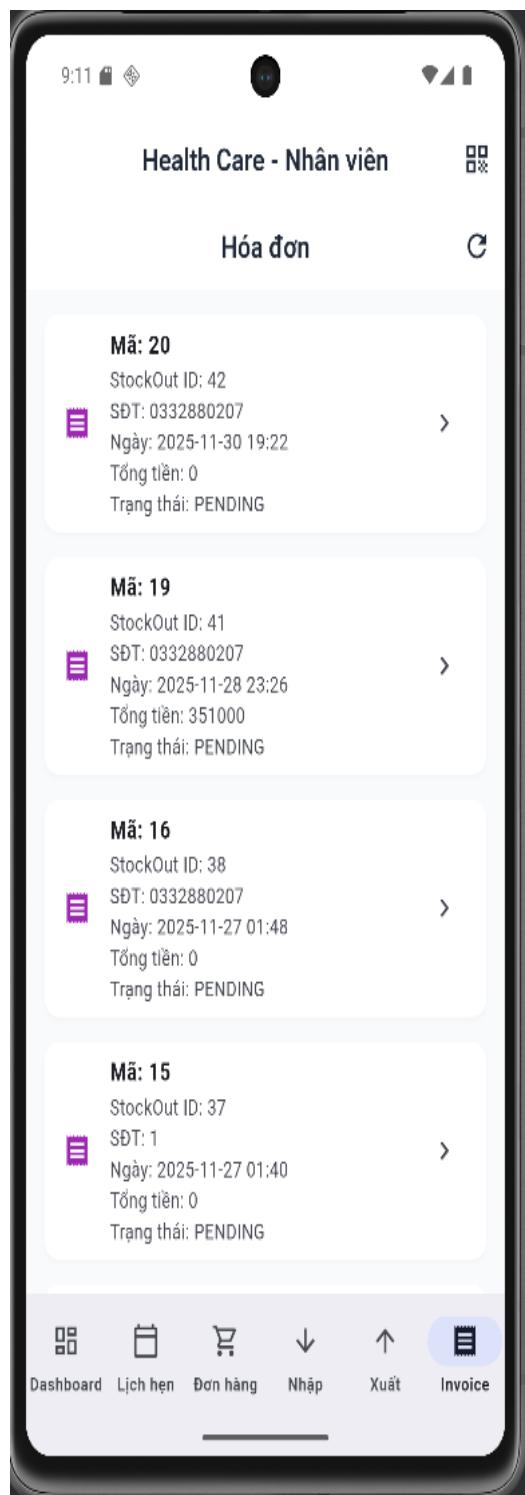
Hình 3.51: Giao diện xác thực chữ ký số bên cơ sở dữ liệu.

Khi chọn xác thực thì hệ thống sẽ xác thực chữ ký số của hóa đơn và hiện lại thông báo trên giao diện ứng dụng.



Hình 3.52: Giao diện Quản lý hóa đơn xuất.

Khi chọn chức năng “Xuất” sẽ hiện thị danh sách các đơn xuất kho kèm thông tin nhân viên, ngày tháng và ghi chú phía dưới.



Hình 3.53: Giao diện Quản lý hóa đơn.

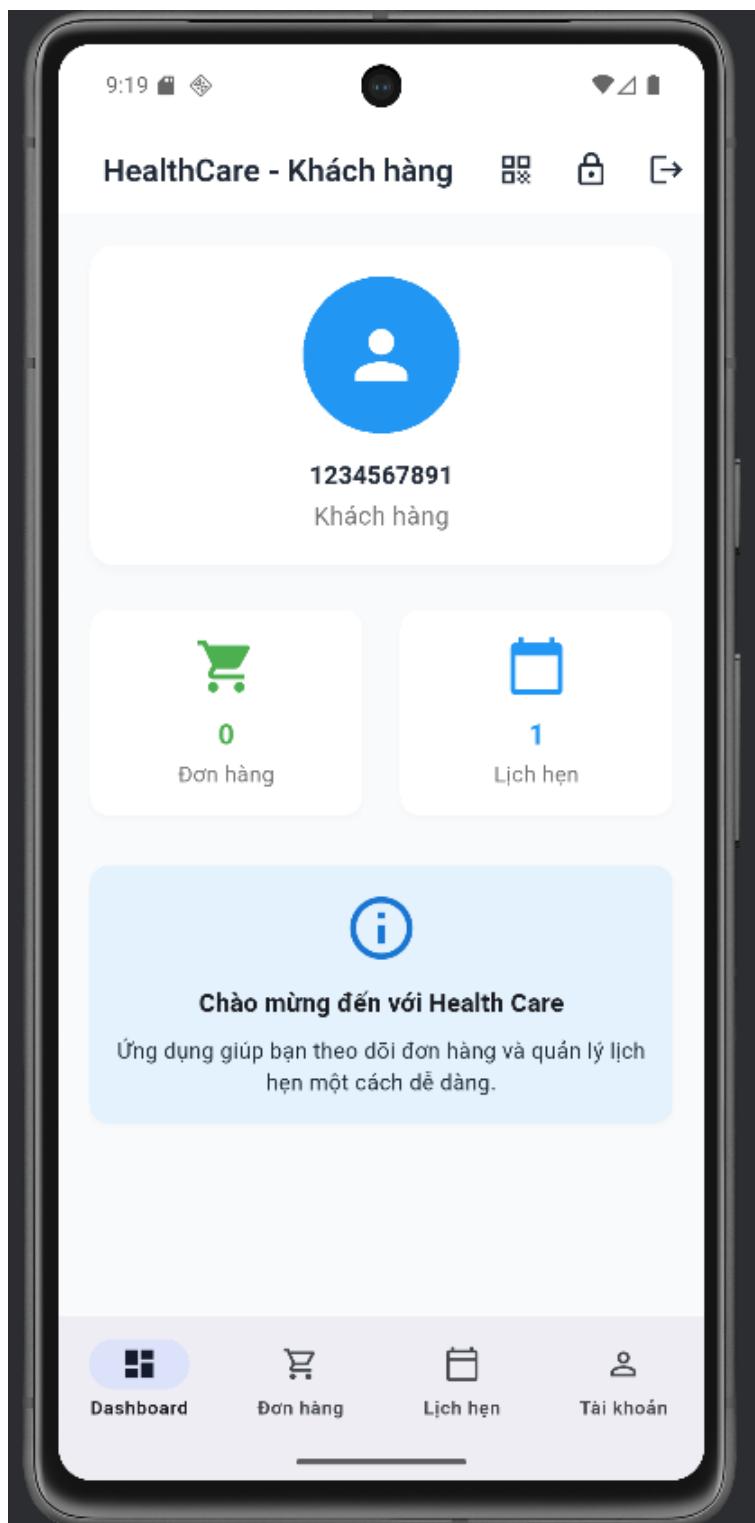
Khi chọn chức năng “Invoice” sẽ hiển thị danh sách các hóa đơn kèm mã đơn xuất, số điện thoại khách hàng, ngày tạo, tổng tiền hóa đơn và trạng thái hóa đơn.



Hình 3.54: Giao diện đổi mật khẩu.

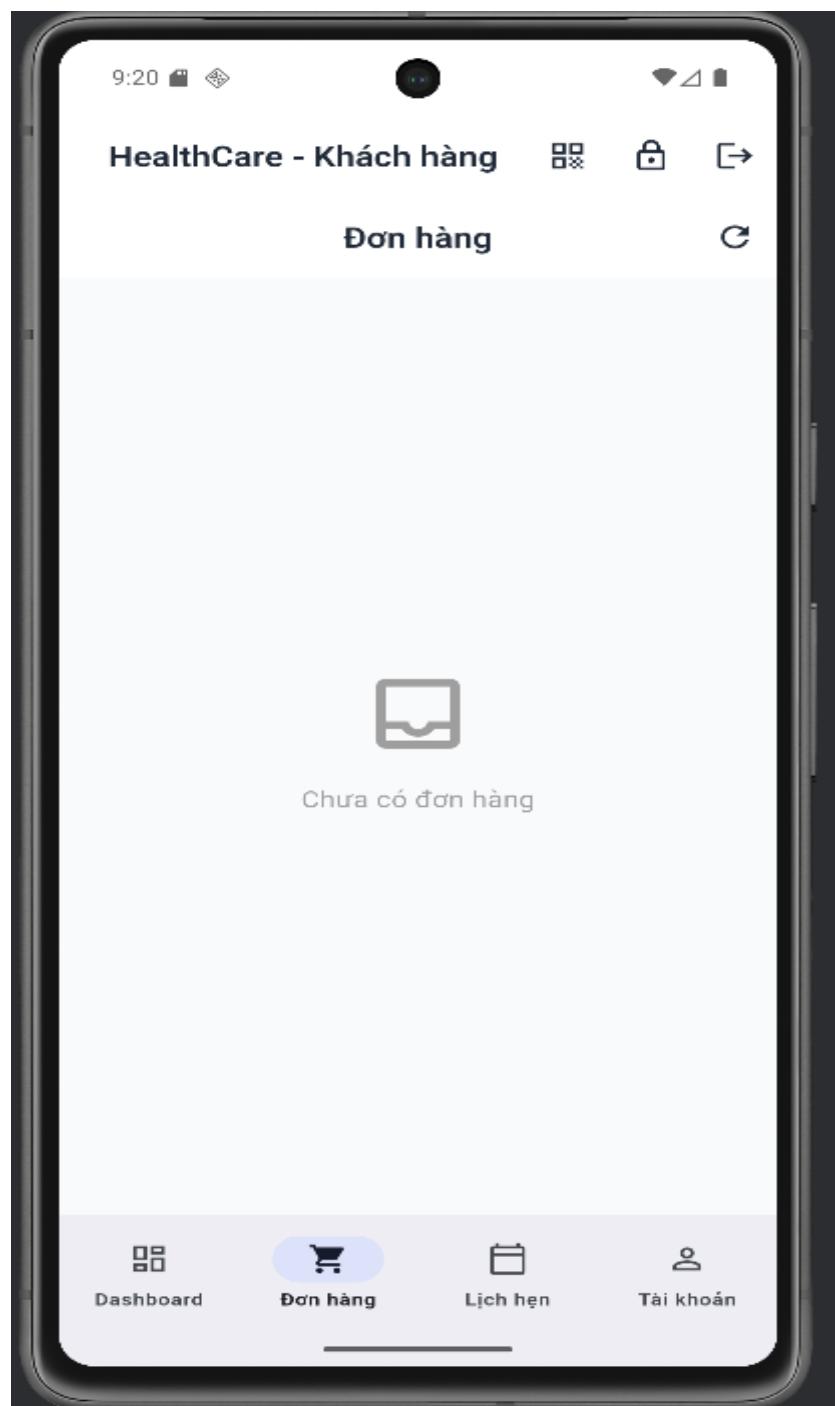
Nhân viên có thể thay đổi mật khẩu thường xuyên để cho tài khoản bảo mật hơn.

3.4.9.2 Giao diện khách hàng



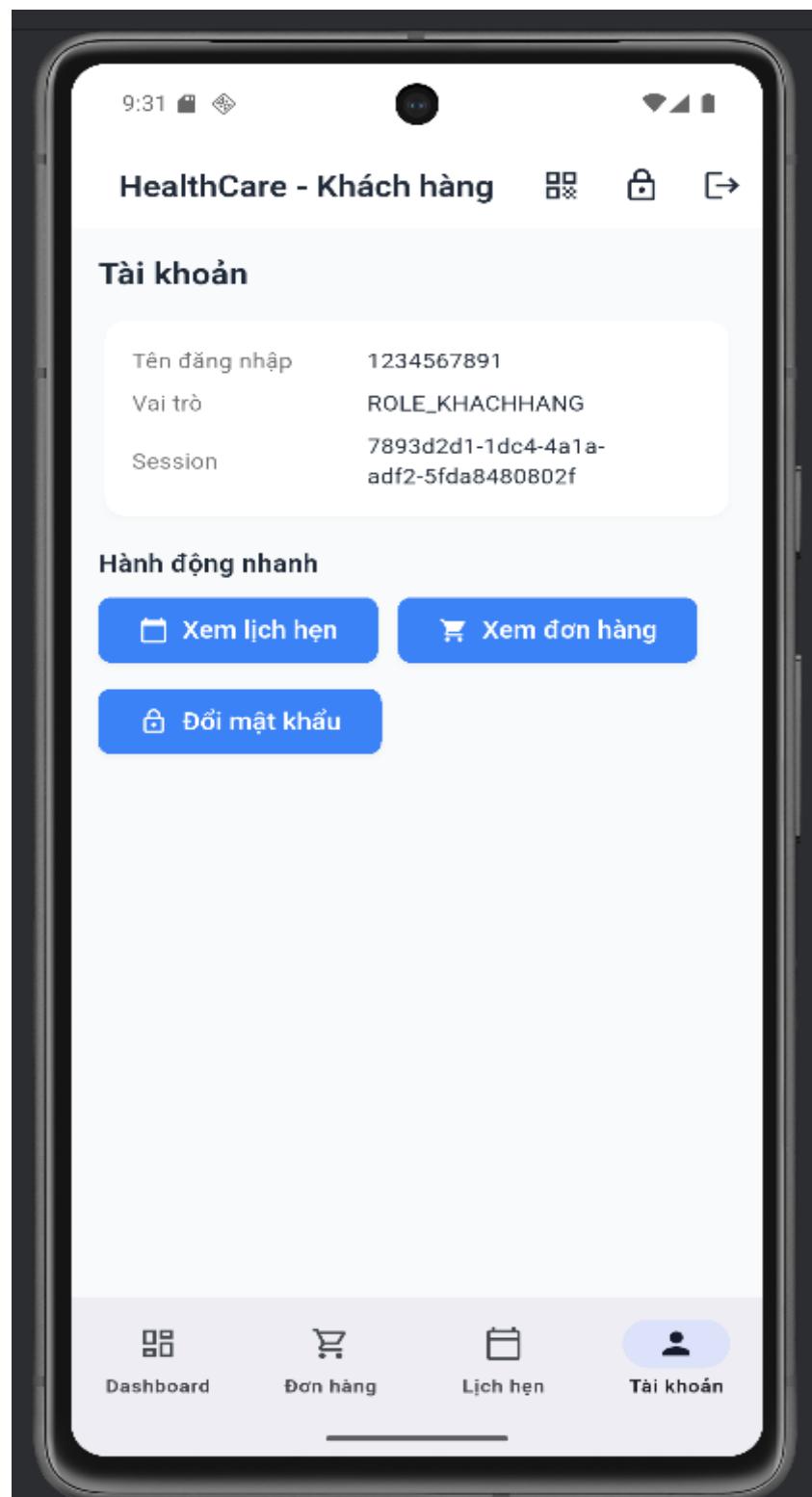
Hình 3.55: Giao diện trang chủ khách hàng.

Đăng nhập vào tài khoản khách hàng sẽ vào giao diện trang chủ của ứng dụng và ở tài khoản khách hàng sẽ có các chức năng quét mã QR để đăng nhập website, đổi mật khẩu, xem đơn hàng, hẹn và xem thông tin tài khoản.



Hình 3.56: Giao diện đơn hàng.

Chọn chức năng “Đơn hàng” để xem các đơn hàng của mình đã đặt.



Hình 3.57: Giao diện thông tin khách hàng.

Chọn chức năng “Tài khoản” để xem các thông tin chi tiết tài khoản của mình

3.5. Thiết kế API

Bảng 3.18: Common API

Phương thức	Đường dẫn API	Mô tả
GET	/api/Common/Appointment	Truy xuất/lấy thông tin các cuộc hẹn đã tồn tại.
GET	/api/Common/Order	Lấy thông tin các đơn hàng sửa chữa hoặc bảo hành
GET	/api/Common/Order/{orderId}/details	Truy vấn thông tin sâu về một đơn hàng cụ thể

Bảng 3.19: Public API

Phương thức	Đường dẫn API	Mô tả
GET	/api/Public/QrLogin/status/{qrLoginId}	Kiểm tra trạng thái của một yêu cầu đăng nhập QR
GET	/api/Public/Security/server-public-key	Cung cấp khóa công khai của máy chủ để client mã hóa dữ liệu nhạy cảm trước khi gửi lên server
GET	/api/Public/WebToMobileQr/status/{qrLoginId}	Kiểm tra trạng thái của một QR kết nối đã được tạo
POST	/api/Public/Customer/login	Xác thực người dùng, kiểm tra thông tin đăng nhập
POST	/api/Public/Customer/login/secure	Xác thực người dùng sử
POST	/api/Public/Customer/register	Tạo tài khoản khách hàng mới và lưu trữ vào cơ sở dữ liệu.
POST	/api/Public/Customer/logout	Đăng xuất tài khoản

Phương thức	Đường dẫn API	Mô tả
POST	/api/Public/Customer/change-password	Đổi mật khẩu với xác thực bổ sung
POST	/api/Public/Customer/change-password/secure	Mã hóa đường truyền khi đổi mật khẩu
POST	/api/Public/Customer/forgot-password/generate-otp	Gửi OTP để xác minh quên mật khẩu
POST	/api/Public/Customer/forgot-password/reset	Đặt lại mật khẩu mới qua OTP
POST	/api/Public/QrLogin/create	Tạo một yêu cầu đăng nhập bằng QR Code
POST	/api/Public/QrLogin/confirm	Được gọi từ ứng dụng mobile sau khi người dùng quét QR và đồng ý kết nối với website
POST	/api/Public/Security/register-client-key	Cho phép client đăng ký khóa công khai của mình với server để server có thể mã hóa dữ liệu gửi ngược lại client
POST	/api/Public/Verify/verify-invoice	xác thực tính hợp lệ của một hóa đơn
POST	/api/Public/WebToMobileQr/create	Tạo một QR code tạm thời để kết nối phiên làm việc từ Website sang ứng dụng Mobile
POST	/api/Public/WebToMobileQr/confirm	Được gọi từ ứng dụng Mobile sau khi người dùng quét QR và đồng ý kết nối với Website

Bảng 3.20: Admin API

Phương thức	Đường dẫn API	Mô tả
GET	/api/Admin/Customer	Lấy toàn bộ dữ liệu hồ sơ khách hàng phục vụ mục đích quản trị

Phương thức	Đường dẫn API	Mô tả
GET	/api/Admin/Employee	tạm thời vô hiệu hóa quyền truy cập của khách hàng vào hệ thống
GET	/api/Admin/Employee/{id}	Khôi phục quyền truy cập cho tài khoản khách hàng đã bị khóa
GET	/api/admin/Import	Lấy danh sách tất cả các phiếu nhập kho hiện có trong hệ thống
GET	/api/admin/Import/{stockinId}/verify	Xác minh tính hợp lệ của một phiếu nhập dựa trên mã stockinId.
GET	/api/admin/Import/{stockinId}/invoice	Lấy thông tin hóa đơn của phiếu nhập tương ứng với stockinId
GET	/api/admin/Import/{stockinId}/details	Lấy toàn bộ thông tin chi tiết của một phiếu nhập
GET	/api/admin/Export	Truy xuất danh sách toàn bộ phiếu xuất kho trong hệ thống
GET	/api/admin/Export/{stockoutId}/invoice	Lấy thông tin hóa đơn của phiếu xuất tương ứng với stockoutId
GET	/api/admin/Export/{stockoutId}/details	Lấy đầy đủ thông tin chi tiết của phiếu xuất
GET	/api/admin/Export/{stockoutId}/verify	Kiểm tra và xác minh phiếu xuất dựa trên ID
GET	/api/Admin/Order/services	Lấy danh sách các dịch vụ có trong hệ thống
GET	/api/Admin/Order/by-order-type	Lọc danh sách đơn hàng theo loại đơn
GET	/api/Admin/Order/{orderId}/details	Lấy thông tin chi tiết của đơn hàng

Phương thức	Đường dẫn API	Mô tả
GET	/api/Admin/Order/{orderId}/services	Lấy danh sách dịch vụ thuộc đơn hàng
GET	/api/Admin/Order/customer-phones	Lấy danh sách số điện thoại khách hàng có trong hệ thống
GET	/api/Admin/Order/handler-usernames	Lấy danh sách tài khoản nhân viên có thể xử lý đơn hàng
GET	/api/admin/Invoice/{invoiceId}/verify	Xác thực (kiểm tra tính hợp lệ) của một hóa đơn dựa trên invoiceId
GET	/api/admin/Invoice	Lấy danh sách tất cả hóa đơn trong hệ thống
GET	/api/admin/Invoice/{invoiceId}/details	Lấy thông tin chi tiết của một hóa đơn, bao gồm: tổng tiền, chi tiết dịch vụ, khách hàng,...
GET	/api/admin/Invoice/{invoiceId}/invoice	Tải về hoặc hiển thị file hóa đơn dạng PDF
GET	/api/Admin/Part	Lấy danh sách tất cả linh kiện trong hệ thống
GET	/api/Admin/Part/in-stock	Lấy danh sách các linh kiện đang còn tồn kho, chưa gán vào đơn hàng nào
GET	/api/Admin/Part/{serial}/details	Lấy thông tin chi tiết của một linh kiện dựa trên số serial
GET	/api/Admin/Part/{orderId}/by-order-id	Lấy danh sách linh kiện được sử dụng trong đơn hàng tương ứng với orderId
GET	/api/Admin/Part/{orderId}/by-part-request	Lấy danh sách linh kiện được yêu cầu cho một đơn hàng

Phương thức	Đường dẫn API	Mô tả
GET	/api/Admin/Partrequest	Lấy danh sách tất cả yêu cầu linh kiện trong hệ thống
GET	/api/Admin/Partrequest/{requestId}/by-request-id	Trả về chi tiết một yêu cầu linh kiện dựa trên requestId
GET	/api/Admin/Profile/users	Lấy danh sách tất cả người dùng và thông tin profile được gán cho từng người dùng
GET	/api/Admin/Profile	Lấy danh sách tất cả các profile trong hệ thống
GET	/api/Admin/Profile/{profileName}	Lấy thông tin chi tiết của một profile dựa trên tên profile
GET	/api/Admin/Role/users	Lấy danh sách người dùng tương ứng với từng vai trò trong hệ thống
GET	/api/Admin/Role/roles	Lấy danh sách tất cả các vai trò trong hệ thống
GET	/api/Admin/Role/{Username}/roles	Lấy danh sách các vai trò mà một người dùng đang có
POST	/api/Admin/Customer/unlock	Mở khóa tài khoản khách hàng bị khóa
POST	/api/Admin/Customer/lock	Khóa tài khoản khách hàng
POST	/api/Admin/Employee/unlock	Mở khóa tài khoản nhân viên bị khóa
POST	/api/Admin/Employee/lock	Khóa tài khoản nhân viên
POST	/api/Admin/Employee/change-password	Thay đổi mật khẩu của nhân viên hiện tại và chưa có xác thực

Phương thức	Đường dẫn API	Mô tả
POST	/api/Admin/Employee/change-password/secure	Thay đổi mật khẩu với xác thực bổ sung
POST	/api/Admin/Employee/login	Đăng nhập vào hệ thống
POST	/api/Admin/Employee/login/secure	Đăng nhập với xác thực bổ sung
POST	/api/Admin/Employee/logout	Đăng xuất khỏi hệ thống
POST	/api/Admin/Employee/register	Đăng ký tài khoản nhân viên mới và chưa có xác thực
POST	/api/Admin/Employee/register/secure	Đăng ký tài khoản với xác thực bổ sung
POST	/api/admin/Import/post	Tạo một phiếu nhập kho mới
POST	/api/admin/Import/create/secure	Tạo phiếu nhập kho với xác thực bổ sung
POST	/api/admin/Export/create	Tạo một phiếu xuất kho mới
POST	/api/admin/Export/create/secure	Tạo phiếu xuất kho với xác thực bổ sung
POST	/api/Admin/Order	Tạo một đơn hàng mới
POST	/api/Admin/Order/{orderId}/cancel	Hủy một đơn hàng theo ID
POST	/api/Admin/Partrequest/{requestId}/accept	Chấp nhận một yêu cầu linh kiện theo ID
POST	/api/Admin/Partrequest/{requestId}/deny	Từ chối một yêu cầu linh kiện theo ID

Phương thức	Đường dẫn API	Mô tả
POST	/api/Admin/Partrequest/post	Tạo một yêu cầu linh kiện mới
POST	/api/Admin/Profile	Tạo một profile phân quyền mới
POST	/api/Admin/Profile/assign	Gán một hoặc nhiều profile cho người dùng
POST	/api/Admin/Role/{roleName}/create	Tạo một vai trò mới trong hệ thống
POST	/api/Admin/Role/assignrole	Gán một vai trò cho người dùng
POST	/api/Admin/Role/revokerole	Thu hồi một vai trò khỏi người dùng
PUT	/api/Admin/Profile/{profileName}	Cập nhật thông tin một hồ sơ phân quyền
DELETE	/api/Admin/Profile/{profileName}	Xóa một hồ sơ phân quyền theo tên
DELETE	/api/Admin/Role/{roleName}	Xóa một vai trò theo tên

CHƯƠNG 4: CÀI ĐẶT HỆ THỐNG

4.1. Thông số cài đặt môi trường

4.1.1. Môi trường Máy chủ (Server)

- Nhóm thực hiện triển khai hệ thống trên nền tảng hệ điều hành **Oracle Linux 8** để đảm bảo tính tương thích tốt nhất với cơ sở dữ liệu Oracle 21c. Cấu hình máy chủ ảo hóa (VMWare) như sau:

Bảng 4.1: Thông số kỹ thuật của máy chủ.

Thành phần	Thông số cấu hình	Giải trình kỹ thuật

CPU	4 vCores	Đảm bảo khả năng xử lý đa luồng cho các tiến trình nền (Background Processes) của Oracle.
RAM	8 GB	Đáp ứng yêu cầu tối thiểu của Oracle 21c Enterprise Edition.
Storage	50GB (NVMe)	Phân vùng theo chuẩn LVM (Logical Volume Manager) để linh hoạt trong việc mở rộng dung lượng lưu trữ sau này.
Network	NAT Mode	Sử dụng cơ chế NAT để máy chủ có thể truy cập Internet tải gói tin, đồng thời giao tiếp được với máy Host.

4.1.2. Cơ sở dữ liệu Oracle 21c

Hệ thống sử dụng phiên bản cơ sở dữ liệu Oracle 21c Enterprise Edition. Đây là phiên bản Innovation Release mang lại các tính năng mới nhất về bảo mật và quản lý dữ liệu đa mô hình.

Quá trình cài đặt và cấu hình Database Instance tuân thủ kiến trúc Multitenant (Kiến trúc đa thuê bao), bao gồm một Container Database (CDB) quản lý chung và một Pluggable Database (PDB) chứa dữ liệu nghiệp vụ riêng biệt.

Các tham số môi trường (Environment Variables) quan trọng được thiết lập trong file .bash_profile của người dùng Oracle như sau:

Bảng 4.2: Cấu hình biến môi trường Oracle.

Biến môi trường	Giá trị thiết lập	Mô tả chức năng
ORACLE_BASE	/opt/Oracle	Thư mục gốc chứa các phần mềm và cấu hình của Oracle.
ORACLE_HOME	/opt/Oracle/product/21c/db_home_1	Thư mục chứa các file nhị phân (binary) và thư viện thực thi.
ORACLE_SID	ORCLCDB	Định danh duy nhất (System Identifier) của Container Database.

PDB_NAME	ORCLPDB1	Tên của Pluggable Database chứa dữ liệu ứng dụng Mobile Service.
NLS_LANG	AMERICAN_AMERICA. AL32UTF8	Thiết lập bộ mã ký tự Unicode để hỗ trợ lưu trữ tiếng Việt đầy đủ.

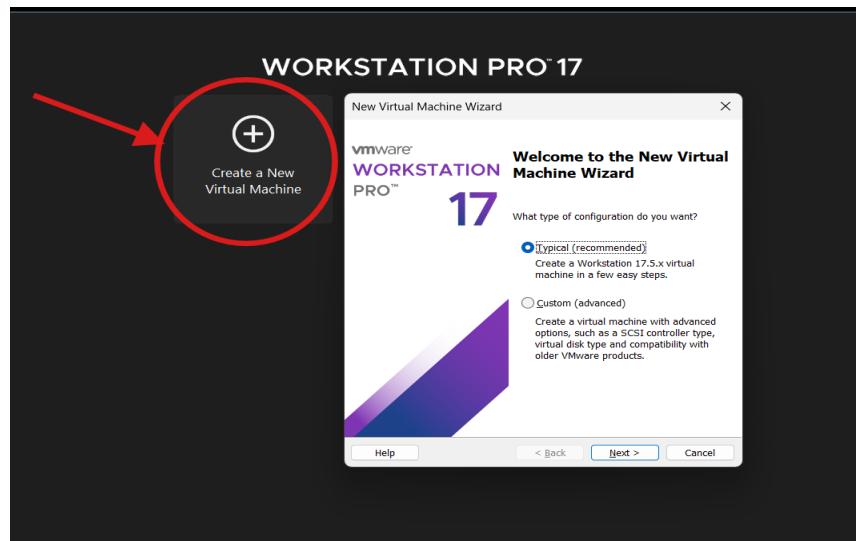
Lệnh cấu hình biến môi trường:

```
echo "export ORACLE_HOME=/opt/Oracle/product/21c/dbhome_1">>>~/.bash_profile
echo "export PATH=$ORACLE_HOME/bin:$PATH">>>~/.bash_profile
echo "export ORACLE_SID=ORCLCDB">>>~/.bash_profile source ~/.bash_profile
```

Các câu lệnh trên giúp thiết lập biến môi trường cho cơ sở dữ liệu Oracle trên Linux, giúp hệ thống và người dùng có thể dễ dàng sử dụng các công cụ và câu lệnh của Oracle.

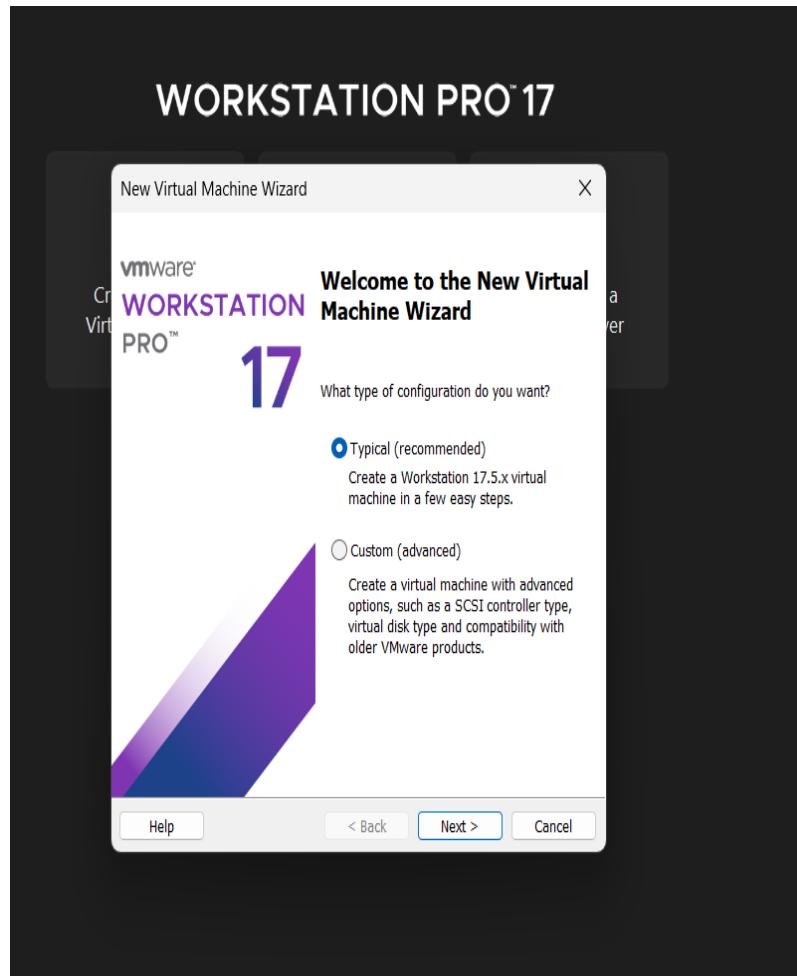
4.2. Triển khai máy chủ trên nền tảng ảo hóa

- Tải iso của Oracle Linux 8 trên website của Oracle
- Đường dẫn: <https://yum.oracle.com/oracle-linux-downloads.html>
- Trong VMWare Workstation Pro, click chọn Create a New Virtual Machine để tạo máy ảo mới



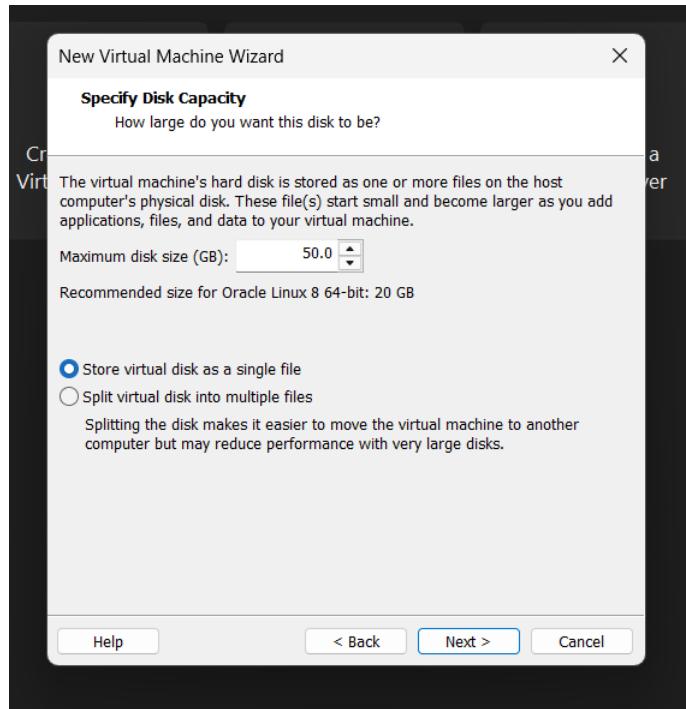
Hình 4.1: Tạo máy ảo.

- + Cửa sổ tạo máy ảo sẽ xuất hiện, tùy theo nhu cầu có thể chọn 2 lựa chọn là Typical hoặc Custom. Ở đây ta sẽ chọn Typical để tạo nhanh



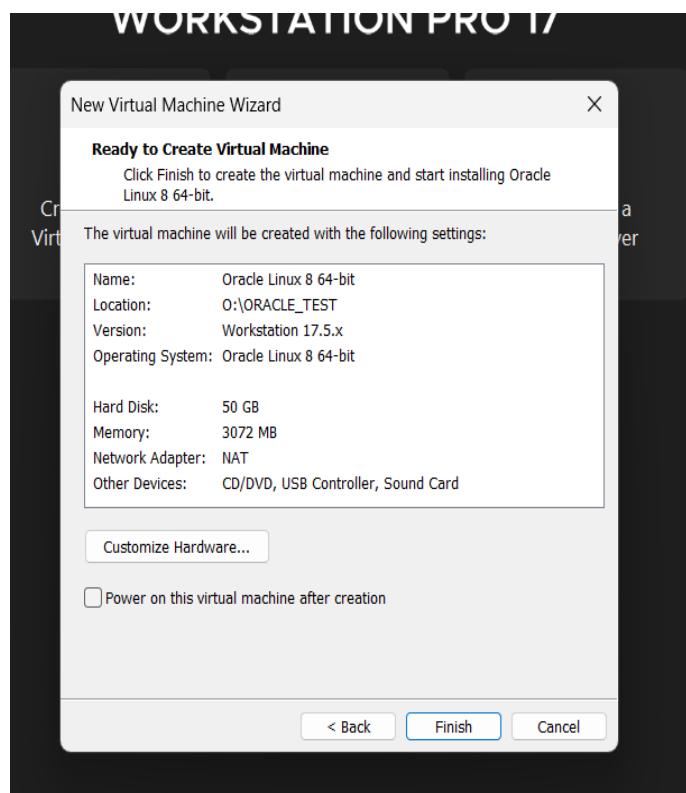
Hình 4.2: Tạo nhanh bằng cách chọn Typical

Cho máy ảo dung lượng tối thiểu 50 GB



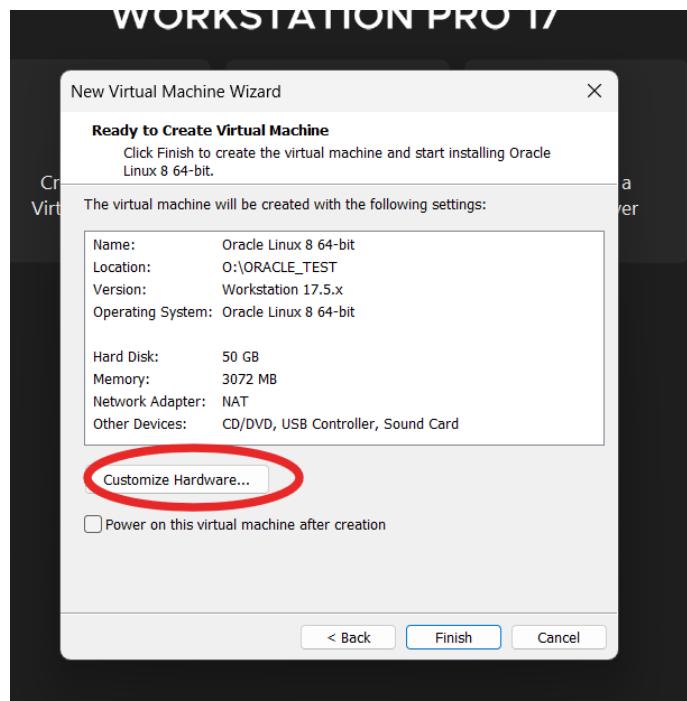
Hình 4.3: Dung lượng máy ảo tối thiểu 50GB.

- + Thông số chi tiết của máy ảo của máy sẽ xuất hiện



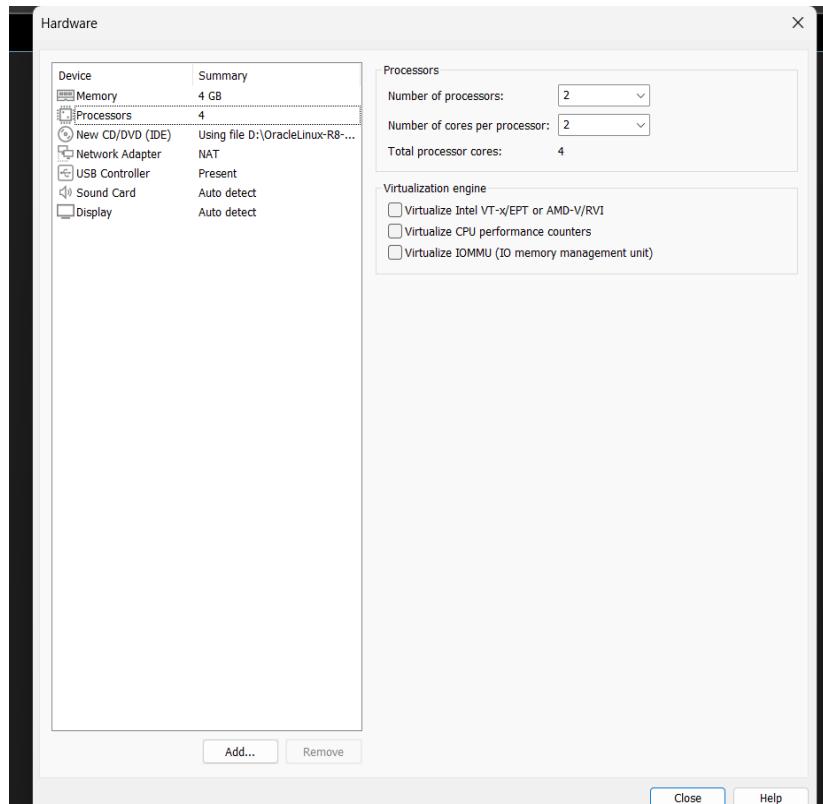
Hình 4.4: Thông số chi tiết của máy chủ.

- + Chọn Customize để cấu hình chi tiết cho máy



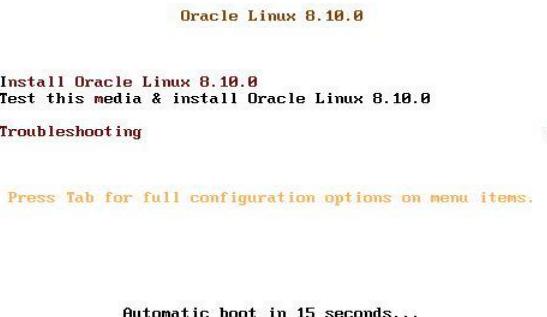
Hình 4.5: Customize để cấu hình chi tiết cho máy.

- + Cấu hình cho máy tối thiểu 4 processors core và 4 GB Memory



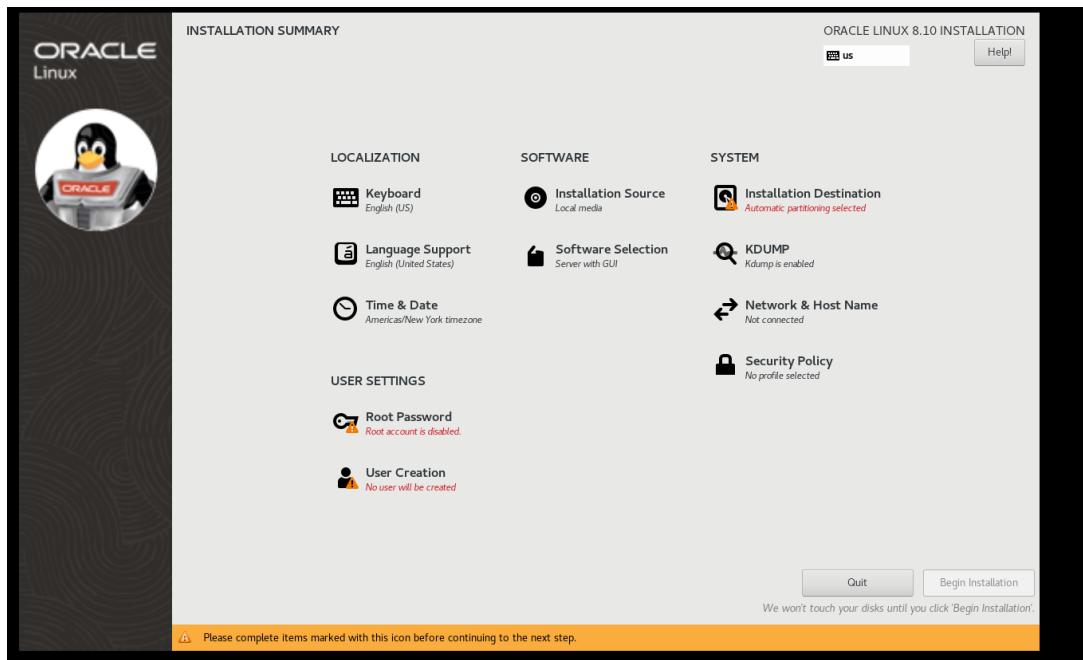
Hình 4.6: Cấu hình processors core và memory

- + Mở máy ảo tiến hành cài đặt
- + Chọn Install Oracle Linux 8.10.0 để tiến hành cài đặt



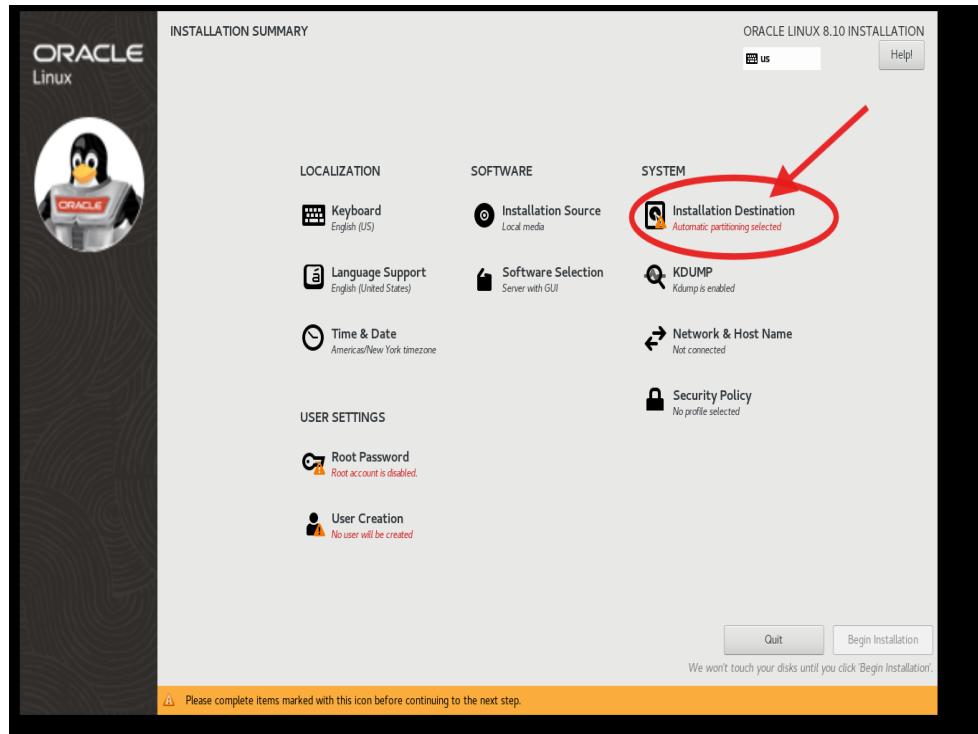
Hình 4.7: Màn hình boot Oracle Linux.

- + Sau khi boot hoàn tất, đợi cho máy ảo bắt đầu cài đặt
- + Sau khi cài đặt hoàn tất, màn hình sẽ hiển thị trang cài đặt các bước cuối cùng
- + Các bước thiết lập cuối cùng



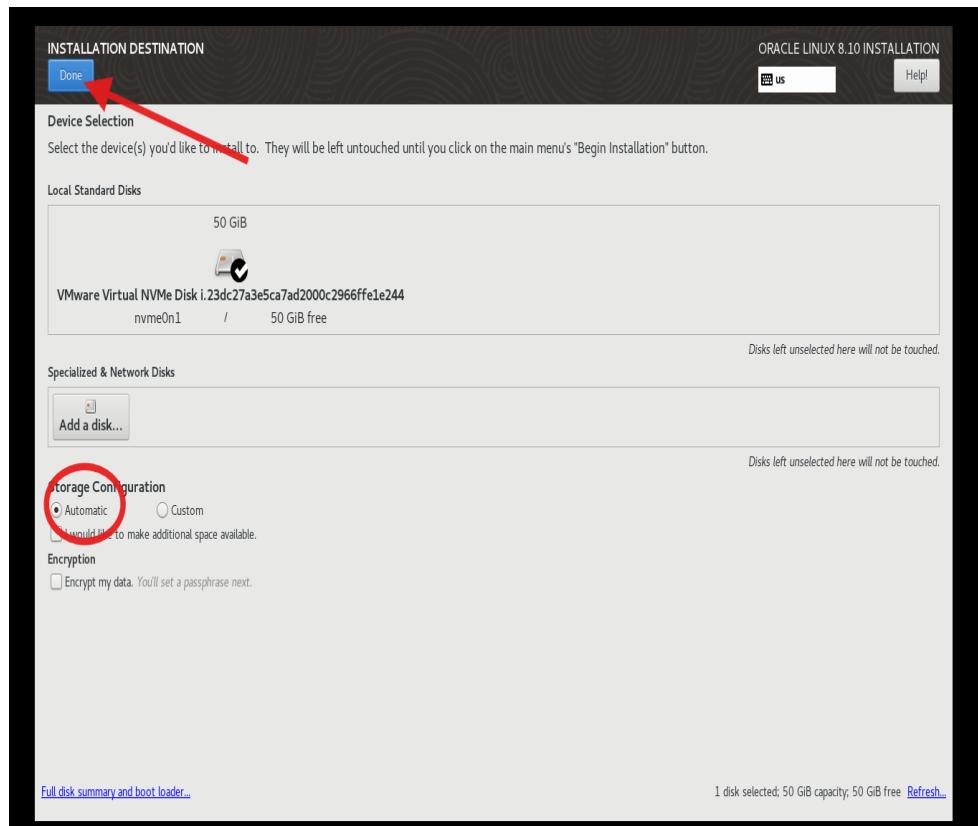
Hình 4.8: Thiết lập cài đặt cuối cùng.

- + Máy ảo cần được phân vùng bộ nhớ trước khi được cài đặt, Click vào Installation Destination để thiết lập phân vùng.



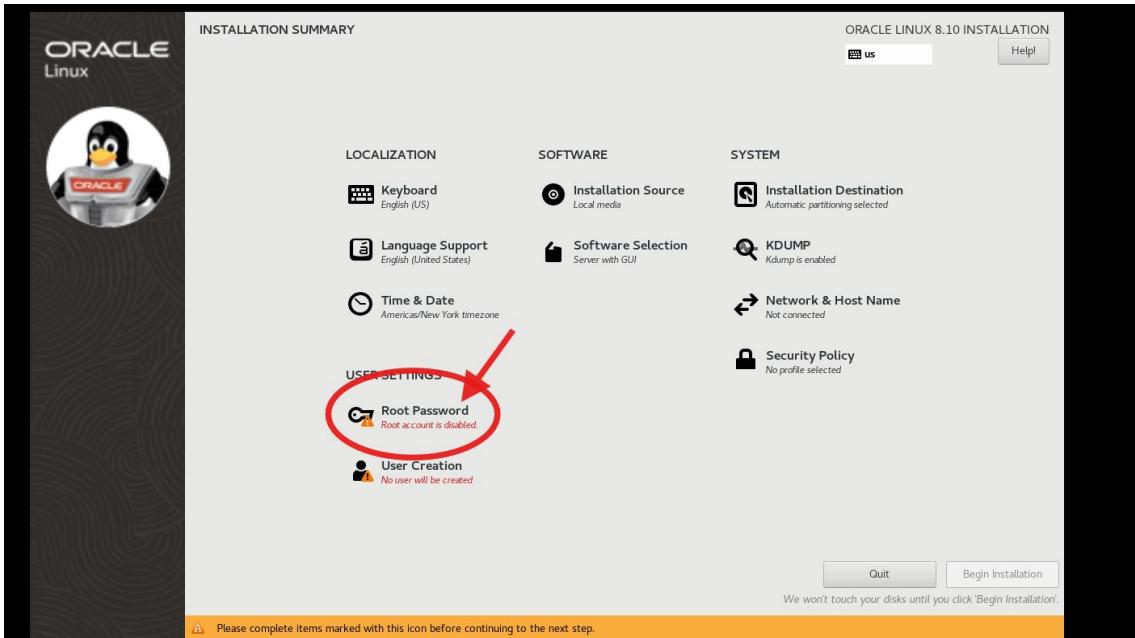
Hình 4.9: Phân vùng bộ nhớ.

- + Chọn Automatic để tự động phân vùng và click done để hoàn thành



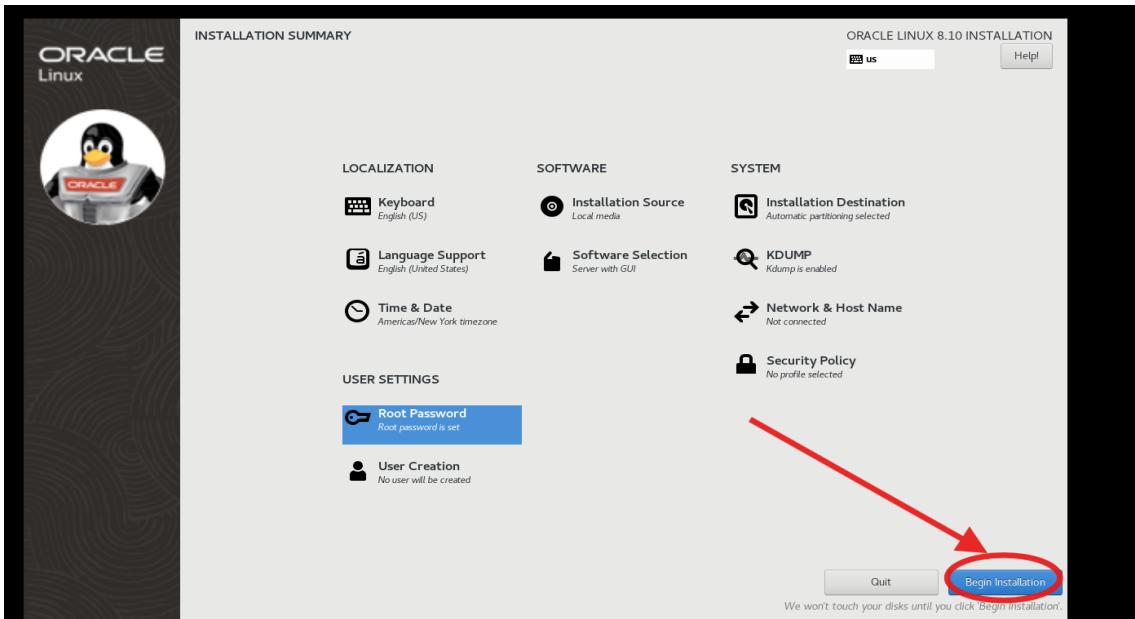
Hình 4.10: Hoàn tất phân vùng bộ nhớ.

- + Tạo mật khẩu cho Root là yêu cầu bắt buộc khi cài đặt linux,



Hình 4.11: Cài đặt mật khẩu root.

- + Sau khi hoàn tất các thiết lập, tiến hành cài đặt, click vào Begin Installation



Hình 4.12: Hoàn tất cài đặt mật khẩu cho root.

- + Trước khi reboot cần phải đồng ý các điều khoản có trong License



Hình 4.13: Đồng ý điều khoản sử dụng để tiến hành khởi động lại.

- + Chọn Done để hoàn tất

4.3. Triển khai cơ sở dữ liệu Oracle 21c

- Trước khi bắt đầu cài đặt database, cần phải cập nhật hệ thống

A screenshot of a terminal window titled "root@localhost:~". The window shows a command-line interface with the following text:

```
File Edit View Search Terminal Help
[tqc@localhost ~]$ su -
Password:
[root@localhost ~]# sudo dnf update -y
```

The terminal is running on a desktop environment with a pink and orange background.

Hình 4.14: Lệnh cập nhập hệ thống.

- + Sau khi cập nhập thành công thực hiện tiếp lệnh sudo dnf install -y oracle-epel-release-el8 để cài gói meta-package do Oracle cung cấp để bật **EPEL (Extra Packages for Enterprise Linux)** repository trên hệ thống Oracle Linux 8.

```
[root@192 ~]# sudo dnf install -y oracle-epel-release-el8
Waiting for process with pid 3582 to finish.
Last metadata expiration check: 0:00:59 ago on Tue 30 Sep 2025 06:23:31 PM EDT.
Dependencies resolved.

=====
Package           Arch    Version      Repository      Size
=====
Installing:
oracle-epel-release-el8 x86_64  1.0-5.el8        ol8_baseos_latest 15 k
Installing dependencies:
yum-utils          noarch   4.0.21-25.0.1.el8  ol8_baseos_latest 75 k

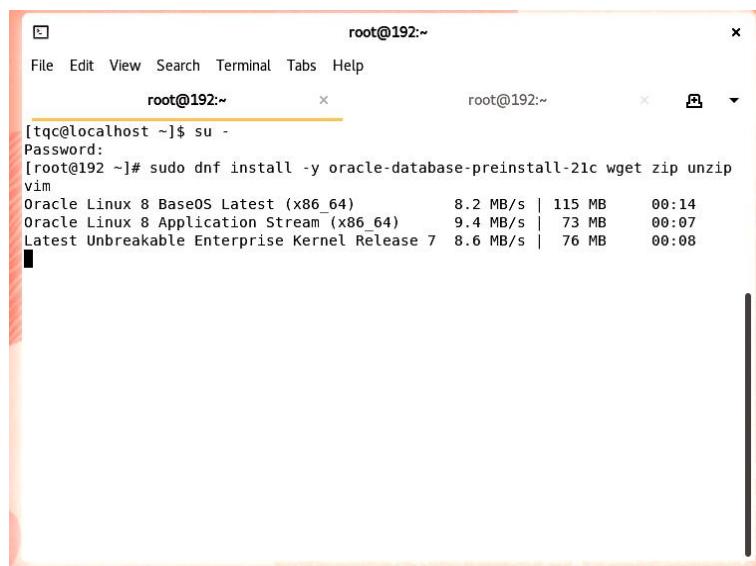
Transaction Summary
=====
Install 2 Packages

Total download size: 90 k
Installed size: 41 k
Downloading Packages:
(1/2): oracle-epel-release-el8-1.0-5.el8.x86_64 54 kB/s | 15 kB  00:00
(2/2): yum-utils-4.0.21-25.0.1.el8.noarch.rpm 235 kB/s | 75 kB  00:00

```

Hình 4.15: Cài gói meta-package của Oracle.

- + Sau khi cài đặt gói meta-package, thực hiện cài đặt oracle-database-preinstall-21c, giúp tạo sẵn môi trường hệ thống để cài **Oracle Database 21c**



Hình 4.16: Cài đặt preinstall tạo sẵn môi trường hệ thống.

- + Sau khi thực hiện các bước thiết lập trên, tiến hành cài đặt Oracle database 21c qua đường dẫn

<https://www.oracle.com/database/technologies/oracle21c-linux-downloads.html>

Hình 4.17: Trang tải gói Oracle database 21c.

- Sau khi tải về máy thành công, tiến hành cài đặt Oracle database 21c
 - + Gõ lệnh sudo dnf localinstall oracle-database-ee-21c-1.0-1.ol8.x86_64.rpm -y

```
[tqc@192 Downloads]$ sudo dnf localinstall oracle-database-ee-21c-1.0-1.ol8.x86_64.rpm -y
[sudo] password for tqc:
Last metadata expiration check: -1 day, 17:38:02 ago on Tue 30 Sep 2025 06:23:31 PM EDT.
Dependencies resolved.
=====
Transaction Summary
=====
Install 1 Package

Total size: 2.6 G
Installed size: 7.1 G
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
=====

```

Hình 4.18: Cài đặt Oracle database 21c.

– Cài đặt thành công

```
tqc@localhost:~/Downloads
S  File Edit View Search Terminal Help
Running transaction check
H Transaction check succeeded.
Running transaction test
D Transaction test succeeded.
Running transaction
D  Preparing      :
M  Running scriptlet: oracle-database-ee-21c-1.0-1.x86_64
P  Installing      : oracle-database-ee-21c-1.0-1.x86_64 [== 1/1] 1/1
V  Installing      : oracle-database-ee-21c-1.0-1.x86_64
Running scriptlet: oracle-database-ee-21c-1.0-1.x86_64
[INFO] Executing post installation scripts...
T [INFO] Oracle home installed successfully and ready to be configured.
To configure a sample Oracle Database you can execute the following service configuration script as root: /etc/init.d/oracledb_ORCLCDB-21c configure
O  Verifying      : oracle-database-ee-21c-1.0-1.x86_64
O Installed:
  oracle-database-ee-21c-1.0-1.x86_64
Complete!
[tqc@192 Downloads]$
```

Hình 4.19: Cài đặt thành công Oracle Database 21c.

- Sau khi cài đặt thành công, chạy lệnh sudo /etc/init.d/oracledb_ORCLCDB-21c configure
- Lệnh sẽ tạo:
 - + Tạo một pluggable database tên là ORCLPDB1
 - + Bật chế độ tự động khởi động (auto startup)
 - + Đặt mật khẩu mặc định cho các người dùng như SYS, SYSTEM, v.v. là Oracle_4U

```
[tqc@192 Downloads]$ sudo /etc/init.d/oracledb_ORCLCDB-21c configure
[sudo] password for tqc:
Configuring Oracle Database ORCLCDB.
Prepare for db operation
8% complete
Copying database files
31% complete
Creating and starting Oracle instance
32% complete
36% complete
40% complete
43% complete
46% complete
Completing Database Creation
51% complete
54% complete
Creating Pluggable Databases
58% complete
77% complete
Executing Post Configuration Actions
100% complete
Database creation complete. For details check the logfiles at:
/opt/oracle/cfgtoollogs/dbca/ORCLCDB.
Database Information:
```

Hình 4.20: Lệnh để tạo sẵn thiết lập cơ bản

- Tiếp theo, thiết lập môi trường cho người dùng Oracle
 - + đăng nhập root sau đó đăng nhập người dùng Oracle

tqc@localhost:~/Downloads

```
[tqc@192 Downloads]$ su -  
Password:  
[root@192 ~]# su - oracle  
[oracle@192 ~]$
```

- + Nhập tuần tự các lệnh sau

```
echo"export  
ORACLE_HOME=/opt/oracle/product/21c/dbhome_1">>~/.bash_profile  
  
echo "export PATH=\$ORACLE_HOME/bin:\$PATH" >> ~/.bash_profile  
  
echo"exportORACLE_SID=ORCLCDB">>~/.bash_profilesource~/.bash_p  
rofile
```

- + Tiến hành mở port 1521 để thực hiện kết nối từ xa

```
[root@192 ~]# sudo firewall-cmd --add-port=1521/tcp --permanent  
success  
[root@192 ~]# sudo firewall-cmd --reload  
success
```

Hình 4.21: Mở cổng 1521 trên tường lửa.

- Cho phép kết nối từ xa trong Listener
- Dùng lệnh sudo vi
\$ORACLE_HOME/network/admin/listener.ora để kiểm tra
listener lắng nghe ở port 1521
 - + Thêm các dòng sau để đảm bảo listener lắng nghe
trên port 1521

```
LISTENER =(DESCRIPTION_LIST = (DESCRIPTION =  
(ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT =  
1521))))
```

- + sau đó dừng listener và khởi động lại

```
lsnrctl stop
```

```
lsnrctl start
```

```
[oracle@192 ~]$ lsnrctl start
LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 30-SEP-2025 13:17:58
Copyright (c) 1991, 2021, Oracle. All rights reserved.

Starting /opt/oracle/product/21c/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 21.0.0.0.0 - Production
System parameter file is /opt/oracle/homes/OraDBHome21cEE/network/admin/listener.ora
Log messages written to /opt/oracle/diag/tnslsnr/192/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.38.136)(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.38.136)(PORT=1521)))
STATUS of the LISTENER
-----
Alias                      LISTENER
Version                   TNSLSNR for Linux: Version 21.0.0.0.0 - Production
Start Date                30-SEP-2025 13:17:58
Uptime                     0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /opt/oracle/homes/OraDBHome21cEE/network/admin/listener.ora
Listener Log File         /opt/oracle/diag/tnslsnr/192/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.38.136)(PORT=1521))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
The listener supports no services
The command completed successfully
```

Hình 4.22: Khởi động listener.

- Tiếp theo, tạo user để truy cập từ xa
 - + Đăng nhập vào SQL plus
 - + Hiển thị danh sách tất cả các Pluggable Databases (PDBs)
 - + Hiển thị tên Container hiện tại đang kết nối tới

```
Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> show pdbs;
    CON_ID CON_NAME           OPEN MODE  RESTRICTED
----- -----
      2 PDB$SEED            READ ONLY  NO
      3 ORCLPDB1            READ WRITE NO
SQL>
```

Hình 4.23: Hiển thị tên Container hiện tại đang kết nối.

- + kiểm tra trạng thái hiện tại của Oracle instance đang kết nối

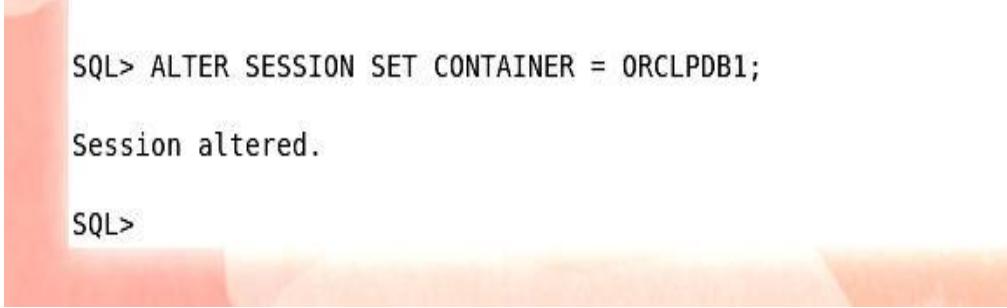
```
CDB$ROOT
SQL> SELECT status FROM v$instance;

STATUS
-----
OPEN

SQL>
```

Hình 4.24: tra trạng thái hiện tại của Oracle instance.

- + Chuyển phiên làm việc (session) hiện tại sang Pluggable Database (PDB) tên là ORCLPDB1.



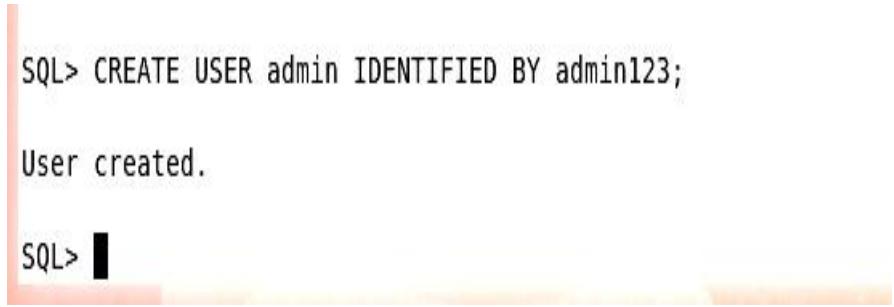
```
SQL> ALTER SESSION SET CONTAINER = ORCLPDB1;
Session altered.
```

SQL>

Hình 4.25: Chuyển phiên làm việc (session).

Để kết nối với database:

- Tạo một người dùng mới tên admin với mật khẩu là admin123 để làm ví dụ.

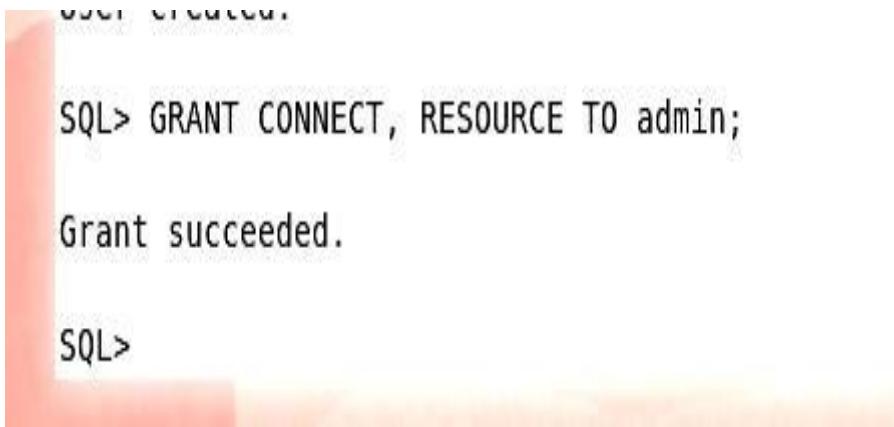


```
SQL> CREATE USER admin IDENTIFIED BY admin123;
User created.
```

SQL>

Hình 4.26: Tạo người dùng admin

- Cấp quyền cơ bản cho người dùng
 - + CONNECT: cho phép người dùng kết nối vào database.
 - + RESOURCE: cấp các quyền để người dùng có thể tạo các đối tượng như table, view, procedure,...

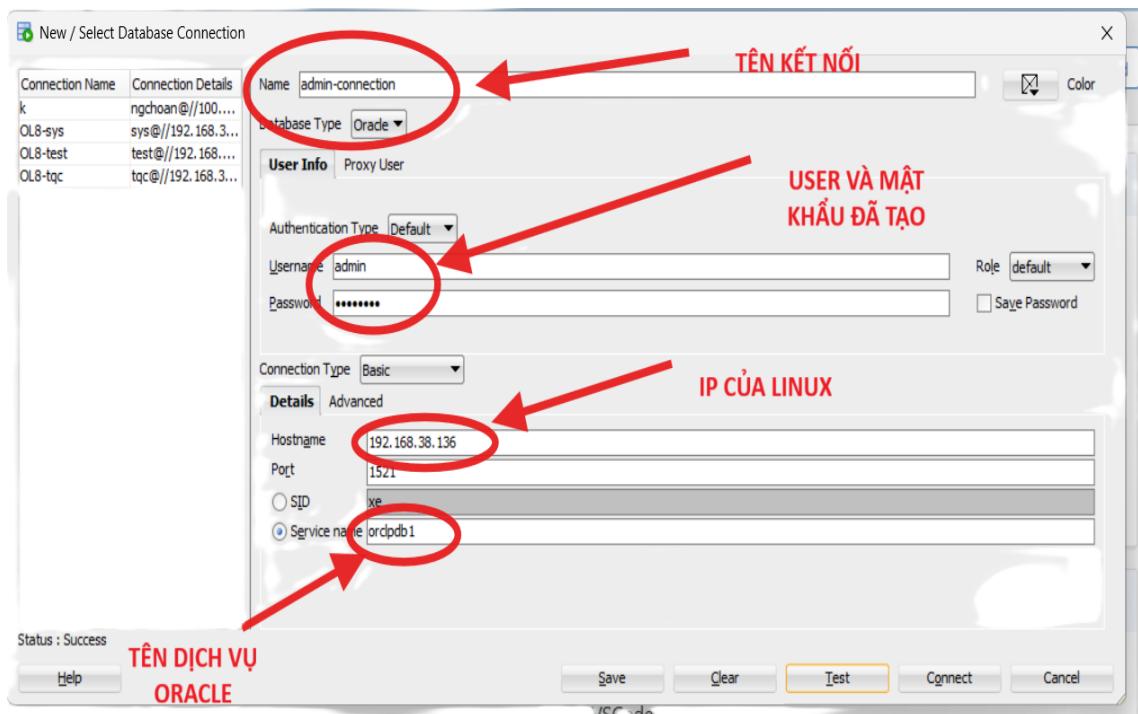


```
SQL> GRANT CONNECT, RESOURCE TO admin;
Grant succeeded.
```

SQL>

Hình 4.27: Cấp quyền cơ bản cho người dùng.

Kiểm thử trên SQL Developer kết nối với database.



Hình 4.28: Kết nối thành công.

Status hiển thị Success là cài đặt thành công.

4.4. Triển khai VPN ZeroTier

Do hệ thống được triển khai trên máy chủ ảo nội bộ (localhost) không có địa chỉ IP Public tĩnh, nhóm nghiên cứu đã tích hợp giải pháp mạng định nghĩa bằng phần mềm VPN ZeroTier để cho phép truy cập từ xa an toàn.

ZeroTier tạo ra một lớp mạng ảo (Virtual Network Interface) hoạt động ở Lớp 2 (Data Link Layer) của mô hình OSI. Mọi dữ liệu truyền tải giữa Client (Máy tính quản lý/Điện thoại) và Server (cơ sở dữ liệu Oracle) đều được mã hóa đầu cuối (End-to-End Encryption) sử dụng thuật toán 256-bit Salsa20.

4.5. Định danh, xác thực và profile

4.5.1. Định danh và xác thực

4.5.1.1 Giới thiệu chung

Hệ thống dùng cơ sở dữ liệu Oracle làm kho định danh duy nhất. Mỗi nhân viên/khách hàng có tài khoản người dùng Oracle riêng. Định danh và xác thực đảm bảo mọi tác nhân (nhân viên, khách hàng, dịch vụ nền) được nhận dạng chính xác và chỉ được cấp quyền sau khi vượt qua xác thực. Phạm vi áp dụng: WebApp nội bộ, WebAPI trung gian, ứng dụng di động khách hàng, dùng chung hạ tầng Oracle. Người dùng cần phiên làm việc Oracle đang mở và JWT Token ứng dụng tương ứng.

```
1 reference
public OracleConnection CreateConnection(string username, string password, string platform
, string sessionId, bool proxy = false)
{
    RemoveAllConnections(username, platform).Wait();
    var oracleUsername = proxy
        ? BuildProxyUsername(username)
        : username;
    var connectionStringTemplate = _configuration.GetConnectionString("OracleDb");
    var connString = connectionStringTemplate
        .Replace("{username}", oracleUsername)
        .Replace("{password}", password);
    var conn = new OracleConnection(connString);
    OracleConnection.ClearPool(conn);
    conn.Open();
    using (var cmd = conn.CreateCommand())
    {
        cmd.CommandText = "BEGIN DBMS_APPLICATION_INFO.SET_MODULE(:module_name, :client_info); END;";
        cmd.Parameters.Add(new OracleParameter("module_name", $"WebAPI-{platform}"));
        cmd.Parameters.Add(new OracleParameter("client_info", sessionId));
        cmd.ExecuteNonQuery();
    }
    using (var cmd = conn.CreateCommand())
    {
        cmd.CommandText = "BEGIN DBMS_SESSION.SET_IDENTIFIER(:sid); END;";
        cmd.Parameters.Add(new OracleParameter("sid", sessionId));
        cmd.ExecuteNonQuery();
    }
    string oracleSid;
    using (var cmd = conn.CreateCommand())
    {
        cmd.CommandText = "SELECT SYS_CONTEXT('USERENV', 'SID') FROM DUAL";
        oracleSid = cmd.ExecuteScalar()?.ToString() ?? "";
    }
    var key = (username, platform, sessionId);
    _connections[key] = new OracleConnInfo(conn, oracleSid);
    return conn;
}
1 reference
private string BuildProxyUsername(string targetUsername)
{
    var proxyUser = _configuration["QrLogin:ProxyUser"]
        ?? _configuration.GetConnectionString("DefaultUsername")
        ?? "APP";
    return $"{proxyUser}{targetUsername}";
}
```

Hình 4.29: Mã nguồn đăng nhập và định danh người dùng

```

public bool TrySetHeaders(HttpClient httpClient, out IActionResult redirectToLogin, bool isAdmin = true)
{
    redirectToLogin = null;
    var httpContext = _httpContextAccessor.HttpContext;
    // Prefix key: "" cho Admin, "C" cho Public
    string prefix = isAdmin ? "" : "C";
    var token = httpContext.Session.GetString($"{prefix}JwtToken");
    var username = httpContext.Session.GetString($"{prefix}Username");
    var platform = httpContext.Session.GetString($"{prefix}Platform");
    var sessionId = httpContext.Session.GetString($"{prefix}SessionId");
    if (string.IsNullOrEmpty(token) ||
        string.IsNullOrEmpty(username) ||
        string.IsNullOrEmpty(platform) ||
        string.IsNullOrEmpty(sessionId))
    {
        // Redirect về Login tương ứng
        if (isAdmin)
            redirectToLogin = new RedirectToActionResult("Login", "Employee", new { area = "Admin" });
        else
            redirectToLogin = new RedirectToActionResult("Login", "Customer", new { area = "Public" });
        return false;
    }
    var testToken = token;
    httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", testToken);

    httpClient.DefaultRequestHeaders.Remove("X-Oracle-Username");
    httpClient.DefaultRequestHeaders.Remove("X-Oracle-Platform");
    httpClient.DefaultRequestHeaders.Remove("X-Oracle-SessionId");

    httpClient.DefaultRequestHeaders.Add("X-Oracle-Username", username);
    httpClient.DefaultRequestHeaders.Add("X-Oracle-Platform", platform);
    httpClient.DefaultRequestHeaders.Add("X-Oracle-SessionId", sessionId);
    return true;
}

```

Hình 4.30: Mã nguồn xác thực gán ở header ở client trước khi gọi API.

4.5.1.2 Singleton Class OracleConnectionManager

- Cấu trúc dữ liệu và khóa định danh

Thông tin người dùng sau khi đăng nhập sẽ được lưu trữ trong một từ điển

```

2 references
private record OracleConnInfo(OracleConnection Conn, string OracleSid);
// Key: (username, platform, sessionId)
private readonly ConcurrentDictionary<(string username, string platform, string sessionId)
    , OracleConnInfo> _connections = new();
10 references

```

Hình 4.31: Mã nguồn từ điển lưu trữ thông tin người dùng.

Trong đó:

- Username là tên đăng nhập người dùng Oracle tương ứng (nhân viên dùng tên đăng nhập nội bộ, khách hàng thì dùng số điện thoại).
- Platform: kênh truy cập bao gồm MOBILE, WEB.
- SessionId: GUID do ứng dụng tạo và được set vào Oracle bằng DBMS.SESSION.SET_IDENTIFIER.
- OracleConnInfo: giữ OracleConnection và OracleSid để tiện ghi log/kill phiên làm việc.

- Nhờ khía 3 thành phần, hệ thống cho phép cùng một người dùng đăng nhập trên nhiều nền tảng, nhưng chỉ 1 phiên / platform.

Quy trình đăng nhập: (Create connection)

- + Khi người dùng nhập username và password, thì thực thi thủ tục LOGIN_EMPLOYEE/LOGIN_CUSTOMER.
- + RemoveAllConnections(username, platform): Để đảm bảo không còn phiên cũ trên cùng nền tảng, đồng thời gửi SignalR ForceLogout đến các client đang dùng session cũ.
- + Từ template và credential nhận được thì mở OracleConnection.
- + Ghi nhận platform và session vào Oracle thông qua SET_MODULE và SET_IDENTIFIER.
- + Lấy SID thật bằng SELECT SYS_CONTEXT('USERENV','SID') FROM DUAL.
- + Ghi thông tin vào dictionary với khóa(username, platform, sessionid). Lúc này coi như là login thành công.
- + Phát JWT Token đến client.
- + Sau đó người dùng tải lên Private Key của bản thân để giúp mã hóa api RSA/AES và các dịch vụ khác.

```
public string GenerateToken(string username, string roles, string sessionId)
{
    var jwtKey = _config["Jwt:Key"];
    var jwtIssuer = _config["Jwt:Issuer"];
    var jwtAudience = _config["Jwt:Audience"];
    var jwtExpiresMinutes = Convert.ToDouble(_config["Jwt:ExpiresInMinutes"]);

    var claims = new[]
    {
        new Claim(ClaimTypes.Name, username),
        new Claim(ClaimTypes.Role, roles ?? ""),
        new Claim("sessionId", sessionId ?? "") // ✓ thêm sessionId
    };

    var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(jwtKey));
    var creds = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);

    var token = new JwtSecurityToken(
        issuer: jwtIssuer,
        audience: jwtAudience,
        claims: claims,
        expires: DateTime.UtcNow.AddMinutes(jwtExpiresMinutes),
        signingCredentials: creds
    );

    return new JwtSecurityTokenHandler().WriteToken(token);
}
```

Hình 4.32: Mã nguồn tạo JWT Token cho client.

Quy trình gọi API

Mỗi request [Authorize] phải gửi:

- Header JWT (Authorization: Bearer ...),
- X-Oracle-Username, X-Oracle-Platform, X-Oracle-SessionId.
- OracleSessionHelper đọc các header trên và gọi GetConnectionIfExists.
- GetConnectionIfExists sẽ:
 - + Tra dictionary theo khóa 3 thành phần: username, platform, sessionid.
 - + Nếu tìm thấy, chạy SELECT 1 FROM DUAL để chắc chắn session Oracle còn mở.
 - + Nếu query thành công → trả lại OracleConnection để stored procedure nghiệp vụ sử dụng.
 - + Nếu thất bại (timeout, killed) → tự động RemoveConnection và trả null, khiến API phản hồi 401 “Oracle session expired”.

```
14 references
public OracleConnection GetConnectionOrUnauthorized(HttpContext httpContext,
    OracleConnectionManager connManager, out IActionResult unauthorizedResult)
{
    unauthorizedResult = null;

    var username = httpContext.Request.Headers["X-Oracle-Username"].ToString();
    var platform = httpContext.Request.Headers["X-Oracle-Platform"].ToString();
    var sessionId = httpContext.Request.Headers["X-Oracle-SessionId"].ToString();

    if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(platform) || string.IsNullOrEmpty(sessionId))
    {
        unauthorizedResult = new UnauthorizedObjectResult(new { message = "Missing Oracle session headers" });
        return null;
    }

    var conn = connManager.GetConnectionIfExists(username, platform, sessionId);
    if (conn == null)
    {
        unauthorizedResult = new UnauthorizedObjectResult(new { message = "Oracle session expired. Please login again." });
        return null;
    }

    return conn;
}
```

Hình 4.33: Mã nguồn Xác thực người dùng mới khi gọi API.

Quy trình đăng xuất (RemoveConnection)

- + Khi người dùng bấm “Đăng xuất” hoặc JWT hết hạn, WebApp/Mobile gọi endpoint logout.
- + API lấy thông tin header và gọi RemoveConnection(username, platform, sessionId).

Bên trong hàm:

- + Xóa entry khỏi dictionary.
- + Đóng connection, ClearPool, giải phóng tài nguyên.

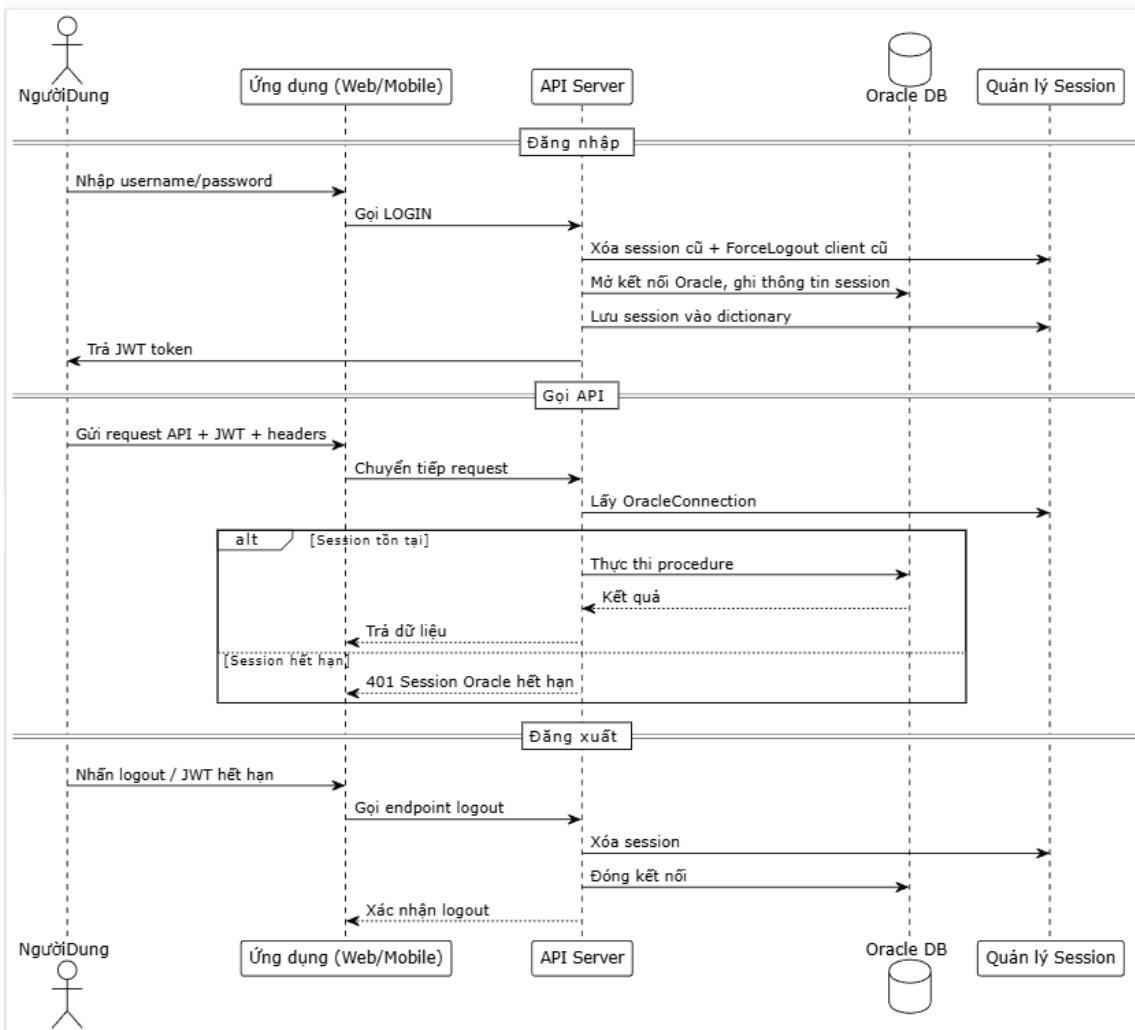
- + Thông báo SignalR ForceLogout cho nhóm sessionId để client điều hướng về trang đăng nhập.
- + Từ đó mọi request dùng sessionId này sẽ bị từ chối cho đến khi đăng nhập lại.

```

public async Task RemoveConnection(string username, string platform, string sessionId)
{
    var key = (username, platform, sessionId);
    // Dùng TryRemove để đảm bảo atomic - chỉ có 1 luồng có thể remove connection
    // Nếu connection đã bị remove rồi, return ngay
    if (!_connections.TryRemove(key, out var info))
    {
        // Connection đã được remove rồi bởi luồng khác, không cần làm gì
        _logger.LogDebug(
            "Connection already removed for user {Username}, platform {Platform}, session {SessionId}",
            username, platform, sessionId);
        return;
    }
    // Chỉ kill session nếu thực sự remove được connection từ dictionary
    // Kill Oracle session bằng procedure trước khi dispose connection
    try
    {
        using var defaultConn = CreateDefaultConnection();
        var killResult = OracleHelper.ExecuteNonQueryWithOutputs(
            defaultConn,
            "APP.KILL_SESSION_BY_CLIENT_ID",
            new[]
            {
                {"p_client_identifier", OracleDbType.Varchar2, (object)sessionId}
            },
            new[]
            {
                {"p_result", OracleDbType.Varchar2}
            });
        var resultMessage = killResult["p_result"]?.ToString() ?? "Unknown";
        _logger.LogInformation(
            "Killed Oracle session for user {Username}, platform {Platform}, session {SessionId}. Result: {Result}",
            username, platform, sessionId, resultMessage);
    }
    catch (Exception ex)
    {
        _logger.LogWarning(ex,
            "Failed to kill Oracle session for user {Username}, platform {Platform}, session {SessionId}. Error: {Error}",
            username, platform, sessionId, ex.Message);
        // Continue with disposal even if kill failed
    }
    // Dispose connection
    var conn = info.Conn;
    try
    {
        if (conn.State == ConnectionState.Open)
            conn.Close();
    }
    catch (Exception ex)
    {
        _logger.LogWarning(ex,
            "Error closing connection for user {Username}, platform {Platform}, session {SessionId}",
            username, platform, sessionId);
    }
    finally
    {
        try { OracleConnection.ClearPool(conn); } catch { }
        conn.Dispose();
        _logger.LogInformation(
            "Closed and disposed connection for user {Username}, platform {Platform}, session {SessionId}, Oracle SID: {OracleSid}",
            username, platform, sessionId, info.OracleSid);
    }
    // Send SignalR logout message to group (sessionId)
    await _hubContext.Clients.Group(sessionId).SendAsync("ForceLogout", "Your session has been logged out.");
}

```

Hình 4.34: Mã nguồn đăng xuất.



Hình 4.35: Sơ đồ quy trình đăng nhập, gọi API, đăng xuất.

4.5.2. Profile

4.5.2.1 Mục tiêu

Mục tiêu của phần này nhằm thiết lập một chính sách kiểm soát truy cập an toàn cho người dùng trong hệ thống quản trị cơ sở dữ liệu Oracle, cụ thể là thiết kế và áp dụng profile bảo mật cho các nhóm người dùng thuộc hệ thống quản lý sửa chữa và bảo hành điện thoại. Profile giúp hạn chế các thông số như: kiểm soát số lần đăng nhập sai mật khẩu, thời hạn mật khẩu, thời gian tự động ngắt session khi không hoạt động,...

Việc áp dụng profile giúp hệ thống nâng cao tính bảo mật, giảm thiểu nguy cơ xâm nhập trái phép và đảm bảo tính toàn vẹn của dữ liệu.

Profile trong Oracle là một tập hợp các giới hạn và chính sách bảo mật áp dụng cho người dùng.

4.5.2.2 Bài toán: Quản lý Profile bảo mật trong cơ sở dữ liệu Oracle

Bài toán đặt ra là quản lý Profile bảo mật trong Oracle cho hệ thống quản lý sửa chữa và bảo hành điện thoại, với các yêu cầu:

- Tạo Profile mới: Cho phép admin định nghĩa các thông số bảo mật và áp dụng cho nhóm người dùng cụ thể.
- Cập nhật Profile: Điều chỉnh các thông số khi có thay đổi yêu cầu bảo mật, ảnh hưởng đến tất cả người dùng đang sử dụng profile đó.
- Kiểm soát truy cập và bảo mật dữ liệu: Mỗi người dùng phải tuân thủ các giới hạn do profile quy định, đảm bảo an toàn và toàn vẹn thông tin.

Kết quả mong đợi:

- Mỗi nhóm người dùng có một Profile bảo mật phù hợp.
- Hệ thống áp dụng các chính sách bảo mật đồng nhất và hiệu quả.
- Admin dễ dàng quản lý, kiểm soát và điều chỉnh các profile theo nhu cầu.

4.5.2.3 Thiết kế profile

Thiết kế Profile bao gồm việc xác định các tham số bảo mật và các chính sách đi kèm:

The screenshot shows two tables side-by-side. The left table, titled 'Danh sách Profile', lists profiles with their settings. The right table, titled 'User và Profile', lists users and their assigned profiles.

Danh sách Profile					
Tên Profile	Idle Time	Connect Time	Failed Login	Lock Time	Inactive Account Time
DEFAULT	UNLIMITED	UNLIMITED	10	1	365
ORA_CIS_PROFILE	DEFAULT	DEFAULT	5	1	120
ORA_STIG_PROFILE	15	DEFAULT	3	UNLIMITED	35
TEST	DEFAULT	DEFAULT	1	UNLIMITED	DEFAULT

User và Profile	
Tên User	Profile
0332880207	DEFAULT
1	DEFAULT
2	DEFAULT
ADMIN	DEFAULT
ANONYMOUS	DEFAULT
APP	DEFAULT

Hình 4.36: Chi tiết thông số của profile.

Bảng 4.3: Bảng tham số của profile.

Tham số	Ý nghĩa	Ghi chú
IDLE_TIME	Thời gian tối đa không hoạt động trong 1 phiên (phút).	Phiên sẽ tự động bị ngắt nếu người dùng không hoạt động trong khoảng thời gian này.
CONNECT_TIME	Thời gian kết nối tối đa của 1 phiên làm việc (phút).	Giới hạn tổng thời gian kết nối để tránh sử dụng tài nguyên lâu dài.

FAILED_LOGIN_ATTEMPTS	Số lần nhập mật khẩu sai tối đa.	Nếu vượt quá số lần này, tài khoản sẽ bị khóa tạm thời.
PASSWORD_LOCK_TIME	Thời gian khóa tài khoản sau khi nhập mật khẩu sai nhiều lần.	Khoảng thời gian mà người dùng không thể đăng nhập; sau khi hết thời gian, có thể thử lại.
INACTIVE_ACCOUNT_TIME	Tự động khóa tài khoản sau khi không đăng nhập trong thời gian dài.	Áp dụng cho các tài khoản không sử dụng, giúp tăng bảo mật và giảm rủi ro truy cập trái phép.

4.5.2.4 Tạo profile

Tạo một profile mới trong hệ thống cơ sở dữ liệu giúp áp dụng các chính sách bảo mật nhất định cho người dùng. Profile xác định các giới hạn về thời gian kết nối, số lần đăng nhập sai, tuổi thọ mật khẩu, thời gian khóa tài khoản, v.v. Việc tạo profile cho phép doanh nghiệp linh hoạt áp dụng các chính sách khác nhau cho các nhóm người dùng khác nhau, như nhân viên bán hàng, quản trị viên hoặc khách vãng lai.

- Thao tác trên giao diện web:
 - + Người quản trị nhập tên profile và các thông số bảo mật.
 - + Nhấn nút Create → hệ thống hiển thị bảng xác nhận chi tiết các thông số.
 - + Hệ thống lưu profile mới và sẵn sàng gán cho các người dùng.

The screenshot shows a modal window titled "Tạo Profile mới". It contains six input fields arranged in two columns. The first column has "Tên Profile" (Name) with placeholder "Nhập tên profile", "Idle Time" with value "UNLIMITED", "Connect Time" with value "UNLIMITED", and "Failed Login Attempts" with value "UNLIMITED". The second column has "Password Lock Time" with value "UNLIMITED" and "Inactive Account Time" with value "UNLIMITED". At the bottom is a blue "Tạo Profile" (Create Profile) button.

Hình 4.37: Tạo profile mới

Thao tác bằng lệnh SQL:

```
create or replace PROCEDURE      "CREATE_PROFILE" (
    p_profile_name IN VARCHAR2,
    p_idle_time IN VARCHAR2 DEFAULT 'UNLIMITED',
    p_connect_time IN VARCHAR2 DEFAULT 'UNLIMITED',
    p_failed_login IN VARCHAR2 DEFAULT 'UNLIMITED',
    p_lock_time IN VARCHAR2 DEFAULT 'UNLIMITED',
    p_inactive_account_time IN VARCHAR2 DEFAULT 'UNLIMITED'
)
AS
BEGIN
    EXECUTE IMMEDIATE 'CREATE PROFILE ' || UPPER(p_profile_name) || ' LIMIT
        IDLE_TIME ' || p_idle_time || '
        CONNECT_TIME ' || p_connect_time || '
        FAILED_LOGIN_ATTEMPTS ' || p_failed_login || '
        PASSWORD_LOCK_TIME ' || p_lock_time || '
        INACTIVE_ACCOUNT_TIME ' || p_inactive_account_time;
END CREATE_PROFILE;
```

Hình 4.38: Thao tác bằng lệnh SQL của tạo profile.

4.5.2.5 Cập nhật profile

Cập nhật profile giúp điều chỉnh các chính sách bảo mật theo yêu cầu thực tế, ví dụ tăng số lần đăng nhập sai, kéo dài thời gian idle hoặc thay đổi tuổi thọ mật khẩu. Việc cập nhật profile ảnh hưởng tất cả người dùng đang sử dụng profile đó, vì vậy cần thực hiện thận trọng.

Thao tác trên giao diện web:

- Chọn Profile cần cấu hình.
- Thiết lập các giới hạn thời gian (Idle Time, Connect Time, Inactive Account Time) và giới hạn an ninh (Failed Login Attempts, Password Lock Time). Các giá trị có thể là UNLIMITED hoặc một số, hoặc để trống để giữ nguyên.
- Nhấn nút "Cập nhật Profile" để lưu thay đổi.

The screenshot shows a web-based configuration interface for updating a profile. The title bar is yellow with the text 'Cập nhật Profile'. Below it is a section labeled 'Chọn Profile' with a dropdown menu containing the option '-- Chọn Profile --'. To the right of this are four input fields, each with a placeholder text in Vietnamese: 'Idle Time (để trống nếu không đổi)' followed by a text input field containing 'UNLIMITED hoặc số'; 'Password Lock Time (để trống nếu không đổi)' followed by a text input field containing 'UNLIMITED hoặc số'; 'Connect Time (để trống nếu không đổi)' followed by a text input field containing 'UNLIMITED hoặc số'; and 'Inactive Account Time (để trống nếu không đổi)' followed by a text input field containing 'UNLIMITED hoặc số'. At the bottom of the form is a yellow button labeled 'Cập nhật Profile'.

Hình 4.39: Cập nhật profile

Thao tác bằng lệnh SQL:

```
create or replace PROCEDURE "UPDATE_PROFILE" (
    p_profile_name IN VARCHAR2,
    p_idle_time IN VARCHAR2 DEFAULT NULL,
    p_connect_time IN VARCHAR2 DEFAULT NULL,
    p_failed_login IN VARCHAR2 DEFAULT NULL,
    p_lock_time IN VARCHAR2 DEFAULT NULL,
    p_inactive_account_time IN VARCHAR2 DEFAULT NULL)
AS
    v_sql VARCHAR2(4000);
BEGIN
    v_sql := 'ALTER PROFILE ' || UPPER(p_profile_name) || ' LIMIT';
    IF p_idle_time IS NOT NULL THEN
        v_sql := v_sql || ' IDLE_TIME ' || p_idle_time;
    END IF;
    IF p_connect_time IS NOT NULL THEN
        v_sql := v_sql || ' CONNECT_TIME ' || p_connect_time;
    END IF;
    IF p_failed_login IS NOT NULL THEN
        v_sql := v_sql || ' FAILED_LOGIN_ATTEMPTS ' || p_failed_login;
    END IF;
    IF p_lock_time IS NOT NULL THEN
        v_sql := v_sql || ' PASSWORD_LOCK_TIME ' || p_lock_time;
    END IF;
    IF p_inactive_account_time IS NOT NULL THEN
        v_sql := v_sql || ' INACTIVE_ACCOUNT_TIME ' || p_inactive_account_time;
    END IF;
    EXECUTE IMMEDIATE v_sql;
END UPDATE_PROFILE;
```

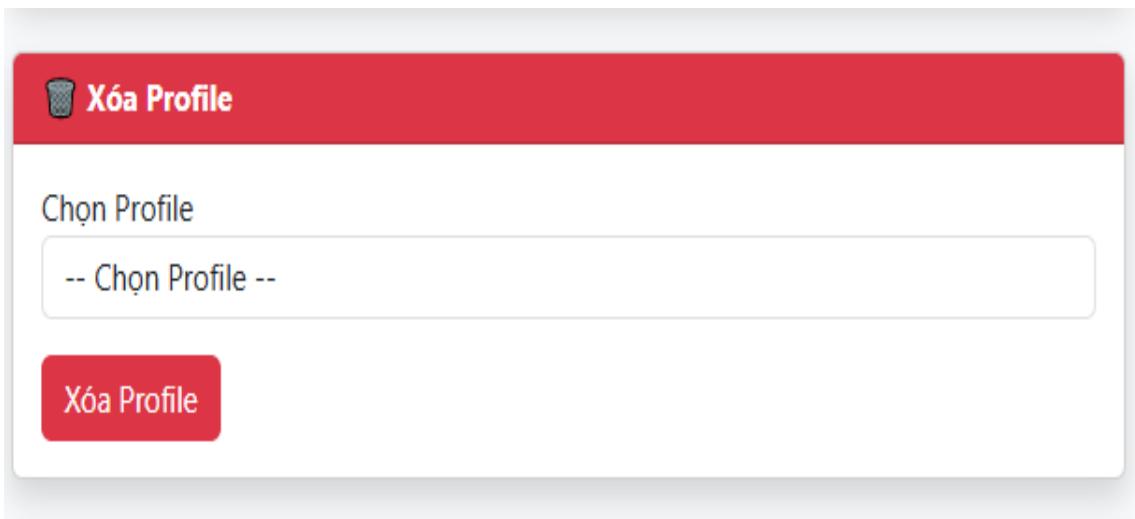
Hình 4.40: Thao tác bằng lệnh SQL của cập nhật profile.

4.5.2.6 Xóa profile

Xóa profile khi không còn sử dụng hoặc thay thế bằng profile khác. Khi xóa profile, tất cả người dùng đang sử dụng profile đó sẽ tự động được chuyển về profile DEFAULT.

Thao tác trên giao diện web:

- Chọn Profile: Người dùng vẫn phải chọn Profile cần xóa từ danh sách/hộp thả xuống.
- Xác nhận Xóa: Sau khi nhấn nút "Xóa Profile", hệ thống sẽ hiển thị một hộp thoại hoặc thông báo xác nhận (ví dụ: "Bạn có chắc chắn muốn xóa Profile [Tên Profile] không? Thao tác này không thể hoàn tác.").
- Hoàn tất: Nhấn "Đồng ý/Xóa" trong hộp thoại xác nhận để hoàn tất thao tác.



Hình 4.41: Xóa profile.

Thao tác bằng lệnh SQL:

```
create or replace PROCEDURE DELETE_PROFILE (
    p_profile_name IN VARCHAR2
)
AS
BEGIN
    EXECUTE IMMEDIATE 'DROP PROFILE ' || UPPER(p_profile_name) || ' CASCADE';
END DELETE_PROFILE;
```

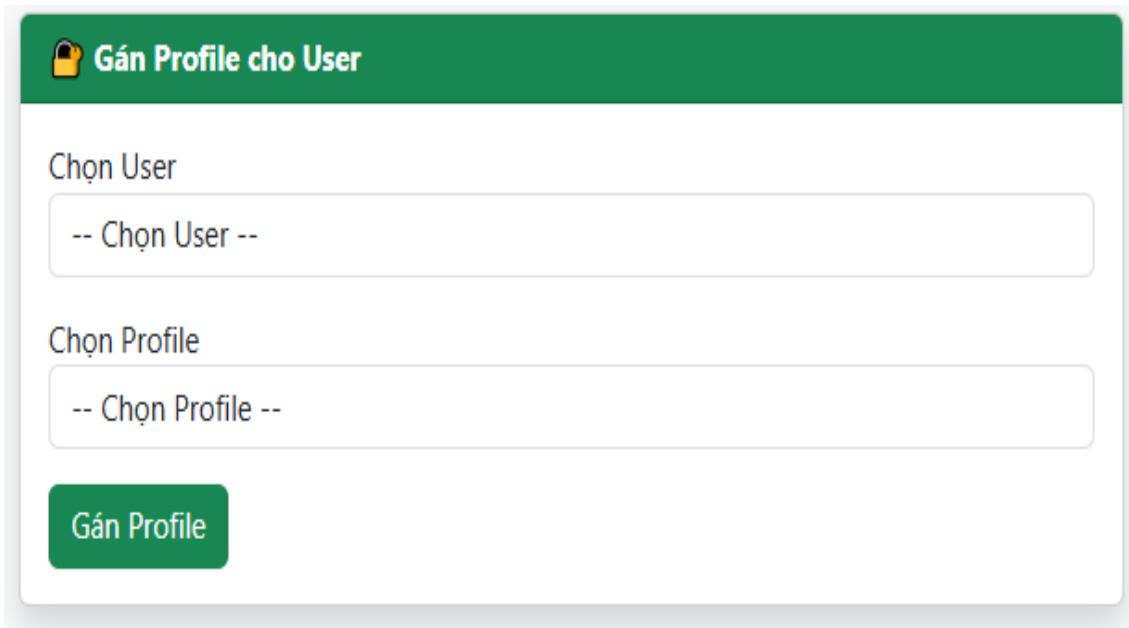
Hình 4.42: Thao tác bằng lệnh SQL của xóa profile.

4.5.2.7 Gán profile cho người dùng

Gán profile cho người dùng giúp kích hoạt các giới hạn bảo mật đã được định nghĩa trong profile. Một người dùng chỉ có thể sử dụng một profile tại một thời điểm.

Thao tác trên giao diện web:

- Chọn Profile: Quản trị viên phải chọn profile cần gán từ danh sách/hộp thả xuống.
- Chọn người dùng: Quản trị viên phải chọn người dùng cần được gán profile từ danh sách/hộp thả xuống.
- Xác nhận gán profile đã chọn cho người dùng đã chọn.



Hình 4.43: Gán profile cho người dùng.

Thao tác bằng lệnh SQL:

```
create or replace PROCEDURE ASSIGN_PROFILE_TO_USER (
    p_username IN VARCHAR2,
    p_profile_name IN VARCHAR2
)
AS
BEGIN
    EXECUTE IMMEDIATE 'ALTER USER "' || p_username || '" PROFILE "' || UPPER(p_profile_name) || '"';
END ASSIGN_PROFILE_TO_USER;
```

Hình 4.44: Thao tác bằng lệnh SQL của gán profile.

4.6. Xác thực đa nền tảng bằng mã QR

4.6.1. Tổng quan

Đăng nhập bằng mã QR cho phép người dùng xác thực trên web bằng cách quét mã QR từ ứng dụng mobile (hoặc ngược lại) — tạo trải nghiệm nhanh, không cần nhập mật khẩu trên thiết bị hiện tại.

Thường dùng cho:

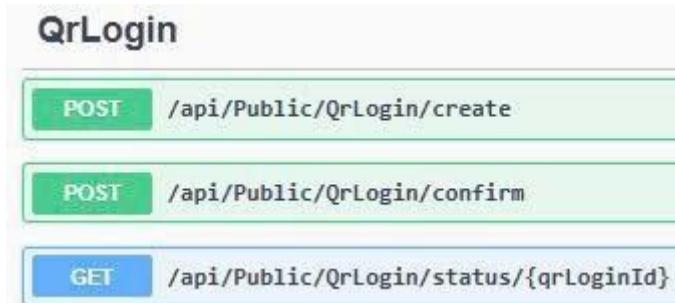
- Xác thực web bằng mobile (web hiển thị mã QR và code → mobile quét → xác thực).
- Xác thực mobile bằng web (web hiển thị mã QR và code → mobile quét → xác thực).

4.6.2. Luồng hoạt động

4.6.2.1 Xác thực Web từ Mobile (Mobile Authenticates Web)

Phía WebApp (Browser)

- Bước 1: Gọi API POST /api/Public/QrLogin/create (cho phép anonymous)
- API sẽ trả về qrLoginId, code, expiresAtUtc.



Hình 4.45: API cho chức năng đăng nhập QR.

- Bước 2: Browser hiển thị mã QR cho người dùng và đồng thời bắt đầu polling trạng thái mã QR mỗi 2-3 giây.
- Bước 3: Kiểm tra trạng thái mã QR
 - Nếu trạng thái là Pending thì tiếp tục polling và hiển thị đồng hồ đếm ngược.
 - Nếu trạng thái là Confirmed thì gọi API POST /Public/Customer/CompleteQrLogin và gửi Gửi: { username, roles, token, sessionId } (lấy từ API polling).
- Bước 4: Người dùng sẽ được đăng nhập vào tài khoản của mình thông qua phương thức Proxy Authentication.

Phía Mobile

- Bước 1: Đã đăng nhập sẵn vào tài khoản và mở chức năng quét hoặc nhập mã code.
- Bước 2: Gửi xác nhận thông qua API POST /api/Public/QrLogin/confirm với JWT Token của mobile.
- Bước 3: Xác thực mã QR và cập nhật trạng thái của mã QR.

```

[HttpPost("confirm")]
[Authorize(AuthenticationSchemes = "Bearer")]
public ActionResult<ApiResponse<string>> Confirm([FromBody] QrLoginConfirmRequest request)
{
    if (request == null || string.IsNullOrWhiteSpace(request.Code))
    {
        return BadRequest(ApiResponse<string>.Fail("Code is required."));
    }
    var session = _qrLoginStore.GetByCode(request.Code);
    if (session == null || session.Status != Services.QrLoginStatus.Pending)
    {
        return BadRequest(ApiResponse<string>.Fail("Code không hợp lệ hoặc đã hết hạn."));
    }
    var username = User?.Identity?.Name;
    if (string.IsNullOrWhiteSpace(username))
    {
        return Unauthorized(ApiResponse<string>.Fail("Không xác định được username từ JWT."));
    }
    try
    {
        var isCustomer = HasCustomerRole(User.Claims);
        var proxyLogin = isCustomer
            ? _proxyLoginService.LoginCustomer(username, "WEB")
            : _proxyLoginService.LoginEmployee(username, "WEB");

        session.Status = Services.QrLoginStatus.Confirmed;
        session.Username = proxyLogin.Username;
        session.Roles = proxyLogin.Roles;
        session.WebToken = proxyLogin.Token;
        session.WebSessionId = proxyLogin.SessionId;

        return Ok(ApiResponse<string>.Ok("CONFIRMED"));
    }
    catch (Exception ex)
    {
        return StatusCode(500, ApiResponse<string>.Fail($"QR login failed: {ex.Message}"));
    }
}

```

Hình 4.46: Mã nguồn xác thực đa nền tảng đăng nhập web từ mobile.

4.6.2.2 Xác thực Mobile từ Web (Web Authenticates Mobile)

Phía WebApp (Browser)

- Bước 1: Đã đăng nhập và gọi API.
POST/api/Public/WebToMobileQr/create để sinh mã QR
- Bước 2: Hiển thị mã QR/code vừa nhận được.

Phía Mobile

- Bước 1: Chọn chức năng đăng nhập bằng mã QR từ web.
- Bước 2: Nhập mã QR/code và gửi API xác nhận POST.
/api/Public/WebToMobileQr/confirm
- Bước 3: Kiểm tra hợp lệ và trạng thái mã QR.
- Bước 4: Người dùng sẽ được đăng nhập vào tài khoản của mình thông qua phương thức Proxy Authentication.

```

[HttpPost("confirm")]
[AllowAnonymous]
0 references
public ActionResult<ApiResponse<WebToMobileQrConfirmResponse>> Confirm([FromBody] WebToMobileQrConfirmRequest request)
{
    if (request == null || string.IsNullOrWhiteSpace(request.Code))
    {
        return BadRequest(ApiResponse<WebToMobileQrConfirmResponse>.Fail("Code is required."));
    }

    var session = _store.GetByCode(request.Code);
    if (session == null || session.Status != WebToMobileQrStatus.Pending)
    {
        return BadRequest(ApiResponse<WebToMobileQrConfirmResponse>.Fail("Code không hợp lệ hoặc đã hết hạn."));
    }

    try
    {
        var platform = string.IsNullOrWhiteSpace(request.Platform)
            ? "MOBILE"
            : request.Platform!;

        var isCustomer = HasCustomerRole(session.SourceRoles);
        var proxyLogin = isCustomer
            ? _proxyLoginService.LoginCustomer(session.SourceUsername, platform)
            : _proxyLoginService.LoginEmployee(session.SourceUsername, platform);

        _store.MarkConfirmed(session, proxyLogin.Token, proxyLogin.SessionId, proxyLogin.Roles);

        var response = new WebToMobileQrConfirmResponse
        {
            Username = proxyLogin.Username,
            Roles = proxyLogin.Roles,
            Token = proxyLogin.Token,
            SessionId = proxyLogin.SessionId
        };
    }

    return Ok(ApiResponse<WebToMobileQrConfirmResponse>.Ok(response));
}

```

Hình 4.47: Mã nguồn xác thực đa nền tảng web từ mobile.

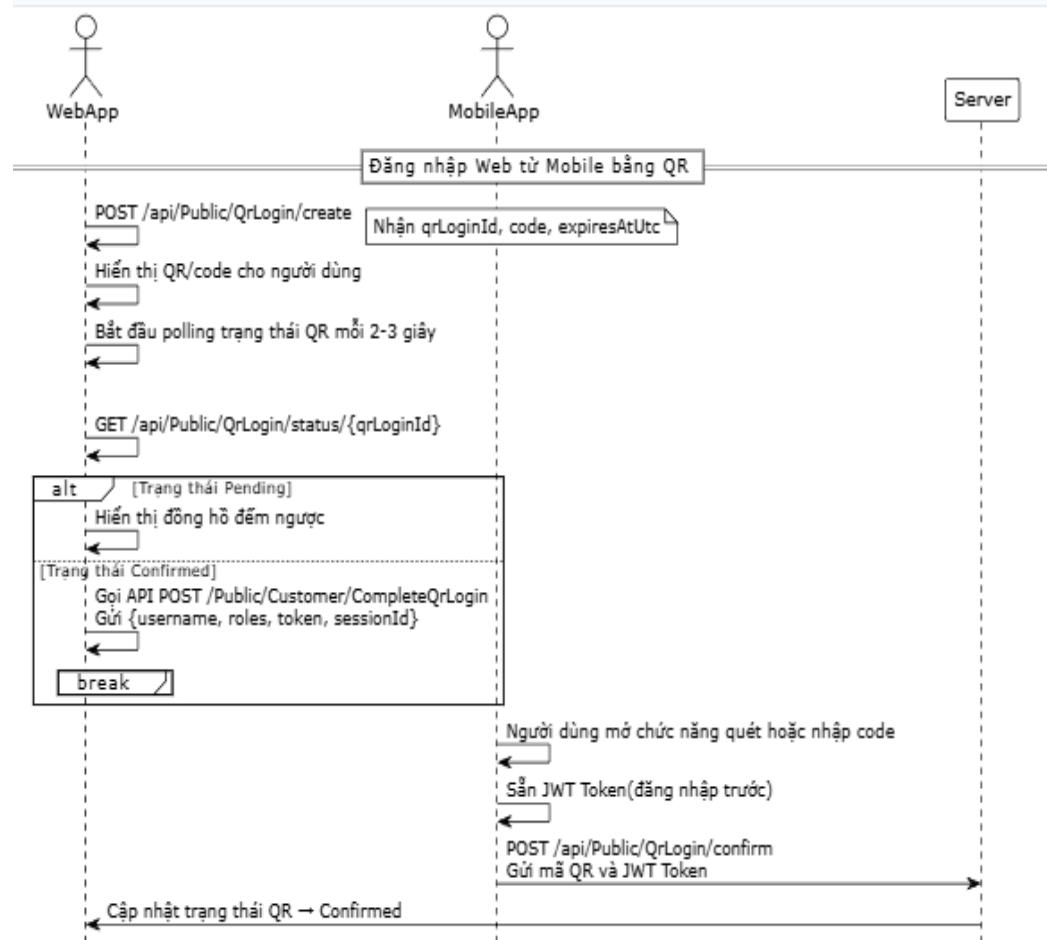
4.6.3. Tóm tắt vai trò

- Xác thực Web từ Mobile

Bảng 4.4: Bảng tóm tắt vai trò xác thực Web từ Mobile bằng mã QR

Bên	API endpoint	Mục đích
Web App	POST /api/Public/QrLogin/create	Tạo mã QR/Code
Web App	GET/api/Public/QrLogin/status/{qrLoginId}	Polling trạng thái
Mobile App	POST /api/Public/QrLogin/confirm	Xác nhận mã (sử dụng JWT mobile)

Web App	POST /Public/Customer/CompleteQrLogin	Hoàn tất đăng nhập web
---------	---------------------------------------	------------------------

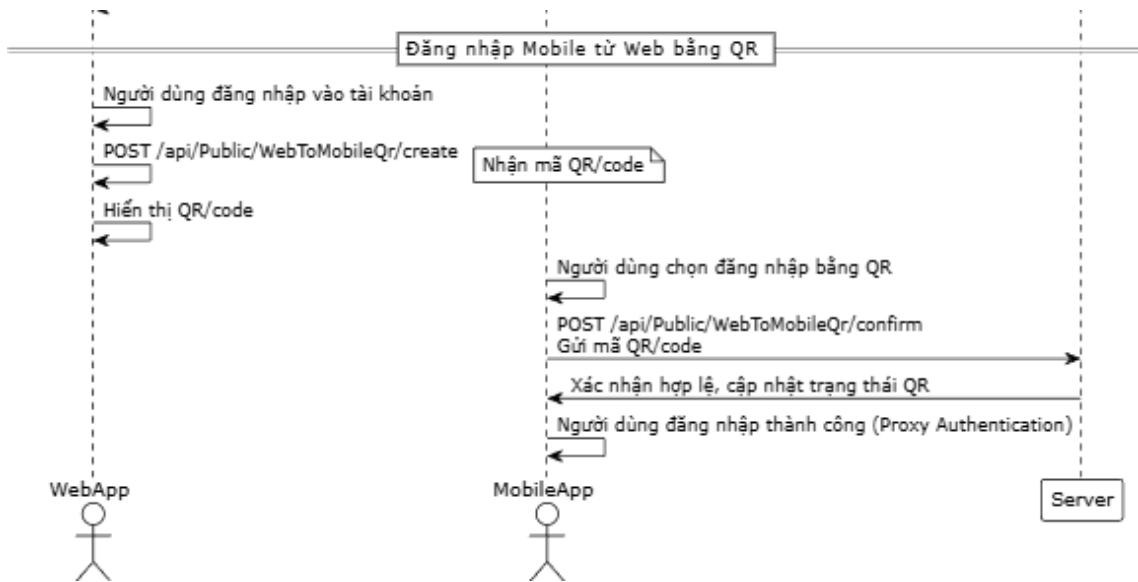


Hình 4.48: Sơ đồ xác thực Web từ Mobile bằng mã QR

– Xác thực Mobile từ Web

Bảng 4.5: Bảng tóm tắt vai trò xác thực điện thoại từ web bằng QR.

Bên	API endpoint	Mục đích
Web App	POST /api/Public/WebToMobileQr/create	Tạo mã QR/Code
MobileApp	POST /api/Public/WebToMobileQr/confirm	Xác nhận và hoàn tất đăng nhập



Hình 4.49: Sơ đồ xác thực Mobile từ Web bằng chức năng QR Login.

4.7. Đảm bảo chỉ 1 phiên làm việc tồn tại trên cùng 1 nền tảng

Hệ thống cho phép một người dùng đăng nhập trên nhiều nền tảng khác nhau (WEB, MOBILE), nhưng mỗi nền tảng chỉ được tồn tại duy nhất 1 session hợp lệ. Cơ chế này giúp ngăn chặn tình trạng chia sẻ tài khoản, giảm rủi ro bảo mật và đảm bảo trạng thái nhất quán của phiên làm việc trên Oracle.

Luồng hoạt động chi tiết:

Khi người dùng đăng nhập, hệ thống thực thi:

RemoveAllConnections(username, platform)

Hàm này tìm tất cả session hiện có ứng với username + platform trong dictionary quản lý kết nối.

Nếu phát hiện phiên cũ, hệ thống sẽ:

- Xóa entry khỏi dictionary. Đóng OracleConnection tương ứng.
- Gửi thông báo SignalR ForceLogout tới client đang sử dụng phiên cũ để buộc quay lại màn hình đăng nhập.

Điều này đảm bảo mọi phiên cũ trên cùng nền tảng bị hủy trước khi tạo phiên mới.

Quá trình xử lý khi login

Người dùng nhập username/password, gọi thủ tục LOGIN_EMPLOYEE / LOGIN_CUSTOMER.

Gọi RemoveAllConnections(username, platform) để dọn dẹp phiên cũ.

Tạo OracleConnection mới. Ghi nhận MODULE + SESSION_IDENTIFIER vào Oracle.

Lưu lại thông tin vào dictionary dưới khóa: (username, platform, sessionId).

```
        public async Task RemoveAllConnections(string username, string platform)
    {
        var keys = _connections.Keys
            .Where(k => k.username == username && k.platform == platform)
            .ToList();

        foreach (var key in keys)
            await RemoveConnection(key.username, key.platform, key.sessionId);
    }
```

Hình 4.50: Mã nguồn dọn dẹp phiên làm việc cũ.

```
using var defaultConn = CreateDefaultConnection();
var killResult = OracleHelper.ExecuteNonQueryWithOutputs(
    defaultConn,
    "APP.KILL_SESSION_BY_CLIENT_ID",
    new[]
    {
        ("p_client_identifier", OracleDbType.Varchar2, (object)sessionId)
    },
    new[]
    {
        ("p_result", OracleDbType.Varchar2)
    });

```

Hình 4.51: Mã nguồn xóa phiên làm việc.

```
conn.Open();
using (var cmd = conn.CreateCommand())
{
    cmd.CommandText = "BEGIN DBMS_APPLICATION_INFO.SET_MODULE(:module_name, :client_info); END;";
    cmd.Parameters.Add(new OracleParameter("module_name", $"WebAPI-{platform}"));
    cmd.Parameters.Add(new OracleParameter("client_info", sessionId));
    cmd.ExecuteNonQuery();
}
using (var cmd = conn.CreateCommand())
{
    cmd.CommandText = "BEGIN DBMS_SESSION.SET_IDENTIFIER(:sid); END;";
    cmd.Parameters.Add(new OracleParameter("sid", sessionId));
    cmd.ExecuteNonQuery();
}
```

Hình 4.52: Mã nguồn định danh kết nối mới.

Ý nghĩa

- Tránh việc 1 tài khoản đăng nhập đồng thời trên nhiều thiết bị cùng loại (VD: 2 trình duyệt web hoặc 2 điện thoại).

- Ngăn các thay đổi trạng thái (SET_IDENTIFIER, session SID) bị ghi đè bởi phiên song song. Đảm bảo mỗi request truy cập API luôn ánh xạ đúng với 1 kết nối Oracle duy nhất. Giảm rủi ro tấn công “session hijacking”.

4.8. Đăng ký

```
-- 5. Tạo Oracle DB user
BEGIN
    EXECUTE IMMEDIATE 'CREATE USER ' || p_username ||
        ' IDENTIFIED BY "' || v_pwd || '" DEFAULT TABLESPACE USERS QUOTA UNLIMITED ON USERS';
    EXECUTE IMMEDIATE 'GRANT CONNECT, RESOURCE TO ' || p_username;
    EXECUTE IMMEDIATE 'GRANT EXECUTE ON LOGIN_EMPLOYEE TO ' || p_username;
    EXECUTE IMMEDIATE 'ALTER USER ' || p_username || ' GRANT CONNECT THROUGH APP';

EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001, 'Không thể tạo DB User cho nhân viên: ' || SQLERRM);
END;
```

Hình 4.53. Mã nguồn đăng ký nhân viên.

Luồng hoạt động chi tiết

Quy trình đăng ký tài khoản không chỉ đơn thuần là lưu trữ thông tin, mà còn kích hoạt cơ chế tự động hóa quản lý người dùng ngay tại tầng Cơ sở dữ liệu (Database Layer). Cụ thể:

- Bước 1: Tiếp nhận và Xác thực: Người dùng nhập thông tin định danh thông qua giao diện ứng dụng. Hệ thống thực hiện các bước kiểm tra hợp lệ (validation) ban đầu.
- Bước 2: Khởi tạo Identity tại DB:
- Sau khi xác thực, ứng dụng gọi thủ tục lưu trữ (Stored Procedure). Tại đây, khôi lệnh PL/SQL sử dụng kỹ thuật Dynamic SQL(EXECUTE IMMEDIATE) để trực tiếp tạo một Schema User mới tương ứng với tên đăng nhập (p_username).
- Bước 3: Phân quyền tự động
- Ngay sau khi user được tạo, hệ thống thực hiện cấp phát quyền hạn theo cơ chế:
 - Cấp quyền cơ bản: CONNECT, RESOURCE (cho phép kết nối và tạo đối tượng cơ bản).
 - Cấp quyền thực thi nghiệp vụ: GRANT EXECUTE ON LOGIN_EMPLOYEE (chỉ cho phép chạy thủ tục đăng nhập, giới hạn quyền truy cập bảng trực tiếp).

- Thiết lập Proxy User: GRANT CONNECT THROUGH APP (Bắt buộc người dùng phải kết nối thông qua ứng dụng định danh APP, ngăn chặn việc đăng nhập trực tiếp trái phép vào DB).

Ý nghĩa

Việc tích hợp logic tạo user vào quy trình đăng ký giúp hoàn thiện chu trình bảo mật đa lớp của hệ thống, cụ thể như sau:

- Đảm bảo tính Nhất quán và Toàn vẹn (Data Consistency & Integrity): Việc tạo tài khoản được xử lý đồng bộ. Dữ liệu định danh tại tầng ứng dụng và tài khoản thực tế dưới tầng cơ sở dữ liệu luôn khớp nhau 1-1. Điều này loại bỏ tình trạng "tài khoản ảo" và giúp việc quản trị (audit) trở nên minh bạch.
- Phòng chống tấn công SQL Injection: Mặc dù sử dụng Dynamic SQL, hệ thống giảm thiểu rủi ro bằng cách kiểm soát chặt chẽ dữ liệu đầu vào. Ở tầng ứng dụng (Application Layer), các tham số được truyền qua cơ chế Parameter Binding (sử dụng oci_bind_by_name hoặc tương đương). Điều này đảm bảo dữ liệu đầu vào được xử lý như một giá trị thuần túy (literal value) thay vì một phần của câu lệnh thực thi, vô hiệu hóa nỗ lực chèn mã độc của kẻ tấn công qua form đăng ký.
- Tuân thủ Nguyên tắc Đặc quyền Tối thiểu (Principle of Least Privilege): Đây là điểm mạnh nhất của thiết kế này. Người dùng mới (role_khachhang) được tạo ra với quyền hạn bị giới hạn nghiêm ngặt:
 - + Cách ly quyền hạn: User khách hàng hoàn toàn tách biệt với các quyền quản trị (DBA) hoặc quyền của nhân viên.
 - + Kiểm soát hành vi: User không được cấp quyền SELECT/INSERT/UPDATE trực tiếp lên các bảng dữ liệu nhạy cảm. Họ chỉ có thể tương tác với dữ liệu thông qua các thủ tục đã được định nghĩa sẵn (như LOGIN_EMPLOYEE). Điều này đảm bảo dù hacker có chiếm được tài khoản này, phạm vi phá hoại cũng bị cô lập hoàn toàn.

4.9. Đổi mật khẩu

```
-- 1. Lấy hash mật khẩu hiện tại từ bảng CUSTOMER
SELECT PASSWORD_HASH
INTO v_old_hash
FROM EMPLOYEE
WHERE USERNAME = p_username;

-- 2. Kiểm tra mật khẩu cũ
IF v_old_hash != HASH_PASSWORD(p_old_password) THEN
    RAISE_APPLICATION_ERROR(-20001, 'Mật khẩu cũ không đúng cho khách hàng: ' || p_username);
END IF;
```

Hình 4.54: Mã nguồn lấy mật khẩu hiện tại và kiểm tra mật khẩu cũ.

```
-- 3. Hash mật khẩu mới
v_new_hash := HASH_PASSWORD(p_new_password);
v_pwd      := HASH_PASSWORD_20CHARS(p_new_password);

-- 4. Cập nhật bảng Employee
BEGIN
    UPDATE EMPLOYEE
    SET PASSWORD_HASH = v_new_hash
    WHERE USERNAME = p_username;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;

-- 5. Cập nhật Oracle DB user password
BEGIN
    EXECUTE IMMEDIATE 'ALTER USER "' || UPPER(p_username) || '" IDENTIFIED BY "' || v_pwd || '"';

```

Hình 4.55: Mã nguồn băm mật khẩu và cập nhập bảng.

Trong hệ thống này, do người dùng ứng dụng cũng chính là người dùng (Schema User) trong Oracle Database, việc đổi mật khẩu không đơn thuần là cập nhật một dòng trong bảng dữ liệu mà là thay đổi thông tin xác thực của Database User. Chức năng này cho phép người dùng tự thao tác thay đổi mật khẩu của mình mà không cần sự can thiệp của quản trị viên, đảm bảo tuân thủ các chính sách bảo mật về vòng đời mật khẩu (Password Lifetime) đã thiết lập trong Profile.

Mục tiêu:

- Xác thực người dùng: Đảm bảo người yêu cầu đổi mật khẩu thực sự là chủ sở hữu tài khoản (bằng cách yêu cầu nhập mật khẩu cũ).
- Tuân thủ chính sách Oracle: Mật khẩu mới phải thỏa mãn các điều kiện về độ phức tạp và lịch sử mật khẩu (Password Reuse) do Oracle Profile quy định.
- Bảo mật phiên làm việc: Ngăn chặn tấn công chiếm quyền phiên (Session Hijacking) để đổi mật khẩu trái phép

Luồng hoạt động chi tiết

- Bước 1: Xác thực lớp JWT Hệ thống nhận Authorization Header chứa JWT. Giải mã token, username. Nếu token không hợp lệ hoặc hết hạn, từ chối yêu cầu ngay lập tức (HTTP 401).
- Bước 2: Kiểm tra dữ liệu đầu vào bên phía frontend. Yêu cầu bắt buộc: Mật khẩu hiện tại (currentPassword) và mật khẩu mới (newPassword). Validation: Mật khẩu mới phải có ít nhất 6 ký tự và khác mật khẩu hiện tại.
- Bước 3: Xác thực mật khẩu cũ tại Oracle. Thay vì so sánh mã băm (hash) như các ứng dụng thông thường, hệ thống thực hiện xác thực "sống" (Real-time Authentication) với Cơ sở dữ liệu:
- Hệ thống backend thử thiết lập một kết nối mới tới Oracle Database sử dụng Username (lấy từ JWT) và currentPassword (người dùng nhập).
- Cơ chế: Nếu kết nối thất bại (Oracle trả về lỗi ORA-01017: invalid username/password), hệ thống xác định ngay lập tức mật khẩu cũ không chính xác và từ chối yêu cầu. Điều này đảm bảo rằng chỉ người nắm giữ mật khẩu hiện tại mới có quyền bắt đầu quy trình thay đổi.
- Bước 4: Thực thi đổi mật khẩu cho người dùng:
 - + Sau khi xác thực thành công ở Bước 3, hệ thống thực thi câu lệnh SQL thay đổi mật khẩu. Để đảm bảo tính nguyên tử (Atomicity), câu lệnh ALTER USER được sử dụng kèm mệnh đề REPLACE:

ALTER USER [username] IDENTIFIED BY "[newPassword]"
REPLACE "[currentPassword]";
 - + Tại bước này, Oracle Database đóng vai trò là "người gác cổng" cuối cùng:
 1. Kiểm tra độ phức tạp (Complexity Verification): Mật khẩu mới được đối chiếu với VERIFY_FUNCTION trong Profile của user (ví dụ: độ dài, chữ hoa/thường, ký tự đặc biệt).
 2. Kiểm tra lịch sử (History Verification): Hệ thống kiểm tra PASSWORD_REUSE_TIME và PASSWORD_REUSE_MAX để đảm bảo người dùng không sử dụng lại các mật khẩu cũ.
 3. Hoàn tất: Nếu tất cả điều kiện thỏa mãn, mật khẩu được cập nhật và commit. Nếu vi phạm, Oracle trả về lỗi cụ thể (ví dụ: ORA-28003: password verification for the specified password failed) để ứng dụng thông báo lại cho người dùng.

Cơ chế đổi mật khẩu này mang lại các lợi ích:

- Chống tấn công CSRF/XSS: Kể cả khi kẻ tấn công đánh cắp được JWT (Session Token), chúng cũng không thể đổi mật khẩu của nạn nhân vì không biết currentPassword (Mật khẩu cũ là bắt buộc để thiết lập kết nối Database ở Bước 3).
- Không lưu trữ Hash: Hệ thống không cần quản lý hay lưu trữ bất kỳ mã băm mật khẩu nào tại tầng ứng dụng, loại bỏ hoàn toàn rủi ro lộ lọt dữ liệu xác thực từ mã nguồn C#.

4.10. Ký số và xác thực hóa đơn

4.10.1. Mục tiêu

- Tạo tệp PDF hóa đơn điện tử từ dữ liệu đơn hàng, phiếu nhập, phiếu xuất trong cơ sở dữ liệu Oracle.
- Xác thực người ký: khóa riêng được lưu ở máy tính cá nhân.
- Đảm bảo ba yếu tố bảo mật: tính toàn vẹn: tính xác thực và không chối bỏ.

4.10.2. Mô hình và kiến trúc hệ thống

4.5.2.1 Cấu trúc file PKCS#12 (.p12/.pfx)

File PKCS#12 (.p12 hoặc .pfx) là một container nhị phân chứa:

Chứng thư X.509 (Public Certificate)

- Tương đương với file .crt hoặc .cer.
- Chứa thông tin công khai: Subject, Issuer, Serial Number, Validity Period.
- Có thể verify bằng public key.

Khóa riêng (Private Key)

- Tương đương với file .key.
- RSA hoặc ECC private key.
- Được mã hóa và bảo vệ bằng password trong PFX.

Chuỗi chứng thư (Certificate Chain)

- Intermediate CA certificates (nếu có).
- Root CA certificate (nếu có).

- Giúp xây dựng trust chain để verify certificate.

Không lưu file rời:

- Hệ thống không lưu riêng file .crt và .key trên đĩa.
- Tất cả thông tin được đóng gói trong file .pfx/.p12 duy nhất.
- PFX được upload từ client, chỉ tồn tại trong bộ nhớ khi xử lý.

4.10.3. Luồng hoạt động

- Bước 1: Client upload file .pfx và password đi kèm.
- Bước 2: Gửi đến api với nội dung đã mã hóa RSA/AES đến server.

POST /api/admin/import/create/secure
POST /api/admin/export/create/secure

- Bước 3: Server nhận request và giải mã payload.
- Bước 4: Tạo các record hóa đơn, phiếu nhập, phiếu xuất trong tương ứng transaction.
- Bước 5: Tạo chữ ký số và lưu cho thông tin ở phía cơ sở dữ liệu, đảm bảo dữ liệu bên cơ sở dữ liệu toàn vẹn và không chối bỏ.
- Bước 6: Tạo pdf chưa có chữ ký số dựa theo thông tin đúng của cơ sở dữ liệu.
- Bước 7: Sử dụng PFX Certificate từ người dùng gửi và ký pdf bằng GroupDocs.Signature.
- Bước 8: Lưu pdf vào cơ sở dữ liệu và trả pdf về cho người dùng để in, gửi,..

```

1 reference
private byte[] GenerateAndSignPdf(
    Func<byte[]> pdfFactory,
    byte[] certificatePfxBytes,
    string certificatePassword,
    string updateProcedureName,
    Action<OracleCommand> configureUpdateProcedure,
    string invoiceType,
    int itemCount,
    int serviceCount = 0)
{
    _pdfSigner.ValidateCertificate(certificatePfxBytes, certificatePassword);

    var pdfBytes = pdfFactory();
    var (left, top) = CalculateSignaturePosition(invoiceType, itemCount, serviceCount);

    var signedPdfBytes = _pdfSigner.SignPdfWithDigitalCertificate(
        pdfBytes,
        certificatePfxBytes,
        certificatePassword,
        options =>
    {
        options.Left = left;
        options.Top = top;
        options.Margin = new Padding(0);
    });

    _pdfSigner.UpdateFinalPDF(
        procedureName: updateProcedureName,
        PDF: signedPdfBytes,
        configureParameters: configureUpdateProcedure);

    return signedPdfBytes;
}

```

Hình 4.56: Mã nguồn tạo và ký PDF.

Created with evaluation version of GroupDocs.Signature © Aspose Pty Ltd 2001-2025. All Rights Reserved.

Công ty TNHH HealthCare
MST: 0123456789
123 Đường ABC, Phường XYZ, Quận 1, TP.HCM
ĐT: 0123456789
Email: info@mobileservice.com

Ký bởi 'E=vinhdtq123123123@gmail.com'
Ngày: 2025.11.30 12:23:09
Lý do: Approved
Địa điểm: Việt Nam

HÓA ĐƠN BÁN HÀNG

Mã hóa đơn: #20
Ngày: 30/11/2025 19:22
Khách hàng: 0332880207
Nhân viên: Admin
Linh kiện:

Tên linh kiện	Hàng	Serial	Giá
TEST	TEST	TEST-0-44	1.000 ₫
		Tổng linh kiện:	1.000 ₫

Dịch vụ:

Tên dịch vụ	SL	Đơn giá	Thành tiền
Dịch vụ Bảo hành	1	0 ₫	0 ₫
		Tổng dịch vụ:	0 ₫

Hình 4.57: Hóa đơn bán hàng ở định dạng PDF.



Công ty TNHH HealthCare

MST: 0123456789
123 Đường ABC, Phường XYZ, Quận 1, TP.HCM
ĐT: 0123456789
Email: info@mobileservice.com

Ký bởi 'E=vinhdltq123123123@gmail.c

Ngày: 2025.11.28 18:05:19

Lý do: Approved

Địa điểm: Việt Nam

HÓA ĐƠN NHẬP KHO

Mã phiếu: #91

Ngày: 29/11/2025 01:05

Nhân viên: Admin

Ghi chú: s

Tên linh kiện	Hãng	Serial	Giá
SSSS	SSSS	SSSS	10.000 đ
		Tổng cộng	10.000 đ

Chữ ký số:
(Đã ký số)

Công ty TNHH HealthCare

Hình 4.58: Hóa đơn nhập kho ở định dạng PDF.



Công ty TNHH HealthCare

MST: 0123456789
123 Đường ABC, Phường XYZ, Quận 1, TP.HCM
ĐT: 0123456789
Email: info@mobileservice.com

Ký bởi 'E=vinhdltq123123123@gmail.c

Ngày: 2025.11.30 12:22:33

Lý do: Approved

Địa điểm: Việt Nam

HÓA ĐƠN XUẤT KHO

Mã phiếu: #42

Ngày: 30/11/2025 19:22

Nhân viên: Admin

Ghi chú: Xuất kho tự động cho Order ID 323

Tên linh kiện	Hãng	Serial	Giá
TEST	TEST	TEST-0-44	1.000 đ
		Tổng cộng	1.000 đ

Chữ ký số:

Công ty TNHH HealthCare

Hình 4.59: Hóa đơn xuất kho ở định dạng PDF.

4.10.4. Xác thực chữ ký số

4.10.4.1 Xác thực chữ ký bên phía cơ sở dữ liệu

– Mục tiêu:

- + Bảo vệ dữ liệu quan trọng: Ví dụ thông tin thanh toán, hợp đồng điện tử, chứng từ.
- + Đảm bảo tính toàn vẹn: Dữ liệu có bị sửa đổi không.
- + Đáp ứng yêu cầu kiểm toán nội bộ.

– Cách hoạt động:

- + Bước 1: Admin hoặc người dùng có quyền tương ứng có thể xác thực chữ ký số được lưu trong cột signature của các bảng hóa đơn, phiếu nhập, phiếu xuất. Mỗi chữ ký số có định dạng empidsigner-signature và việc xác thực dựa trên public key của người ký.
- + Bước 2: Hệ thống kiểm tra tính hợp lệ của dữ liệu bằng cách so sánh chữ ký số với dữ liệu gốc. Nếu dữ liệu bị thay đổi, chữ ký số sẽ không khớp, hệ thống sẽ cảnh báo.

```
[HttpGet("{stockinId}/verify")]
[Authorize]
0 references
public async Task<IActionResult> VerifyStockInSignature(int stockinId)
{
    return await _helper.ExecuteWithConnection(HttpContext, conn =>
    {

        // 3. Get signature
        string? signature = OracleHelper.ExecuteClobOutput(
            conn,
            "APP.GET_STOCKIN_SIGNATURE",
            "p_signature",
            ("p_stockin_id", OracleDbType.Int32, stockinId));

        if (string.IsNullOrEmpty(signature))
            return NotFound(new { message = $"Signature của StockIn ID {stockinId} không tồn tại" });

        string empIdStr = signature.Split('-')[0]; // Lấy phần trước dấu "-"
        int empId = int.Parse(empIdStr); // Chuyển sang int
        signature = signature.Split('-')[1];

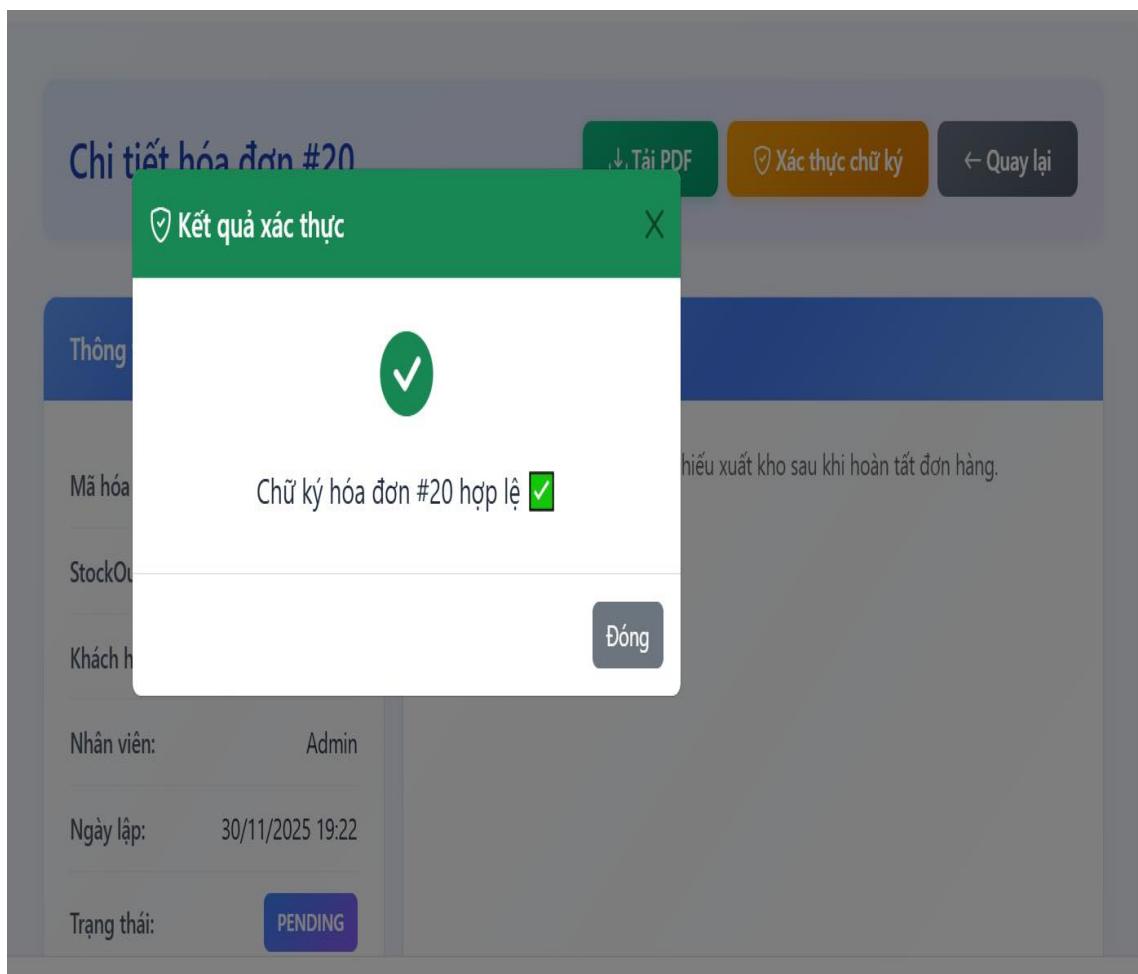
        string? publicKey = OracleHelper.ExecuteClobOutput(
            conn,
            "APP.GET_EMPLOYEE_PUBLIC_KEY",
            "p_pub_key",
            ("p_emp_id", OracleDbType.Int32, empId));

        if (string.IsNullOrEmpty(publicKey))
            return NotFound(new { message = $"Public Key của Employee ID {empId} không tồn tại" });
        // 4. Verify signature
        var verifyOutput = OracleHelper.ExecuteNonQueryWithOutputs(
            conn,
            "APP.VERIFY_STOCKIN_SIGNATURE",
            new[]
            {
                ("p_stockin_id", OracleDbType.Int32, (object)stockinId),
                ("p_public_key", OracleDbType.Varchar2, publicKey),
                ("p_signature", OracleDbType.Clob, signature)
            },
            new[] { ("p_is_valid", OracleDbType.Int32) });

        int isValid = ((Oracle.ManagedDataAccess.Types.OracleDecimal)verifyOutput["p_is_valid"]).ToInt32();

        return Ok(new
        {
            StockinId = stockinId,
            IsValid = isValid == 1
        });
    }, "Lỗi xác thực chữ ký phiếu nhập");
}
```

Hình 4.60: Mã nguồn xác thực chữ ký số



Hình 4.61: Kết quả khi xác thực chữ ký hợp lệ.

4.10.4.2 Xác thực chữ ký bên phía hóa đơn

- Mục tiêu:
 - + Xác nhận rằng hóa đơn được tạo ra và ký bởi người/đơn vị của cửa hàng.
 - + Đảm bảo tính toàn vẹn của hóa đơn: dữ liệu trên hóa đơn không bị sửa đổi sau khi ký.
 - + Hỗ trợ kiểm toán nội bộ và tuân thủ các quy định pháp lý về lưu trữ và xác thực hóa đơn điện tử.
 - + Ngăn ngừa gian lận, giả mạo hóa đơn trong hệ thống.
- Cách thức hoạt động:
 - + Bước 1: Client sẽ upload hóa đơn lên Web.
 - + Bước 2: Server sẽ so sánh 2 file pdf ở phía cơ sở dữ liệu và client. Nếu Bytes khác thì chứng tỏ pdf của client đã bị chỉnh sửa.

Xác thực PDF hóa đơn đã ký

File PDF đã ký

Choose File No file chosen

Hệ thống sẽ tự đọc mã hóa đơn từ file (ví dụ: **Invoice_4.pdf**).

Xác thực

PDF trùng khớp với bản lưu trong hệ thống.

Lưu ý: hệ thống chỉ so sánh hash SHA-256 giữa file bạn tải lên và bản lưu trong cơ sở dữ liệu.

Hình 4.62: Kết quả với PDF trong bản lưu hệ thống.

Xác thực PDF hóa đơn đã ký

File PDF đã ký

Choose File Invoice_4_fake.pdf

Hệ thống sẽ tự đọc mã hóa đơn từ file (ví dụ: **Invoice_4.pdf**).

Xác thực

PDF không trùng với bản lưu trong hệ thống.

Lưu ý: hệ thống chỉ so sánh hash SHA-256 giữa file bạn tải lên và bản lưu trong cơ sở dữ liệu.

Hình 4.63: Kết quả không trùng với PDF trong bản lưu hệ thống.

4.11. Tích hợp mã QR trong tra cứu linh kiện

```
[HttpGet("{serial}/details")]
[Authorize]
public async Task<IActionResult> GetPartBySerial(string serial)
{
    return await _helper.ExecuteWithConnection(HttpContext, conn =>
    {
        var list = OracleHelper.ExecuteRefCursor(conn, "APP.GET_PART_BY_SERIAL", "p_cursor",
            reader => MapPart(reader),
            ("p_serial", OracleDbType.Varchar2, serial));

        if (list.Count == 0)
            return NotFound(new { message = $"Part with serial {serial} not found" });

        return Ok(list[0]);
    }, "Lỗi khi lấy chi tiết linh kiện");
}
```

Hình 4.64: Mã nguồn lấy chi tiết linh kiện.

```
<td data-label="QR">
@if (item.QRImage != null && item.QRImage.Length > 0)
{
    var base64 = Convert.ToBase64String(item.QRImage);
    var imgSrc = $"data:image/png;base64,{base64}";
    
}
</td>
```

Hình 4.65: Mã nguồn tạo ảnh QR.

Luồng hoạt động chi tiết (Detailed Workflow)

Quy trình được chia thành hai giai đoạn chính: Hiển thị mã QR trên giao diện quản lý và Truy xuất thông tin khi quét mã.

- Giai đoạn A: Hiển thị mã QR trên giao diện web Để nhân viên có thể in hoặc quét mã trực tiếp từ màn hình, hệ thống thực hiện chuyển đổi dữ liệu nhị phân sang định dạng hình ảnh ngay trên trình duyệt:
 - + Dữ liệu nguồn: Mã QR được lưu trữ dưới dạng mảng byte (byte[]) trong cơ sở dữ liệu (thuộc tính item.QRImage).
 - + Xử lý tại View (Frontend): Sử dụng Razor Syntax để kiểm tra dữ liệu. Nếu hình ảnh tồn tại, hệ thống sử dụng phương thức Convert.ToBase64String để chuyển đổi mảng byte thành chuỗi Base64.
 - + Rendering: Chuỗi Base64 được nhúng trực tiếp vào thẻ HTML với định dạng data: image/png;base64,... Cách tiếp cận này giúp giảm tải request đến server vì không cần tạo đường dẫn tệp ảnh vật lý riêng biệt, đồng thời tăng tốc độ hiển thị danh sách.

- Giai đoạn B: Truy xuất thông tin qua API (Backend) Khi thiết bị quét mã QR (hoặc ứng dụng di động) giải mã được chuỗi Serial, một yêu cầu HTTP GET sẽ được gửi đến hệ thống:
- Endpoint: GET /api/parts/{serial}/details.
- Cơ chế xác thực (Authentication): API được bảo vệ bởi attribute [Authorize]. Điều này bắt buộc người thực hiện thao tác quét phải là nhân viên đã đăng nhập và có phiên làm việc (Token) hợp lệ, ngăn chặn truy cập trái phép từ bên ngoài.
 - + Xử lý logic:
 - Hàm GetPartBySerial tiếp nhận tham số serial từ URL.
 - Hệ thống sử dụng _helper.ExecuteNonQuery để khởi tạo và quản lý vòng đời kết nối an toàn tới Oracle Database.
 - Thực thi thủ tục lưu trữ (Stored Procedure) có tên APP.GET_PART_BY_SERIAL thông qua OracleHelper.ExecuteReader. Kết quả trả về được ánh xạ (Map) sang đối tượng linh kiện.
 - + Phản hồi (Response):
 - Nếu tìm thấy dữ liệu (list.Count > 0): Trả về mã 200 OK kèm theo đối tượng JSON chứa thông tin chi tiết.
 - Nếu không tìm thấy (list.Count == 0): Trả về mã 404 Not Found với thông báo lỗi cụ thể, giúp người dùng nhận biết mã QR không tồn tại trong hệ thống.

Giải pháp cài đặt thể hiện sự tối ưu trong thiết kế hệ thống:

- Hiệu năng hiển thị (Rendering Performance): Việc sử dụng Base64 Image (data:image/png) loại bỏ hoàn toàn độ trễ I/O ổ đĩa so với việc lưu file ảnh vật lý, phù hợp với các danh sách linh kiện dài cần load nhanh.
- Bảo mật dữ liệu (Data Security):
 - + Việc sử dụng tham số OracleDbType.Varchar2 khi truyền serial vào câu lệnh SQL là biện pháp phòng chống tấn công SQL Injection hiệu quả nhất. Hacker không thể chèn mã độc vào mã QR để phá hoại cơ sở dữ liệu.
 - + Attribute [Authorize] đảm bảo tuân thủ nguyên tắc Zero Trust: không tin tưởng bất kỳ request nào chưa được xác thực, dù đó là thao tác đọc dữ liệu đơn giản.

- Kiến trúc tách biệt (Layered Architecture): Code tách biệt rõ ràng giữa việc lấy dữ liệu (OracleHelper), xử lý logic (Controller) và hiển thị (View), giúp mã nguồn dễ bảo trì và mở rộng sau này.

4.12. Quản lý vai trò (Role)

4.12.1. Mục tiêu

Mục tiêu của phần này là xây dựng cơ chế phân quyền theo vai trò (RBAC – Role-Based Access Control) trong hệ thống quản trị cơ sở dữ liệu Oracle, nhằm quản lý quyền truy cập một cách tập trung, linh hoạt và dễ bảo trì. Việc tạo ra các vai trò giúp:

- Gom các quyền thực thi thủ tục theo từng chức năng thay vì cấp trực tiếp cho từng người dùng.
- Đảm bảo nguyên tắc “Ít quyền nhất” (Least Privilege)
- Dễ mở rộng, thay đổi khi có người dùng mới.
- Giảm thiểu sai sót bảo mật khi cấp quyền trực tiếp thủ công.

Trong hệ thống sửa chữa và bảo hành điện thoại, các vai trò chính bao gồm.

Bảng 4.6: Bảng chi tiết tham số của vai trò

Role	Quyền hạn chính
ROLE_ADMIN	Toàn quyền CRUD trên hệ thống ứng dụng.
ROLE_THUKHO	Quản lý nhập hàng, linh kiện
ROLE_TIEPTAN	Quản lý Đơn hàng, chi tiết đơn hàng, thông tin khách hàng, lịch hẹn..
ROLE_KITHUATVIEN	Quản lý Đơn hàng, yêu cầu linh kiện.
ROLE_KHACHHANG	Hẹn lịch, xem thông tin đơn hàng,...



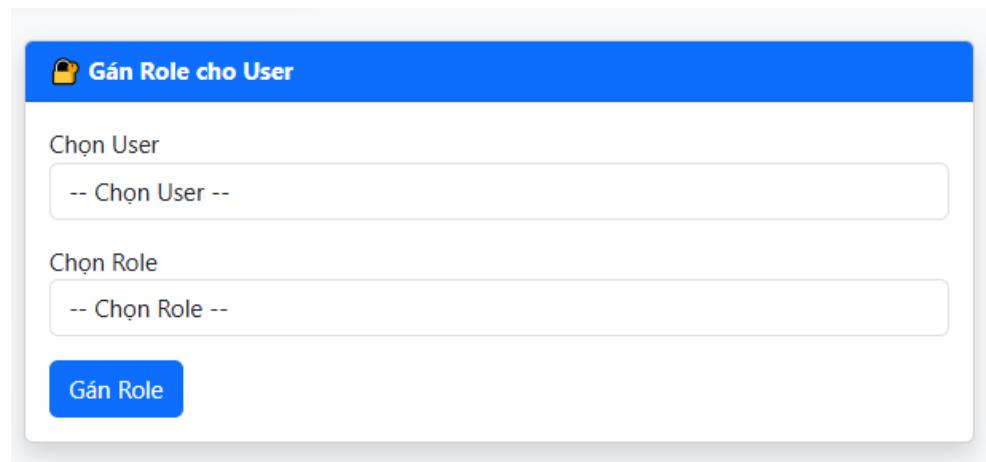
Hình 4.66: Vai trò (Role) của người dùng.

4.12.1.1 Gán vai trò cho người dùng

Gán vai trò cho người dùng giúp kích hoạt các chức năng đã được định nghĩa. Một người dùng chỉ có thể được gán một vai trò nhất định.

Thao tác trên giao diện web:

- Chọn người dùng: Quản trị viên phải chọn người dùng để gán vai trò.
- Chọn quyền: Sau khi quản trị viên chọn xong người dùng thì phải chọn vai trò từ danh sách/hộp thả xuống.
- Xác nhận gán vai trò cho người dùng đã chọn.



Hình 4.67: Gán vai trò.

Thao tác bằng lệnh SQL:

```
create or replace PROCEDURE ASSIGN_ROLE_PROC(
    p_username IN VARCHAR2,
    p_role_name IN VARCHAR2
)
AS
BEGIN
    EXECUTE IMMEDIATE 'GRANT "' || p_role_name || '" TO "' || p_username || '"';
END;
```

Hình 4.68: Thao tác bằng lệnh SQL của gán vai trò.

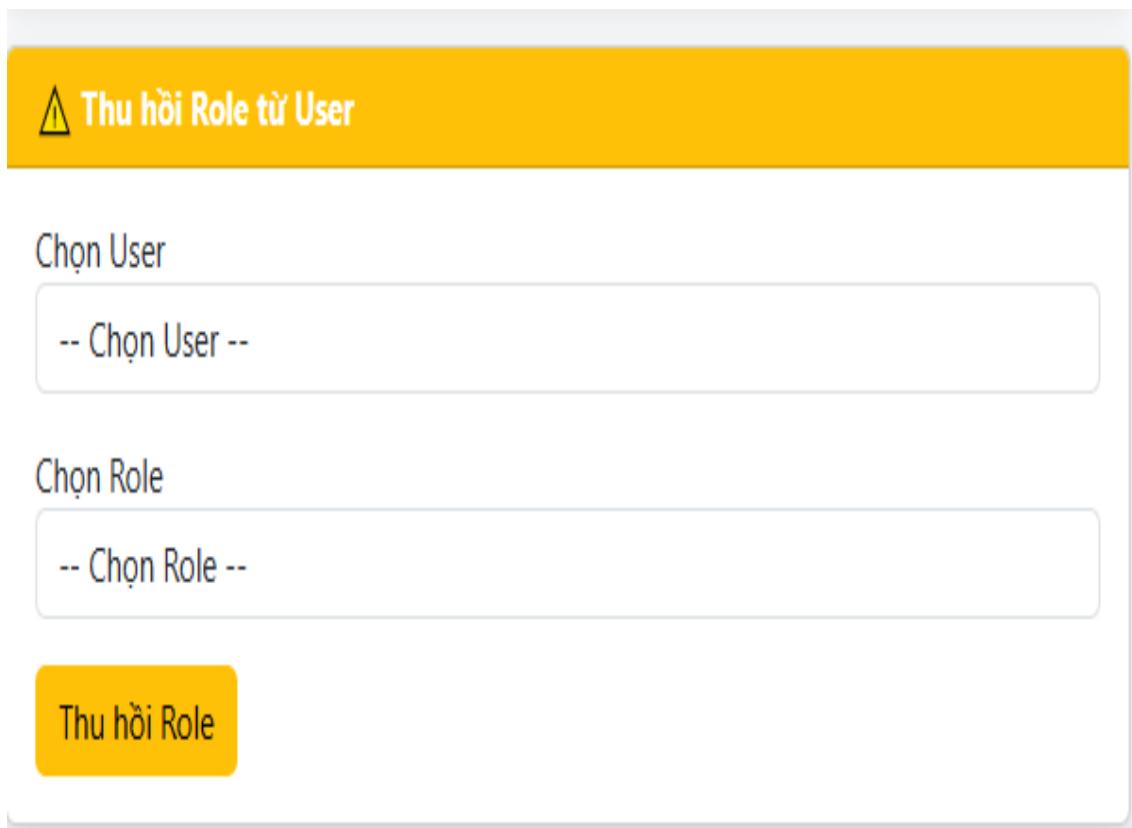
4.12.1.2 Thu hồi vai trò của người dùng

Thu hồi vai trò cho người dùng sẽ cấm người dùng vào các chức năng đã được định nghĩa.

Thao tác trên giao diện web:

- Chọn người dùng: Quản trị viên phải chọn người dùng để thu hồi vai trò.

- Chọn quyền: Sau khi quản trị viên chọn xong người dùng thì phải chọn vai trò từ danh sách/hộp thả xuống.
- Xác nhận thu hồi vai trò cho người dùng đã chọn.



Hình 4.69: Thu hồi vai trò.

Thao tác bằng lệnh SQL:

```
\create or replace PROCEDURE REVOKE_ROLE_PROC(
    p_username IN VARCHAR2,
    p_role_name IN VARCHAR2
)
AS
BEGIN
    EXECUTE IMMEDIATE 'REVOKE "' || p_role_name || '" FROM "' || p_username || '"';
END;
```

Hình 4.70: Thao tác bằng lệnh SQL của thu hồi vai trò.

4.13. Phân quyền, điều khiển truy cập (VPD)

4.13.1. Mục tiêu

Bảo vệ dữ liệu nhạy cảm theo vai trò: đảm bảo khách hàng chỉ xem được đơn hàng và lịch hẹn của chính họ, trong khi admin vẫn thấy toàn bộ dữ liệu.

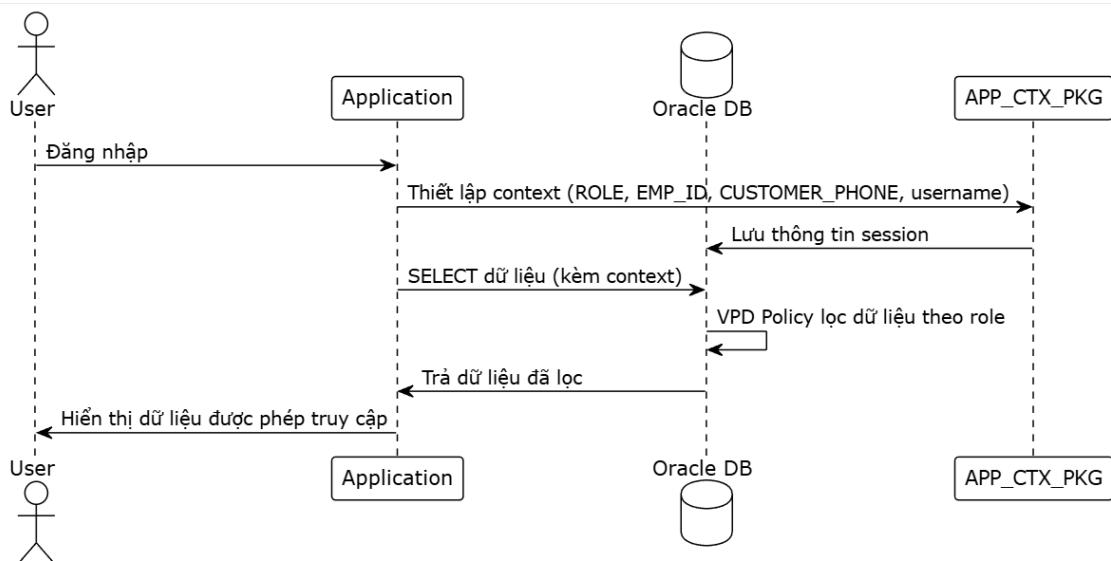
Giảm rủi ro rò rỉ dữ liệu từ tầng ứng dụng: kể cả khi API có bug hoặc bị gọi sai, chính Oracle sẽ lọc các dòng không thuộc phạm vi truy cập.

Đồng bộ audit & trách nhiệm: Oracle session lưu CLIENT_IDENTIFIER, ROLE, DEVICE, giúp đối chiếu truy vết giữa log ứng dụng và DB.

4.13.2. Mô tả bài toán

Trong hệ thống quản lý doanh nghiệp, dữ liệu như thông tin khách hàng, đơn hàng, lịch hẹn, phiếu xuất/nhập, hóa đơn... là cực kỳ nhạy cảm. Người dùng thuộc các vai trò khác nhau (admin, thủ kho, kỹ thuật viên, lễ tân, khách hàng) có quyền truy cập khác nhau. Nếu phân quyền chỉ thực hiện ở tầng ứng dụng, lỗi API hoặc thao tác sai có thể dẫn đến rò rỉ dữ liệu. Cần cơ chế lọc dữ liệu trực tiếp ở cơ sở dữ liệu và ghi nhận đầy đủ hành vi truy cập để phục vụ audit.

- Mỗi người dùng chỉ thấy dữ liệu được phép theo vai trò.
- Lọc dữ liệu ngay ở tầng cơ sở dữ liệu thông qua VPD policy.
- Audit log đầy đủ, bao gồm thông tin người dùng và thiết bị.
- Hệ thống dễ mở rộng cho các bảng và vai trò mới.



Hình 4.71: Sơ đồ luồng phân quyền, truy cập.

4.13.3. Các chức năng chính

Thông tin session người dùng được lưu vào Application Context, bao gồm role, id_employee, sdt_customer và username.

VPD policy sử dụng các giá trị trong context để lọc dữ liệu theo vai trò tương ứng.

Hệ thống kiểm soát truy cập trên nhiều bảng quan trọng như:
CUSTOMER_APPOINTMENT, CUSTOMER, EMPLOYEE, ORDERS, PART,
INVOICE, PART_REQUEST, STOCK_IN/OUT.

4.13.4. Luồng hoạt động

Khi người dùng đăng nhập, ứng dụng gọi package APP_CTX_PKG để thiết lập các giá trị context.

VPD policy sử dụng các giá trị này như: EMP_ID hoặc:
CUSTOMER_PHONE để lọc dữ liệu tương ứng với vai trò.

Audit log lưu trữ đầy đủ thông tin người dùng và thiết bị.

4.13.5. Phạm vi áp dụng

Bảng 4.7: Bảng phạm vi áp dụng của VPD.

TABLE	STATEMENT_TYPE	ADMIN	THUKHO	KITHUATVIEN	TIEPTAN	KHACHHANG
CUSTOMER_APPOINTMENT	Select	1=1	1=0	1=0	1=1	CUSTOMER_PHONE =: CUSTOMER_PHONE
CUSTOMER	Select	1=1	1=0	1=0	1=1	CUSTOMER_PHONE =: CUSTOMER_PHONE
EMPLOYEE	Select	=1	1=1	1=1	1=1	1=0

ORDERS	Select	=1	1=0	HANDLER_ EMP =: EMP_ID	1=0	CUSTOMER PHONE =: CUSTOMER _PHONE
PART	Select	=1	1=1	1=1	1=0	1=0
INVOICE, INVOICE_I TE	Select	=1	1=0	1=0	1=0	CUSTOMER PHONE =: CUSTOMER _PHONE
PART_REQ UEST, PART_REQ UEST_ITEM	Select	=1	1=1	HANDLER_ EMP =: EMP_ID	1=0	1=0
STOCK_IN, STOCK_IN ITEM, STOCK_OU T, STOCK_OU T_ITEM	Select	=1	=1	1=0	1=0	1=0

4.13.6. Cài đặt và áp dụng VPD

4.13.6.1 Xây dựng hàm chính sách (Policy Function)

Hệ thống sử dụng SYS_CONTEXT để lấy thông tin người dùng hiện hành. Dựa trên vai trò (ROLE) và mã định danh (ID/Phone), hàm chính sách sẽ trả về chuỗi điều kiện (predicate) phù hợp để lọc dữ liệu ở cấp độ dòng (Row-level security).

```

CREATE OR REPLACE FUNCTION ORDERS_VPD_PREDICATE(
    p_schema IN VARCHAR2,
    p_object IN VARCHAR2
) RETURN VARCHAR2
AS
    v_role VARCHAR2(100) := SYS_CONTEXT('APP_CTX','ROLE_NAME');
    v_emp VARCHAR2(100) := SYS_CONTEXT('APP_CTX','EMP_ID');
    v_cus VARCHAR2(100) := SYS_CONTEXT('APP_CTX','CUSTOMER_PHONE');
BEGIN
    IF v_role IS NULL THEN
        RETURN '1=0';
    END IF;

    IF v_role IN ('ROLE_ADMIN','ROLE_TIEPTAN') THEN
        RETURN '1=1';
    END IF;

    IF v_role = 'ROLE_KITHUATVIEN' THEN
        IF v_emp IS NULL THEN
            RETURN '1=0';
        END IF;
        RETURN 'HANDLER_EMP = ' || TO_NUMBER(v_emp);
    END IF;

    IF v_role = 'ROLE_KHACHHANG' THEN
        IF v_cus IS NULL THEN
            RETURN '1=0';
        END IF;
        RETURN 'CUSTOMER_PHONE = ' || REPLACE(v_cus,'***','*****') || '***';
    END IF;

    IF v_role = 'ROLE_THUKHO' THEN
        RETURN '1=0';
    END IF;

    RETURN '1=0';

```

Hình 4.72: Mã nguồn hàm chính sách kiểm soát truy cập đơn hàng.

```

BEGIN
    DBMS_RLS.ADD_POLICY(
        object_schema => USER,
        object_name   => 'ORDERS',
        policy_name   => 'ORDERS_VPD',
        function_schema => USER,
        policy_function => 'ORDERS_VPD_PREDICATE',
        statement_types => 'SELECT',
        update_check   => FALSE,
        enable         => TRUE
    );
END;
/

```

Hình 4.73: Cấu hình chính sách VPD (Virtual Private Database) cho bảng ORDERS

4.14. Mã hóa dữ liệu

4.14.1. Mã hóa mật khẩu người dùng cơ sở dữ liệu

Hệ thống không sử dụng mật khẩu người dùng ở dạng thuần (plaintext) để làm mật khẩu cơ sở dữ liệu. Thay vào đó, mật khẩu cơ sở dữ liệu được tạo ra từ mật khẩu người dùng thông qua cơ chế mã hóa nhằm đảm bảo bảo mật kết nối đến cơ sở dữ liệu.

Phương pháp sử dụng trong hệ thống là HASH_PASSWORD_20CHARS, hoạt động theo các bước sau:

- Sinh chuỗi hash từ mật khẩu gốc bằng thuật toán băm (SHA-256 hoặc tương đương).
- Trích xuất 20 ký tự từ chuỗi hash:
 - + 10 ký tự đầu
 - + 10 ký tự cuối

Ghép hai phần này thành chuỗi hash rút gọn 20 ký tự. Đây là giá trị chính là mật khẩu người dùng cơ sở dữ liệu.

Phương pháp này đảm bảo:

- Mật khẩu cơ sở dữ liệu được tạo động từ mật khẩu người dùng, tăng tính bảo mật.
- Hạn chế rủi ro lộ mật khẩu cơ sở dữ liệu do không lưu trữ trực tiếp.

4.14.2. Mã hóa dữ liệu đường truyền

4.14.2.1 Phạm vi

Áp dụng cho một vài API truyền dữ liệu quan trọng giữa Client và Server, bao gồm:

- Đăng nhập, đăng ký, đổi mật khẩu.
- Tạo phiếu nhập, phiếu xuất, hóa đơn.

Đảm bảo mã hóa payload 2 chiều và ký số tài liệu (PDF, báo cáo nghiệp vụ).

Mục tiêu: bảo mật dữ liệu truyền tải, tránh rò rỉ thông tin, đảm bảo integrity và confidentiality.

4.14.2.2 Luồng hoạt động

```
public async Task<TRes> PostEncryptedAndGetEncryptedAsync<TReq, TRes>(string path, TReq request)
{
    if (_serverPublicKeyBase64 == null || _clientPrivateKeyBase64 == null)
        throw new InvalidOperationException("SecurityClient chưa được khởi tạo.");

    // 1) Serialize + mã hóa request
    var json = JsonSerializer.Serialize(request);
    var enc = EncryptHelper.HybridEncrypt(json, _serverPublicKeyBase64);

    // 2) Gửi request (chi gửi key mã hóa + cipher)
    var res = await _httpClient.PostAsJsonAsync(path, new
    {
        encryptedKeyBlockBase64 = enc.EncryptedKeyBlock,
        cipherDataBase64 = enc.CipherData
    });

    // 3) Nếu HTTP trả lỗi -> throw ngay
    if (!res.IsSuccessStatusCode)
        throw new InvalidOperationException(
            await res.Content.ReadAsStringAsync()
            ?? $"HTTP {(int)res.StatusCode} {res.ReasonPhrase}"
        );

    // 4) Đọc envelope (response đã mã hóa)
    var api = await res.Content.ReadFromJsonAsync<ApiResponse<Envelope>>()
        ?? throw new InvalidOperationException("Không đọc được Envelope từ server.");
    if (!api.Success || api.Data == null)
        throw new InvalidOperationException(api.Error ?? "Envelope không hợp lệ.");

    // 5) Giải mã envelope để lấy JSON thật
    var decrypted = EncryptHelper.HybridDecrypt(
        api.Data.EncryptedKeyBlockBase64!,
        api.Data.CipherDataBase64!,
        _clientPrivateKeyBase64);

    // 6) Deserialize JSON thành ApiResponse<TRes>
    var apiRes = JsonSerializer.Deserialize<ApiResponse<TRes>>(decrypted);
    if (apiRes?.Success == true && apiRes.Data != null)
        return apiRes.Data;

    // 7) Trường hợp server trả về TRes trực tiếp (không bọc ApiResponse)
    var direct = JsonSerializer.Deserialize<TRes>(decrypted);
    if (direct != null)
        return direct;

    throw new InvalidOperationException($"Response không hợp lệ. JSON: {decrypted}");
}
```

Trao đổi public key RSA giữa Client và Server:

Client lấy public key của Server:

GET /api/Public/Security/server-public-key

Client đăng ký public key của mình lên Server:

POST /api/Public/Security/register-client-key

Lưu ý: Nếu đã đăng nhập bằng tài khoản nhân viên thì Client đăng ký public key của mình lên Server là public key cá nhân được lưu ở cơ sở dữ liệu.

Tài khoản khách hàng và chưa đăng nhập thì public key và private key sẽ được tự động tạo.

A: Khi Client gửi dữ liệu lên Server

- Client thực hiện:
 - + Sinh ngẫu nhiên AES key 256 bit + IV 128 bit
 - + Mã hóa dữ liệu (JSON, ...) bằng AES

- + Ghép key + IV, mã hóa bằng RSA public key Server → EncryptedKeyBlockBase64
- + Gửi API body: { EncryptedKeyBlockBase64, CipherDataBase64 }

```

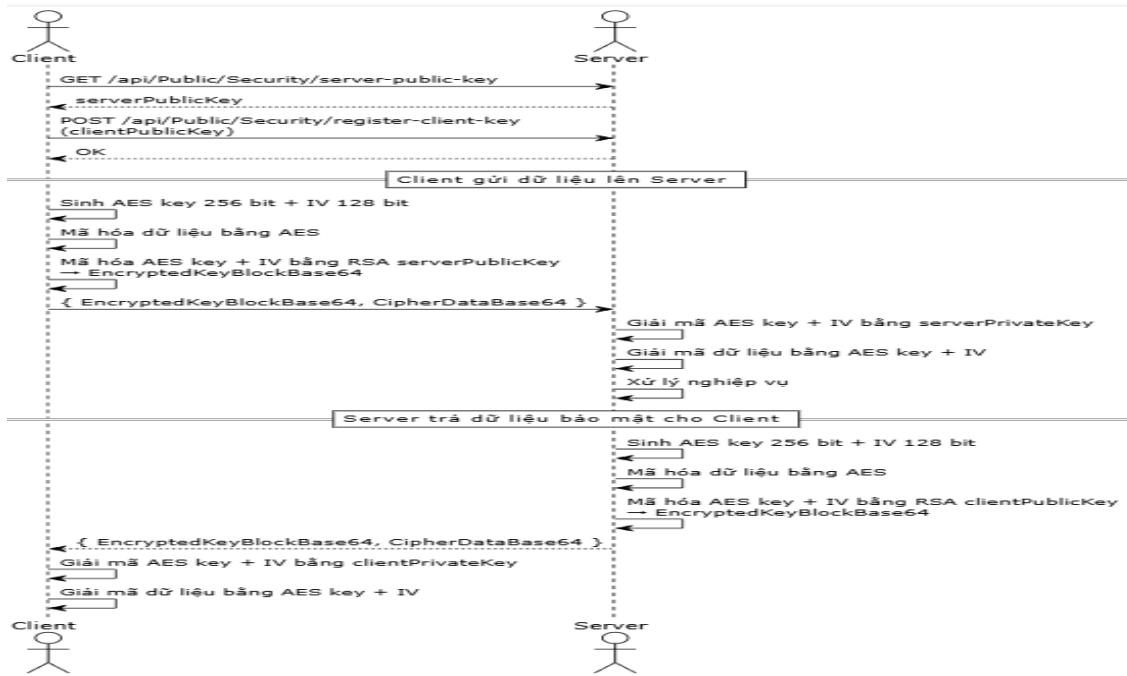
2 references
public static (string EncryptedKeyBlock, string CipherData) HybridEncrypt(string plaintext, string publicKeyBase64)
{
    // Tạo AES Key + IV
    using (Aes aes = Aes.Create())
    {
        aes.GenerateKey();
        aes.GenerateIV();
        byte[] key = aes.Key;
        byte[] iv = aes.IV;

        // Encrypt dữ liệu bằng AES
        ICryptoTransform encryptor = aes.CreateEncryptor(key, iv);
        byte[] dataBytes = Encoding.UTF8.GetBytes(plaintext);
        byte[] cipherBytes;
        using (var ms = new MemoryStream())
        using (var cs = new CryptoStream(ms, encryptor, CryptoStreamMode.Write))
        {
            cs.Write(dataBytes, 0, dataBytes.Length);
            cs.FlushFinalBlock();
            cipherBytes = ms.ToArray();
        }
    }
}

```

Hình 4.74: Mã nguồn mã hóa RSA/AES

- Server thực hiện:
 - + Giải mã AES key + IV bằng private key Server
 - + Dùng AES key + IV giải mã dữ liệu
 - + Xử lý nghiệp vụ
- B: Khi Server trả dữ liệu bảo mật cho Client
 - Server thực hiện:
 - + Sinh AES key + IV mới
 - + Mã hóa dữ liệu bằng AES
 - + Ghép key + IV, mã hóa bằng RSA public key Client
 - + Trả API body: { EncryptedKeyBlockBase64, CipherDataBase64 }
 - Client thực hiện:
 - + Giải mã AES key + IV bằng private key Client
 - + Giải mã dữ liệu để đọc nội dung API response



Hình 4.75: Sơ đồ quy trình mã hóa giải mã dữ liệu đường truyền.

4.15. Kiểm toán và giải trình (Standard Audit, Trigger)

4.15.1. Cơ sở lý thuyết

Standard Audit

- Standard Audit là cơ chế kiểm toán cơ bản của hệ quản trị cơ sở dữ liệu, ghi lại các sự kiện quan trọng như đăng nhập, đăng xuất, thay đổi quyền hạn, tạo/xóa đối tượng, và các thao tác dữ liệu.
- Dữ liệu audit được lưu trong các bảng hệ thống và có thể được trích xuất để phân tích, lập báo cáo hoặc phục vụ điều tra khi có sự cố.
- Standard Audit cung cấp một cái nhìn tổng quan về hoạt động của người dùng và hệ thống, giúp đánh giá tuân thủ chính sách bảo mật và hỗ trợ việc quản lý rủi ro.

Trigger

- Thiết lập hệ thống giám sát hành động quan trọng trong CSDL, đảm bảo tính giải trình (Accountability) và chống chối bỏ (Non-repudiation).
- Theo dõi chính xác người dùng thực hiện (người dùng Oracle, client identifier, vai trò ứng dụng, nhân viên liên quan).
- Lưu giá trị cũ/mới, cột thay đổi, và các thông tin bổ sung (máy, module, note) để phục vụ truy vết và kiểm toán.

4.15.2. Triển khai Trigger trong Cơ sở dữ liệu

Bước 1: Tạo table audit_alert_log chứa các thông tin kiểm toán

Bảng 4.8: Tham số bảng thông tin kiểm toán.

Cột	Ý nghĩa
log_id	Khóa chính, tự tăng
event_ts	Thời gian sự kiện xảy ra
db_user	Tài khoản CSDL thực hiện thao tác
os_user	Người dùng hệ điều hành
machine	Máy thực hiện thao tác
module	Tên module/ứng dụng gọi thao tác
app_role	Vai trò trong ứng dụng (ví dụ: THUKHO, ADMIN)
emp_id	Mã nhân viên liên quan
customer_phone	Thông tin khách hàng liên quan (nếu có)
object_schema	Bảng/tài nguyên bị thao tác
object_name	Bảng/tài nguyên bị thao tác
Cột	Ý nghĩa
policy_name	Policy kiểm toán đang áp dụng
client_identifier	Username từ session (oci_set_client_identifier)
dml_type	Loại thao tác: INSERT, UPDATE, DELETE
old_values	JSON giá trị cũ (trước khi thay đổi)
new_values	JSON giá trị mới (sau khi thay đổi)
changed_columns	Danh sách cột bị thay đổi, phân tách bằng dấu phẩy
note	Ghi chú bổ sung

Bước 2: Tạo Trigger gắn vào bảng nghiệp vụ

- Trigger được gắn vào bảng cần giám sát
- Kích hoạt AFTER INSERT, UPDATE, DELETE.
- Trigger sẽ thực hiện các công việc sau:
 - + Xác định ngữ cảnh
 - + Đọc SYS_CONTEXT('USERENV','CLIENT_IDENTIFIER') để lấy username thật của người dùng nghiệp vụ.
 - + Thu thập các thông tin môi trường: db_user, os_user, machine, module.
 - + Ghi lại thao tác DML
 - + dml_type xác định thao tác: INSERT/UPDATE/DELETE.
 - + Với UPDATE: xác định old_values và new_values theo JSON.

- + Xác định changed_columns (các cột bị thay đổi).
- + Lưu log vào bảng audit_alert_log
- + Trigger thực hiện INSERT vào audit_log với đầy đủ thông tin: thời gian, người thực hiện, dữ liệu cũ/mới, cột thay đổi, chính sách áp dụng, vai trò ứng dụng, ghi chú.

Bước 3: Truy xuất và giải trình

- Gọi API hoặc dashboard gọi dữ liệu từ audit_log.
- Cho phép lọc theo:
 - + emp_id / client_identifier (người thao tác)
 - + object_name / object_schema (bảng/tài nguyên)
 - + policy_name (loại policy kiểm toán)
 - + Khoảng thời gian
- Dữ liệu truy xuất gồm cả JSON giá trị cũ/mới để dễ dàng kiểm tra và giải trình.

4.15.3. Triển khai Standard Audit trong Cơ sở dữ liệu

- Bước 1: Kiểm tra trạng thái audit.

```
SHOW PARAMETER audit_trail;
```

- Nếu audit chưa bật, bật chế độ DB + EXTENDED để ghi chi tiết thao tác, bao gồm cả SQL và bind variables:

```
ALTER SYSTEM SET audit_trail = DB, EXTENDED SCOPE=SPFILE;
```

- Bước 2: Tạo policy kiểm toán

-- EMPLOYEE
<i>AUDIT INSERT, UPDATE, DELETE ON EMPLOYEE BY ACCESS;</i>
-- CUSTOMER
<i>AUDIT INSERT, UPDATE, DELETE ON CUSTOMER BY ACCESS;</i>
-- ORDERS
<i>AUDIT INSERT, UPDATE, DELETE ON ORDERS BY ACCESS;</i>
-- STOCK_IN
<i>AUDIT INSERT, UPDATE, DELETE ON STOCK_IN BY ACCESS;</i>
-- STOCK_IN_ITEM
<i>AUDIT INSERT, UPDATE, DELETE ON STOCK_IN_ITEM BY ACCESS;</i>

-- PART
<i>AUDIT INSERT, UPDATE, DELETE ON PART BY ACCESS;</i>
-- STOCK_OUT
<i>AUDIT INSERT, UPDATE, DELETE ON STOCK_OUT BY ACCESS;</i>
-- STOCK_OUT_ITEM
<i>AUDIT INSERT, UPDATE, DELETE ON STOCK_OUT_ITEM BY ACCESS;</i>
-- PART_REQUEST
<i>AUDIT INSERT, UPDATE, DELETE ON PART_REQUEST BY ACCESS;</i>
-- PART_REQUEST_ITEM
<i>AUDIT INSERT, UPDATE, DELETE ON PART_REQUEST_ITEM BY ACCESS;</i>
-- USER_OTP_LOG
<i>AUDIT INSERT, UPDATE, DELETE ON USER_OTP_LOG BY ACCESS;</i>
-- EMPLOYEE_SHIFT
<i>AUDIT INSERT, UPDATE, DELETE ON EMPLOYEE_SHIFT BY ACCESS;</i>
-- CUSTOMER_APPOINTMENT
<i>AUDIT INSERT, UPDATE, DELETE ON CUSTOMER_APPOINTMENT BY ACCESS;</i>
-- INVOICE
<i>AUDIT INSERT, UPDATE, DELETE ON INVOICE BY ACCESS;</i>
-- INVOICE_ITEM
<i>AUDIT INSERT, UPDATE, DELETE ON INVOICE_ITEM BY ACCESS;</i>
-- SERVICE
<i>AUDIT INSERT, UPDATE, DELETE ON SERVICE BY ACCESS;</i>
-- ORDER_SERVICE
<i>AUDIT INSERT, UPDATE, DELETE ON ORDER_SERVICE BY ACCESS;</i>
-- INVOICE_SERVICE
<i>AUDIT INSERT, UPDATE, DELETE ON INVOICE_SERVICE BY ACCESS;</i>

– Bước 3: Truy xuất và giải trình:

Gọi API hoặc dashboard gọi dữ liệu từ cơ sở dữ liệu:

Thời gian	DB User	Schema	Object	Action	SQL Text
23/11/2025 09:20:58	ADMIN	APP	INVOICE	UPDATE	

23/11/2025 09:20:57	ADMIN	APP	STOCK_OUT	UPDATE	
23/11/2025 09:20:36	ADMIN	APP	ORDERS	UPDATE	

4.16. Sao lưu và phục hồi cơ sở dữ liệu

4.16.1. Mục tiêu

Mục tiêu là thiết lập một chiến lược sao lưu (backup) và phục hồi (recovery) đáng tin cậy cho hệ thống cơ sở dữ liệu Oracle của ứng dụng bảo hành điện thoại.

Trong một hệ thống nghiệp vụ, dữ liệu là tài sản cốt lõi. Các sự cố như hỏng hóc phần cứng (mất ổ đĩa), lỗi logic của ứng dụng, hoặc sai sót của con người (vô tình xóa bảng) có thể dẫn đến mất mát dữ liệu, gây trì hoãn kinh doanh và thiệt hại tài chính nghiêm trọng.

4.16.2. Chiến lược sao lưu tự động (Backup Strategy)

Quy trình sao lưu được thiết kế để vận hành tự động thông qua sự phối hợp giữa Oracle Job Scheduler và các Shell Script trên hệ điều hành Linux.

- Cơ chế: Thực hiện sao lưu toàn bộ (Full Backup) kết hợp sao lưu Archive Logs nhằm đảm bảo không mất mát dữ liệu giao dịch.
- Lịch trình: Job RUN_RMAN_PDB_BACKUP được cấu hình chạy định kỳ vào lúc 02:00 AM hàng ngày. Đây là khung giờ hệ thống có lưu lượng truy cập thấp nhất, giúp giảm thiểu ảnh hưởng đến hiệu năng phục vụ người dùng.
- Chính sách lưu trữ (Retention Policy): Áp dụng "Recovery Window of 7 days". Hệ thống tự động xóa các bản sao lưu và log cũ hơn 7 ngày để tối ưu hóa dung lượng ổ cứng.

```

6
7 # PDB cần backup
8 PDB_HOST=localhost
9 PDB_PORT=1521
10 PDB_SERVICE=ORCLPDB1
11 RMAN_USER=rman
12 RMAN_PWD=rmanpwd
13

```

Hình 4.734.76: Cấu hình tham số kết nối trong script sao lưu tự động.

```

#!/bin/bash

# Thiết lập môi trường Oracle
export ORACLE_HOME=/opt/oracle/product/21c/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH

# PDB cần backup
PDB_HOST=localhost
PDB_PORT=1521
PDB_SERVICE=ORCLPDB1
RMAN_USER=rman
RMAN_PWD=rmanpwd

# Thư mục log
LOG_DIR="/home/oracle/backup_web/logs"
mkdir -p "${LOG_DIR}"
LOG_FILE="${LOG_DIR}/rman_backup_$(date +%F_%H%M%S).log"

# Chạy RMAN backup PDB + archivelog bằng username/password via EZCONNECT
$ORACLE_HOME/bin/rman target ${RMAN_USER}/${RMAN_PWD}@//${PDB_HOST}:${PDB_PORT}/${PDB_SERVICE} <<EOF > ${LOG_FILE} 2>&1
RUN {
    BACKUP DATABASE PLUS ARCHIVELOG;
}
EOF

# Kiểm tra exit code
RET=$?
if [ $RET -ne 0 ]; then
    echo "RMAN FAILED with code $RET. Kiểm tra log: ${LOG_FILE}" >&2
else
    echo "RMAN backup of ${PDB_SERVICE} completed successfully. Log: ${LOG_FILE}"
fi

exit $RET

```

```

#!/bin/bash

# -----
# Script RMAN Restore PDB
# -----


export ORACLE_HOME=/opt/oracle/product/21c/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
export ORACLE_SID=ORCLCDB

PDB_SERVICE=ORCLPDB1
POINTOFTIME="$1"

LOG_DIR="/home/oracle/backup_web/logs"
mkdir -p ${LOG_DIR}
LOG_FILE="${LOG_DIR}/rman_restore_$(date +%F_%H%M%S).log"

echo "====> Closing PDB ${PDB_SERVICE}..."
sqlplus -s / as sysdba <<EOF
ALTER PLUGGABLE DATABASE ${PDB_SERVICE} CLOSE IMMEDIATE;
EOF

echo "====> Running RMAN restore..."
if [ -z "$POINTOFTIME" ]; then
    # Restore latest backup
    rman target / <<EOF >> ${LOG_FILE} 2>&1
RUN {
    RESTORE PLUGGABLE DATABASE ${PDB_SERVICE};
    RECOVER PLUGGABLE DATABASE ${PDB_SERVICE};
}
EOF

    echo "====> Opening PDB ${PDB_SERVICE} (no RESETLOGS)..."
    sqlplus -s / as sysdba <<EOF
ALTER PLUGGABLE DATABASE ${PDB_SERVICE} OPEN;
EOF
else
    # PITR restore
    rman target / <<EOF >> ${LOG_FILE} 2>&1
RUN {
    SET UNTIL TIME "to_date('${POINTOFTIME}', 'YYYY-MM-DD HH24:MI:SS')";
    RESTORE PLUGGABLE DATABASE ${PDB_SERVICE};
    RECOVER PLUGGABLE DATABASE ${PDB_SERVICE};
}
EOF

    echo "====> Opening PDB ${PDB_SERVICE} with RESETLOGS..."
    sqlplus -s / as sysdba <<EOF
ALTER PLUGGABLE DATABASE ${PDB_SERVICE} OPEN RESETLOGS;
EOF
fi

RET=$?
if [ $RET -ne 0 ]; then
    echo "✖ RESTORE FAILED - check log: ${LOG_FILE}"
else
    echo "✔ RESTORE SUCCESS - log: ${LOG_FILE}"
fi

exit $RET

```

CHƯƠNG 5: THỬ NGHIỆM VÀ TRIỂN KHAI

5.1. Kịch bản kiểm thử

5.1.1. Thử nghiệm kết nối

Để kiểm chứng khả năng vận hành của hệ thống trong môi trường mạng thực tế, nhóm đã thực hiện hai kịch bản kiểm thử kết nối từ máy trạm (Client) đến máy chủ cơ sở dữ liệu (Server).

Kịch bản 1: Kiểm thử độ trễ và thông mạng (Ping Test)

Kịch bản này nhằm xác nhận đường truyền VPN giữa máy Client (Windows) và Server (Linux) đã được thiết lập thành công.

Wireless LAN adapter Wi-Fi:

```
Connection-specific DNS Suffix . :  
IPv6 Address . . . . . : 2402:800:63b9:8f30:2052:7055:6f82:153a  
Temporary IPv6 Address . . . . . : 2402:800:63b9:8f30:bd3c:11e7:8f37:8a4  
Link-local IPv6 Address . . . . . : fe80::82d4:2744:6190:3be5%17  
IPv4 Address . . . . . : 192.168.1.11  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : fe80::1%17  
192.168.1.1
```

Hình 5.1: Ip của máy client.

```
valid_lft forever preferred_lft forever  
3: ztrfykhu4r: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2800 qdisc fq_codel state UNKNOWN group default qlen 1000  
link/ether 1a:91:e6:0b:f1:da brd ff:ff:ff:ff:ff:ff  
inet 10.147.20.157/24 brd 10.147.20.255 scope global ztrfykhu4r  
    valid_lft forever preferred_lft forever  
    inet6 fe80::1891:e6ff:fe0b:f1da/64 scope link  
        valid_lft forever preferred_lft forever  
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000  
    link/ether 52:54:00:8c:54:aa brd ff:ff:ff:ff:ff:ff  
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0  
        valid_lft forever preferred_lft forever  
[ntd@localhost ~]$
```

Hình 5.2: Ip ZeroTier của Server Linux.

- Thao tác: Từ Command Prompt (CMD) trên Windows, thực hiện lệnh ping tới địa chỉ IP ZeroTier của Server Linux.
- Kết quả: Gói tin được gửi và nhận thành công với độ trễ trung bình thấp (<100ms), không xảy ra hiện tượng mất gói tin (Packet Loss).

```
C:\Users\ASUS>ping 10.147.20.157
Pinging 10.147.20.157 with 32 bytes of data:
Reply from 10.147.20.157: bytes=32 time=32ms TTL=64
Reply from 10.147.20.157: bytes=32 time=9ms TTL=64
Reply from 10.147.20.157: bytes=32 time=8ms TTL=64
Reply from 10.147.20.157: bytes=32 time=2ms TTL=64

Ping statistics for 10.147.20.157:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 32ms, Average = 12ms
```

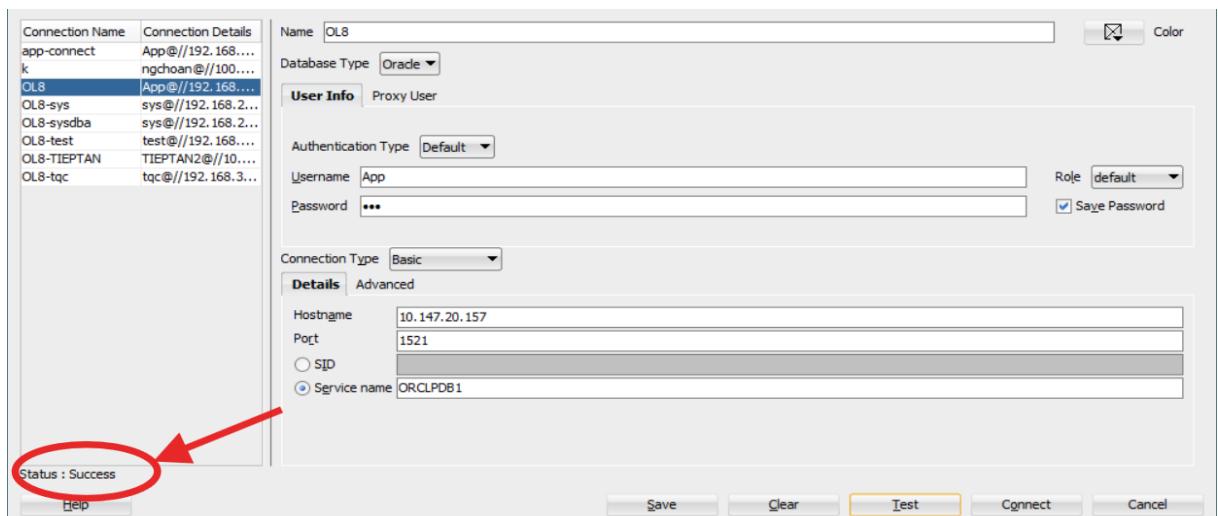
Hình 5.3: Kết quả kiểm tra kết nối mạng VPN thành công.

Kịch bản 2: Kiểm thử kết nối Ứng dụng (SQL Connectivity)

Kịch bản này nhằm xác nhận Client có thể kết nối tới cơ sở dữ liệu Oracle thông qua lớp mạng ảo, phục vụ cho việc triển khai ứng dụng quản lý.

- Công cụ: Oracle SQL Developer.
- Thông số kết nối:
 - + Hostname: 10.147.20.157
 - + Port: 1521
 - + Service Name: ORCLPDB1.
 - + User: App

Kết quả: Trạng thái kết nối báo Success. Hệ thống sẵn sàng tiếp nhận truy vấn từ ứng dụng.



Hình 5.4: Kết nối thành công qua IP ZeroTier.

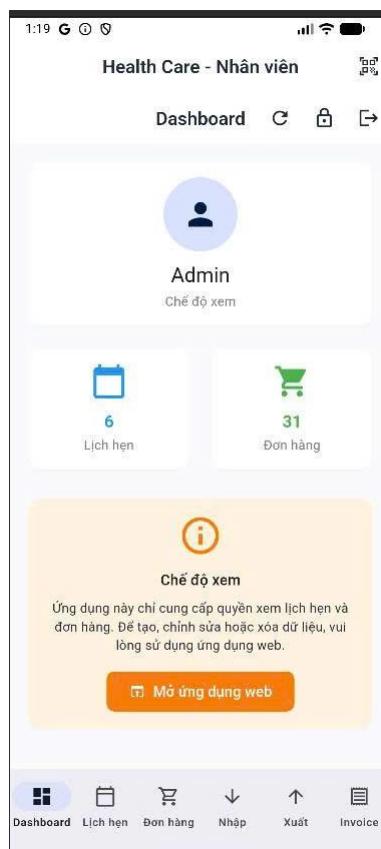
5.1.2. Đăng nhập đa nền tảng

- Thao tác: đăng nhập người dùng Admin ở hai nền tảng là web và mobile
- Tại giao diện web, kiểm tra thông tin phiên làm việc hiện tại.

- Mở ứng dụng trên điện thoại (Mobile App), đăng nhập bằng cùng tài khoản trên.



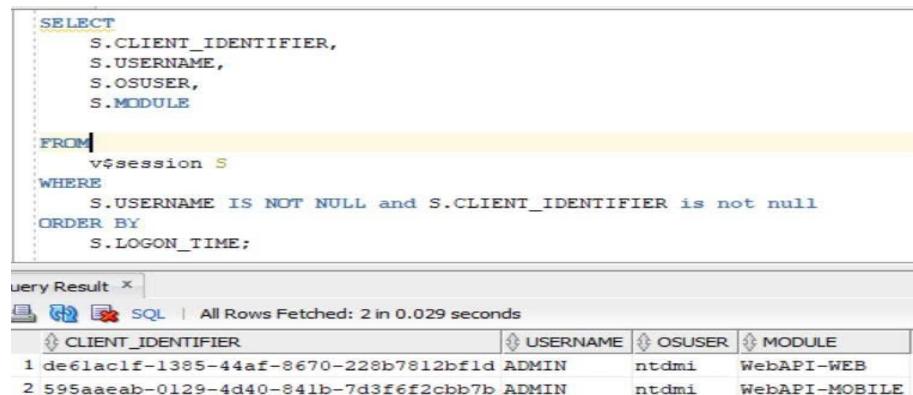
Hình 5.5: Thông tin phiên làm việc trên web.



Hình 5.6: Đăng nhập cùng tài khoản trên mobile.

- Kết quả mong đợi:
 - + Tài khoản đăng nhập thành công trên cả 2 thiết bị.
 - + Sau khi đăng nhập Mobile, session trên Web KHÔNG bị vắng (Logout). Cả hai thiết bị đều có thể thao tác dữ liệu bình thường.

- Kết quả thực tế:



```

SELECT
    S.CLIENT_IDENTIFIER,
    S.USERNAME,
    S.OSUSER,
    S MODULE
FROM
    v$session S
WHERE
    S.USERNAME IS NOT NULL AND S.CLIENT_IDENTIFIER IS NOT NULL
ORDER BY
    S.LOGON_TIME;

```

Query Result x

CLIENT_IDENTIFIER	USERNAME	OSUSER	MODULE
1 de61ac1f-1385-44af-8670-228b7812bf1d	ADMIN	ntdmi	WebAPI-WEB
2 595aaeab-0129-4d40-841b-7d3f6f2cbb7b	ADMIN	ntdmi	WebAPI-MOBILE

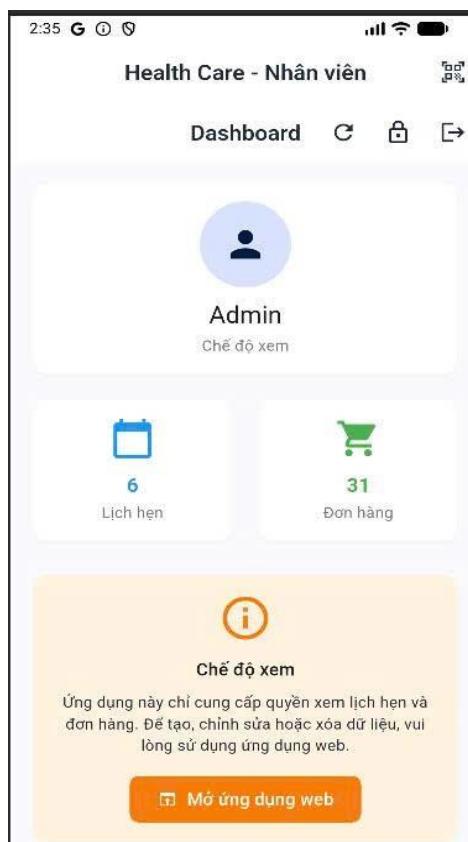
Hình 5.7: Kết quả thực tế khi đăng nhập 2 nền tảng

5.1.3. Xác thực đa nền tảng bằng mã QR

5.1.3.1 Kịch bản 1: Xác thực web từ mobile

- Thao tác

- + Đăng nhập tài khoản bất kỳ trên nền tảng mobile
- + Mở chức năng quét đăng nhập web bằng QR
- + Thực hiện quét mã QR hiện trên trang đăng nhập trên nền tảng web.



Hình 5.8: Giao diện trang chủ của nhân viên trên mobile



Hình 5.9: Mở camera để bắt đầu quét QR trên web

Đăng nhập Admin bằng ứng dụng di động (nhân viên)

Mở app Mobile Service (đăng nhập nhân viên) > Chọn “Đăng nhập Web” > nhập mã dưới đây.

Nhập mã này trên app Mobile Service (nhân viên) để đăng nhập Admin.



Hình 5.10: Mã QR trên web.

- Kết quả mong đợi:
 - + Mobile xác thực QR và cho phép nền tảng web đăng nhập với người dùng sẵn có
- Kết quả thực tế:

Hình 5.11: Thông tin phiên làm việc trên nền tảng web

```

SELECT
    S.CLIENT_IDENTIFIER,
    S.USERNAME,
    S.OSUSER,
    S MODULE
FROM
    v$session S
WHERE
    S.USERNAME IS NOT NULL AND S.CLIENT_IDENTIFIER IS NOT NULL
ORDER BY
    S.LOGON_TIME;

```

CLIENT_IDENTIFIER	USERNAME	OSUSER	MODULE
b8f1e5b8-6866-450f-92da-ddaabce3722b	ADMIN	ntdmi	WebAPI-MOBILE
2fc3780f-7784-4ee6-9c6c-aedbf9801ac2	ADMIN	ntdmi	WebAPI-WEB

Hình 5.12: Kiểm tra phiên làm việc ở cơ sở dữ liệu.

5.1.3.2 Kịch bản 2: Xác thực Mobile từ web

- Thao tác
 - + Thực hiện đăng nhập tài khoản bất kỳ
 - + Mở chức năng tạo mã QR đăng nhập mobile trên web

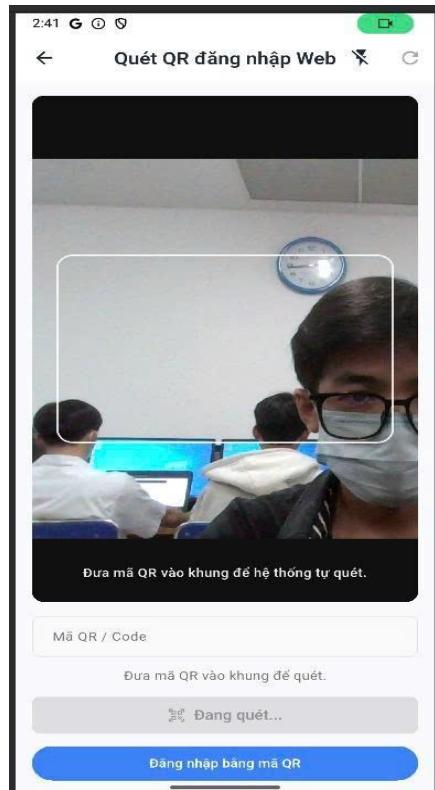
- + Mở chức năng quét QR đăng nhập ở trang login điện thoại, có thể nhập QR code thay thế nếu camera hư.



Hình 5.13: Mã QR để đăng nhập vào ứng dụng mobile

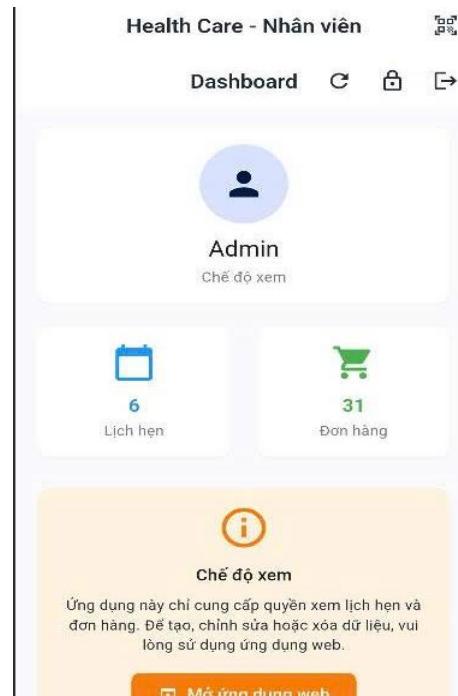


Hình 5.14: Chọn đăng nhập bằng tài khoản nhân viên



Hình 5.15: Mở camera quét mã QR để đăng nhập

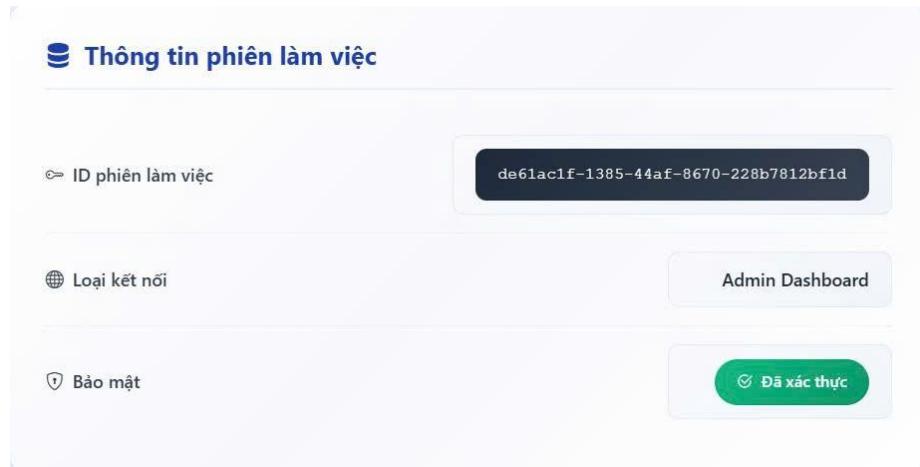
- kết quả mong đợi: bên mobile sẽ đăng nhập với session mới
- kết quả thực tế:



Hình 5.16: Đăng nhập thành công

5.1.4. Đăng nhập tranh chấp trên web

- Thao tác:
 - + đăng nhập một tài khoản bất kì trên một trình duyệt.
 - + kiểm tra thông tin phiên làm việc.
 - + mở trình duyệt thứ hai, đăng nhập cùng tài khoản



Hình 5.17: Thông tin phiên làm việc đầu tiên.



Hình 5.18: Thông tin phiên làm việc thứ hai.

- Kết quả mong đợi: Phiên làm việc ở trình duyệt đầu tiên sẽ bị hủy và thế bằng trình duyệt thứ hai
- Kết quả thực tế:
 - + Trước khi thực hiện đăng nhập trình duyệt khác, session tồn tại trên cơ sở dữ liệu de61ac1f.....

```

SELECT
    S.CLIENT_IDENTIFIER,
    S.USERNAME,
    S.OSUSER,
    S MODULE

FROM
    v$session S
WHERE
    S.USERNAME IS NOT NULL and S.CLIENT_IDENTIFIER is not null
ORDER BY
    S.LOGON_TIME;

```

Query Result

All Rows Fetched: 2 in 0.029 seconds

CLIENT_IDENTIFIER	USERNAME	OSUSER	MODULE
de61ac1f-1385-44af-8670-228b7812bf1d	ADMIN	ntdmi	WebAPI-WEB
595aaeab-0129-4d40-841b-7d3f6f2cbb7b	ADMIN	ntdmi	WebAPI-MOBILE

Hình 5.19: Session trước khi thực hiện.

- + Sau khi thực hiện đăng nhập ở trình duyệt, Session cũ de61ac1f... biến mất và bị thay thế session mới b39af018...

```

SELECT
    S.CLIENT_IDENTIFIER,
    S.USERNAME,
    S.OSUSER,
    S MODULE

FROM
    v$session S
WHERE
    S.USERNAME IS NOT NULL and S.CLIENT_IDENTIFIER is not null
ORDER BY
    S.LOGON_TIME;

```

Query Result

All Rows Fetched: 2 in 0.003 seconds

CLIENT_IDENTIFIER	USERNAME	OSUSER	MODULE
595aaeab-0129-4d40-841b-7d3f6f2cbb7b	ADMIN	ntdmi	WebAPI-MOBILE
b39af018-75b5-40b8-9394-5cbf00fb7df7	ADMIN	ntdmi	WebAPI-WEB

Hình 5.20: Kết quả thực tế Session cũ đã bị thay thế.

5.1.5. Chính sách profile khóa tài khoản

kịch bản: Đăng nhập sai 2 lần với tài khoản TIEPTAN và bị khóa

- Thao tác:

- + Kiểm tra thông tin người dùng TIEPTAN trong database để xác nhận trạng thái trước khi test (OPEN, chưa bị khóa, còn hạn).
- + Truy cập trang login dành cho nhân viên đăng nhập vào tài khoản nhân viên với vai trò Tiếp tân bằng mật khẩu sai lần 1.

The screenshot shows a MySQL Workbench interface. On the left, a code editor displays a SQL query:

```
SELECT
    USERNAME,
    ACCOUNT_STATUS,
    LOCK_DATE,
    EXPIRY_DATE,
    CREATED
FROM
    DBA_USERS
WHERE
    USERNAME = 'TIEPTAN';
```

On the right, a "Query Result" window shows the execution results:

	USERNAME	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	CREATED
1	TIEPTAN	OPEN	(null)	13-MAY-26	05-NOV-25

All rows were fetched in 0.019 seconds.

Hình 5.21: Kiểm tra trạng thái người dùng TIEPTAN.

Đăng nhập Admin bằng mật khẩu

Username

Tieptan

Password

Tieptan



Đăng nhập

Hình 5.22: Đăng nhập người dùng TIEPTAN với mật khẩu sai.

Sai username hoặc mật khẩu.

Hình 5.23: Thông báo sau khi nhập mật khẩu sai.

- + Thủ đăng nhập lại bằng mật khẩu sai lần 2.

- Kết quả mong đợi:
- Hiển thị thông báo: "Sai username hoặc mật khẩu."
- Tài khoản ngay lập tức bị khóa theo chính sách Profile.
- Các lần đăng nhập sau hiển thị thông báo: "Tài khoản bị khóa."
- Kết quả thực tế:

The screenshot shows a SQL query in the SQL editor and its execution results in the Result tab.

```

SELECT
    USERNAME,
    ACCOUNT_STATUS,
    LOCK_DATE,
    EXPIRY_DATE,
    CREATED
FROM
    DBA_USERS
WHERE
    USERNAME = 'TIEPTAN';
  
```

Result

All Rows Fetched: 1 in 0.005 seconds

USERNAME	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	CREATED
TIEPTAN	LOCKED	03-DEC-25	(null)	05-NOV-25

Hình 5.24: Trạng thái của người dùng TIEPTAN sau khi nhập sai mật khẩu.

5.1.6. Đăng xuất

- Thao tác:
- Đăng nhập vào tài khoản bất kỳ và ghi nhận Id phiên làm việc hiện tại
- Thực hiện nhấn nút đăng xuất, sẽ gọi procedure để xóa phiên làm việc dựa theo p_client_identifier

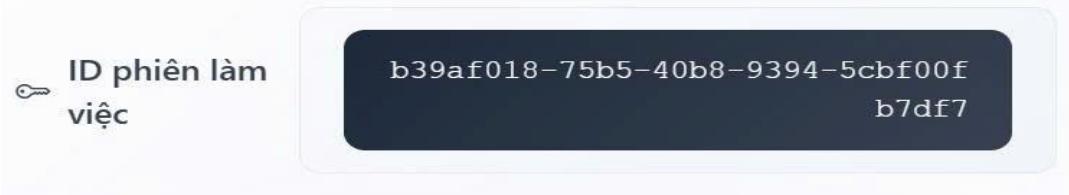
The screenshot shows a SQL query in the SQL editor.

```

SELECT
    S.CLIENT_IDENTIFIER,
    S.USERNAME,
    S.OSUSER,
    S MODULE
FROM
    v$session S
WHERE
    S.USERNAME IS NOT NULL and S.CLIENT_IDENTIFIER is not null
ORDER BY
    S.LOGON_TIME;
  
```

Hình 5.25: SQL kiểm tra phiên làm việc dựa theo p_client_identifier.

Thông tin phiên làm việc



Hình 5.26: Phiên làm việc hiện tại của người dùng trên nền tảng web.

- Kết quả mong đợi: Phiên làm việc sẽ xóa trong database
- Kết quả:

Query Result			
SQL	All Rows Fetched: 1 in 0.014 seconds		
CLIENT_IDENTIFIER	USERNAME	OSUSER	MODULE
1 595aaeab-0129-4d40-841b-7d3f6f2cbb7b	ADMIN	ntdmi	WebAPI-MOBILE

Hình 5.27: Phiên làm việc trên web đã bị xóa

5.1.7. Đăng ký

5.1.7.1 Kịch bản 1: Đăng ký khách hàng

- Thao tác:
 - + Kiểm tra người dùng có trên cơ sở dữ liệu hay không.
 - + Thực hiện đăng ký người dùng mới ở trang đăng ký khách hàng với số điện thoại ‘0332880205’.

SELECT USERNAME, ACCOUNT_STATUS, LOCK_DATE, EXPIRY_DATE, CREATED FROM DBA_USERS WHERE USERNAME = '0332880205';				
Query Result				
SQL	All Rows Fetched: 0 in 0.338 seconds			
USERNAME	ACCOUNT...	LOCK_DATE	EXPIRY_D...	CREATED

Hình 5.28: SQL kiểm tra người dùng có trên cơ sở dữ liệu hay không.

Register Customer

Full Name: Nguyễn Trần Dinh

Password:

Email: vinhdtq1@gmail.com

Phone: 0332880205

Address: trung đại học Công Thương

Register

Hình 5.29: Thực hiện đăng ký khách hàng.

- Kết quả mong đợi: Khách hàng xuất hiện trong database
- Kết quả thực tế:

```

SELECT
    USERNAME,
    ACCOUNT_STATUS,
    LOCK_DATE,
    EXPIRY_DATE,
    CREATED
FROM
    DBA_USERS
WHERE
    USERNAME = '0332880205';

```

Try Result | SQL | All Rows Fetched: 1 in 0.022 seconds

USERNAME	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	CREATED
0332880205	OPEN	(null)	01-JUN-26	03-DEC-25

Hình 5.30: Dữ liệu khách hàng cập nhập lên cơ sở dữ liệu.

5.1.7.2 Kịch bản 2: Đăng ký nhân viên

- Thao tác
 - + Kiểm tra cơ sở dữ liệu có tồn tại nhân viên đó không
 - + Nhân viên phải có quyền quản trị viên để thực hiện đăng ký nhân viên mới
 - + Thực hiện đăng ký nhân viên mới với username ‘Test’

```
SELECT
    USERNAME,
    ACCOUNT_STATUS,
    LOCK_DATE,
    EXPIRY_DATE,
    CREATED
FROM
    DBA_USERS
WHERE
    USERNAME = 'TEST';
```

The screenshot shows an Oracle SQL Developer interface. At the top, there is a code editor window containing the SQL query. Below it is a results window titled "Query Result" with the message "All Rows Fetched: 0 in 0.027 seconds". The results grid has columns labeled "USERNAME", "ACCOUNT...", "LOCK_DATE", "EXPIRY_D...", and "CREATED".

Hình 5.31: Kiểm tra sự tồn tại của username Test.

The screenshot shows a web-based registration form. It consists of several input fields with labels:

- FullName: Nguyễn Trần Dinh
- Username: Test
- Password: (redacted)
- Email: vinhdt12312313323@gmail.com
- Phone: 0332880119

A large blue button at the bottom right is labeled "Register".

Hình 5.32: Đăng ký nhân viên với username Test.

- Kết quả mong đợi: Tài khoản nhân viên được tạo với vai trò trống và 1 PrivateKey tương ứng cho người dùng nhân viên.
- Kết quả thực tế:



Hình 5.33: Private key tương ứng cho nhân viên

```

SELECT
    USERNAME,
    ACCOUNT_STATUS,
    LOCK_DATE,
    EXPIRY_DATE,
    CREATED
FROM
    DBA_USERS
WHERE
    USERNAME = 'TEST';

```

USERNAME	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	CREATED
TEST	OPEN	(null)	01-JUN-26	03-DEC-25

Hình 5.34: Dữ liệu nhân viên cập nhật trên cơ sở dữ liệu.

5.1.8. Đổi mật khẩu

Kịch bản: Đổi mật khẩu nhân viên có username ‘Test’ với mật khẩu Susu123@

- Thao tác:
 - + Kiểm tra database xem mã hash lưu sẵn trên database
 - + Ở trang đổi mật khẩu, điền form đổi mật khẩu cho nhân viên ‘Test’

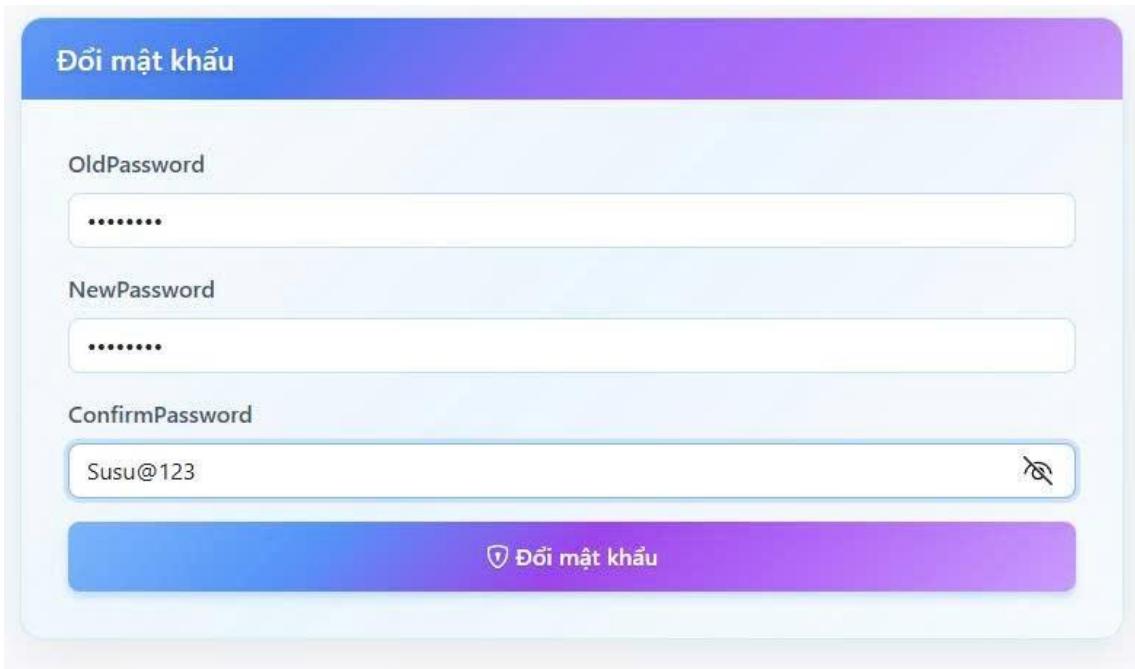
```

EXEC APP_CTX_PKG.set_role('ROLE_ADMIN');
select username,password_hash from employee where username='Test';
// với mật khẩu là Susu123@

```

USERNAME	PASSWORD_HASH
Test	81B6D7839B1B7F999CC846F8A37771B230D82F1C1C391CB0F490F4DE55704A1E

Hình 5.35: Mã băm mật khẩu lưu trên cơ sở dữ liệu.



Hình 5.36: *Đổi mật khẩu của người dùng.*

- Kết quả mong đợi:
- Kết quả thực tế:

USERNAME	PASSWORD_HASH
Test	D74BCF9DFB611C10D3B6292DD36EBA5E1225F095B5878456F72CA68AC3D2DFDA

Hình 5.37: *Mã hash ban đầu đã bị thay đổi.*

5.1.9. Nhập kho

Kịch bản nhập kho dùng chữ ký số doanh nghiệp .pfx

- Thao tác
 - + Thêm thông tin linh kiện nhập kho và xuất hóa đơn nhập kho.
 - + Kiểm tra trên cơ sở dữ liệu thông tin của linh kiện vừa nhập kho.

The screenshot shows a software application interface with three main sections:

- Thông tin Import** (Import Information):
 - Ghi chú: Test nhập kho có kí số với chữ kí số của doanh nghiệp với mk là Susu123@
 - PFX Certificate (.pfx):
 - Choose File: personal.pfx
 - Chọn file PFX certificate từ CA
 - Mật khẩu PFX:
 -
 - Mật khẩu bảo vệ file PFX
- Tạo nhiều item tự động** (Create many items automatically):
 - Part Name: VD: Cảm biến độ ẩm
 - Manufacturer: VD: Arduino
 - Serial Prefix: VD: A-
 - Số lượng: 1
 - Giá mỗi cái:
- Danh sách Items** (Items list):

Part Name	Manufacturer	Serial	Price	Xóa
Máy lùng điện thoại Iphone 17 Cam	Apple	MLIP17C	200000	Xóa

Hình 5.38: Thực hiện nhập kho vật phẩm mẫu

```
I reference
private byte[] GenerateAndSignPdf(
    Func<byte[]> pdfFactory,
    byte[] certificatePfxBytes,
    string certificatePassword,
    string updateProcedureName,
    Action<OracleCommand> configureUpdateProcedure,
    string invoiceType,
    int itemCount,
    int serviceCount = 0)
{
    _pdfSigner.ValidateCertificate(certificatePfxBytes, certificatePassword);

    var pdfBytes = pdfFactory();
    var (left, top) = CalculateSignaturePosition(invoiceType, itemCount, serviceCount);

    var signedPdfBytes = _pdfSigner.SignPdfWithDigitalCertificate(
        pdfBytes,
        certificatePfxBytes,
        certificatePassword,
        options =>
    {
        options.Left = left;
        options.Top = top;
        options.Margin = new Padding(0);
    });

    _pdfSigner.UpdateFinalPDF(
        procedureName: updateProcedureName,
        PDF: signedPdfBytes,
        configureParameters: configureUpdateProcedure);

    return signedPdfBytes;
}
```

Hình 5.39: Mã nguồn tạo và ký số PDF.

- Kết quả mong đợi:
 - + File pdf đã kí số lưu ở cơ sở dữ liệu.
 - + Lưu chữ kí số đảm bảo toàn vẹn dữ liệu ban đầu bên cơ sở dữ liệu.

- Kết quả thực tế:

```
E/EC APP_CTX_PKG.set_role('ROLE_ADMIN');
select * from stock_in where STOCKIN_ID = 92;
```

t Output x | Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x

SQL | All Rows Fetched: 1 in 0.019 seconds

STOCKIN_ID	EMP_ID	IN_DATE	NOTE
92	1	03-DEC-25	Test nhập kho có kí số với chữ kí số của doanh nghiệp với mk là Susu123@ (BLOB) 1-EIKj70Wa8THShpxgXhexZegUbwtbZR/rMD9qlvLqI

Hình 5.40: Thông tin chi tiết của đơn nhập kho trên cơ sở dữ liệu.

Created with evaluation version of GroupDocs.Signature © Aspose Pty Ltd 2001-2025. All Rights Reserved.

Công ty TNHH HealthCare
MST: 0123456789
123 Đường ABC, Phường XYZ, Quận 1, TP.HCM
ĐT: 0123456789
Email: info@mobileservice.com

Ký bởi 'E=vinhdtq123123123@gmail.com'
Ngày: 2025.12.03 08:17:08
Lý do: Approved
Địa điểm: Việt Nam

HÓA ĐƠN NHẬP KHO

Mã phiếu: #93
Ngày: 03/12/2025 15:17
Nhân viên: Admin
Ghi chú: Test nhập kho có kí số với chữ kí số của doanh nghiệp với mk là Susu123@

Tên linh kiện	Hãng	Serial	Giá
Mặt lưng điện thoại Iphone 17 Cam	Apple	MLIP17C	200.000 đ
Tổng cộng			200.000 đ

Chữ ký số:
(Đã ký số)
Công ty TNHH HealthCare

Hình 5.41: PDF đã được ký số.

5.1.10. Xuất hóa đơn, xuất kho

- Thao tác:

- + Đăng nhập tài khoản nhân viên và truy cập vào đơn hàng đang sửa chữa, ví dụ đơn hàng có Order ID: 343
- + Tạo yêu cầu linh kiện cho Order ID 343
- + Sau khi hệ thống khởi tạo yêu cầu linh kiện, duyệt yêu cầu linh kiện
- + Xác nhận hoàn thành đơn hàng và tải lên PFX để ký số, chọn xác nhận và xuất đơn.

Thông tin đơn hàng

Customer Phone	Handler Username
0332880205	Kithuatvien
Order Type	
REPAIR	
Description	Đơn hàng sửa mặt lưng iphone 17 Cam

Dịch vụ

Dịch vụ	Đơn giá	Số lượng	Thành tiền
Dịch vụ Sửa chữa tổng quát	350.000	1	350.000

Thêm dòng

Hình 5.42: Tạo đơn hàng mẫu

Tạo yêu cầu linh kiện

Thông tin yêu cầu

Order ID: 343	Nhân viên: Admin	Ngày yêu cầu: 2025-12-03 15:27	Trạng thái: Đang chờ
------------------	---------------------	-----------------------------------	-------------------------

Danh sách linh kiện

Linh kiện	
Mặt lưng điện thoại iPhone 17 Cam (MUP17C)	Xóa

Chưa có linh kiện nào. Vui lòng thêm linh kiện.

Hình 5.43: Tạo yêu cầu linh kiện mẫu

Request Id	EMPLOYEE_USERNAME	In Date	ORDER_ID	STATUS	ACTION
60	Admin	2025-12-03 15:27	343	Đang chờ	<input type="button" value="Xem chi tiết"/> <input type="button" value="Duyệt yêu cầu"/> <input type="button" value="Tùy chỉnh"/>

Hình 5.44: Hệ thống hiển thị yêu cầu.

Chi tiết Export #43

Employee Name: Admin	Out Date: 12/3/2025 3:29:26 PM	Note: Xuất kho tự động cho Order ID 343
-------------------------	-----------------------------------	--

Items

Part Name	Manufacturer	Serial	Price
Mặt lưng điện thoại iPhone 17 Cam	Apple	MUP17C	20000

Hình 5.45: Thông tin chi tiết cho đơn xuất kho trên hệ thống

- Kết quả mong đợi: Hệ thống trả về hóa đơn bán hàng và hóa đơn xuất kho.
- Kết quả thực tế:



Công ty TNHH HealthCare

MST: 0123456789
123 Đường ABC, Phường XYZ, Quận 1, TP.HCM
ĐT: 0123456789
Email: info@mobileservice.com

Ký bởi 'E=vinhdtq123123123@gmail.com'

Ngày: 2025.12.03 08:29:26

Lý do: Approved

Địa điểm: Việt Nam

HÓA ĐƠN XUẤT KHO

Mã phiếu: #43

Ngày: 03/12/2025 15:29

Nhân viên: Admin

Ghi chú: Xuất kho tự động cho Order ID 343

Tên linh kiện	Hãng	Serial	Giá
Mặt lưng điện thoại Iphone 17 Cam	Apple	MLIP17C	200.000 đ
Tổng cộng			200.000 đ

Chữ ký số:

Công ty TNHH HealthCare

Hình 5.46: PDF hóa đơn xuất kho.



Công ty TNHH HealthCare

MST: 0123456789
123 Đường ABC, Phường XYZ, Quận 1, TP.HCM
ĐT: 0123456789
Email: info@mobileservice.com

Ký bởi 'E=vinhdtq123123123@gmail.com'

Ngày: 2025.12.03 08:29:26

Lý do: Approved

Địa điểm: Việt Nam

HÓA ĐƠN BÁN HÀNG

Mã hóa đơn: #21

Ngày: 03/12/2025 15:29

Khách hàng: 0332880205

Nhân viên: Admin

Linh kiện:

Tên linh kiện	Hãng	Serial	Giá
Mặt lưng điện thoại Iphone 17 Cam	Apple	MLIP17C	200.000 đ
Tổng linh kiện:			200.000 đ

Dịch vụ:

Tên dịch vụ	SL	Đơn giá	Thành tiền
Dịch vụ Sửa chữa tổng quát	1	350.000 đ	350.000 đ
Tổng dịch vụ:			350.000 đ

TỔNG CỘNG: 550.000 đ

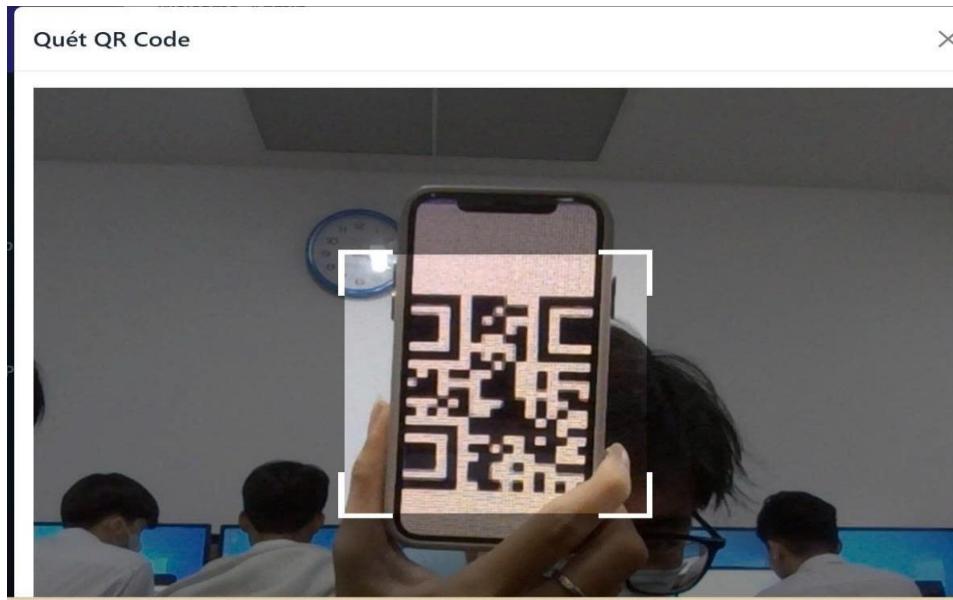
Chữ ký số:
(Đã ký số)

Công ty TNHH HealthCare

Hình 5.47: PDF hóa đơn bán hàng.

5.1.11. Quét linh kiện bằng mã QR

- Thao tác
 - + Mở ứng dụng mobile với tài khoản nhân viên
 - + Từ màn hình Dashboard, nhấn nút quét QR code
 - + Cấp quyền truy cập camera nếu được yêu cầu
 - + Đưa camera quét QR code của linh kiện



Hình 5.48: Mở Camera quét mã QR của linh kiện

- Kết quả mong đợi: Hệ thống hiển thị chi tiết linh kiện
- Kết quả thực tế:

Chi tiết linh kiện #302

Tên linh kiện Mặt kính lưng IP XS	Hãng sản xuất Apple	Serial MKLXS-1
Trạng thái Trong kho	Stockin ID 36	Order ID -
Giá 10,000 đ	QR Code	

Hình 5.49: Thông tin linh kiện được hiển thị sau khi quét.

5.1.12. Phân quyền và xác thực theo VPD

Kịch bản 1: Truy cập với quyền Quản trị viên (Admin)

- Thao tác: Đăng nhập người dùng Admin, thiết lập context vai trò là ADMIN.
- Kết quả mong đợi: Hệ thống hiển thị toàn bộ 20 dòng dữ liệu.
- Kết quả thực tế:

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. Below these, a code editor window contains the following SQL script:

```
-- 1. Thiết lập vai trò ADMIN trong Application Context
EXEC APP_CTX_PKG.SET_ROLE('ROLE_ADMIN');
-- 2. Truy vấn dữ liệu đơn hàng (ADMIN sẽ thấy toàn bộ 10 đơn)
SELECT * FROM ORDERS;
```

Below the code editor is a toolbar with icons for Script Output, Query Result 3, and Query Result 4. The 'Script Output' tab is selected. The status bar at the bottom indicates 'All Rows Fetched: 20 in 0.022 seconds'. The main area displays a grid of data from the ORDERS table:

ORDER_ID	CUSTOMER_PHONE	RECEIVER_EMP	HANDLER_EMP	ORDER_TYPE	RECEIVED_DATE	STATUS	DESCRIPTION
1	121 0332880207		1	3 WARRANTY	11-NOV-25	Đã hoàn thành TEST	
2	161 0332880207		1	1 WARRANTY	17-NOV-25	Đã hoàn thành ABV	
3	201 1		1	21 WARRANTY	20-NOV-25	Đã hoàn thành AAA	
4	242 0909000123		1	1 WARRANTY	27-NOV-25	Đã tiếp nhận bao hanh linh kien	
5	243 0909000123		1	3 REPAIR	27-NOV-25	Đã tiếp nhận Sua chua linh kien	
6	122 0332880207		1	3 REPAIR	11-NOV-25	Đã hoàn thành Admin	
7	123 0332880207		1	1 WARRANTY	11-NOV-25	Đã hoàn thành Admin1	
8	124 1		1	3 REPAIR	11-NOV-25	Đã hoàn thành Admin2	
9	125 1		1	3 WARRANTY	11-NOV-25	Đã hoàn thành Admin3	
10	181 1		1	1 WARRANTY	18-NOV-25	Đã hoàn thành A	
11	202 0332880207		4	3 WARRANTY	20-NOV-25	Đã hoàn thành ABC	
12	222 0332880207		1	21 WARRANTY	24-NOV-25	Đã hủy ABC	
13	141 1		1	3 WARRANTY	14-NOV-25	Đã hoàn thành Bảo hành	
14	142 0332880207		1	1 REPAIR	14-NOV-25	Đã hoàn thành ABC	
15	143 0332880207		1	3 WARRANTY	14-NOV-25	Đã hoàn thành Xuất hóa đơn	
16	144 0332880207		1	3 WARRANTY	14-NOV-25	Đã hoàn thành ABC	
17	83 1		1	3 REPAIR	10-NOV-25	Đã hoàn thành ABC	
18	84 0332880207		1	3 REPAIR	10-NOV-25	Đã hoàn thành Test dịch vụ	
19	85 0332880207		1	3 WARRANTY	10-NOV-25	Đã hoàn thành Bảo hành điện thoại	
20	101 0332880207		1	3 WARRANTY	10-NOV-25	Đã hoàn thành BẢO HÀNH ĐTH	

Hình 5.50: Kết quả truy vấn bảng ORDERS dưới quyền Admin.

Kịch bản 2: Truy cập với quyền Khách hàng

- Thao tác: Giả lập đăng nhập khách hàng Nguyen Van A (SĐT: 0909000123), thiết lập context vai trò là KHACHHANG.
- Kết quả mong đợi: Hệ thống tự động thêm điều kiện lọc WHERE CUSTOMER_PHONE = '0909000123', chỉ hiển thị 2 dòng dữ liệu.
- Kết quả thực tế:

```
-- . Thiết lập role + phone vào Application Context
EXEC APP_CTX_PKG.SET_ROLE('ROLE_KHACHHANG');
EXEC APP_CTX_PKG.SET_CUSTOMER('0909000123');

SELECT * FROM ORDERS;
```

Script Output | Query Result 3 | SQL | All Rows Fetched: 2 in 0.003 seconds

ORDER_ID	CUSTOMER_PHONE	RECEIVER_EMP	HANDLER_EMP	ORDER_TYPE	RECEIVED_DATE	STATUS	DESCRIPTION
1	242 0909000123	1	1	WARRANTY	27-NOV-25	Đã tiếp nhận bao hanh linh kien	
2	243 0909000123	1	3	REPAIR	27-NOV-25	Đã tiếp nhận Sua chua linh kien	

Hình 5.51: Kết quả truy vấn bảng ORDERS dưới quyền Khách hàng.

5.1.13. Kịch bản kiểm thử Audit

Để đánh giá hiệu quả của hệ thống giám sát, nhóm đã tiến hành kiểm thử hai cấp độ bảo mật: giám sát hành vi người dùng (through Standard Audit) và giám sát tính toàn vẹn dữ liệu (through Trigger).

5.1.13.1 Kiểm thử Standard Audit

Hệ thống được cấu hình để ghi nhật ký các hành động nhạy cảm như xóa dữ liệu khách hàng.

- Thao tác:** Tài khoản nhân viên thực hiện lệnh xóa một bản ghi trong bảng CUSTOMER.

```

-- Thiết lập context vai trò ADMIN
EXEC APP.APP_CTX_PKG.SET_ROLE('ROLE_ADMIN');
EXEC APP.APP_CTX_PKG.SET_EMP('1');

SELECT
    SYS_CONTEXT('APP_CTX','ROLE_NAME') AS vai_tro,
    SYS_CONTEXT('APP_CTX','EMP_ID') AS ma_nhanvien,
    USER AS current_user
FROM dual;

-- Lưu timestamp để dễ tra cứu audit sau
SELECT TO_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH24:MI:SS') AS thoi_gian_bat_dau
FROM dual;

-- Kiểm tra đơn hàng sẽ xóa
SELECT ORDER_ID, CUSTOMER_PHONE, STATUS
FROM APP.ORDERS
WHERE ORDER_ID = 222;
-- Xóa đơn hàng
DELETE FROM APP.ORDERS
WHERE ORDER_ID = 222;
SELECT TO_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH24:MI:SS') AS thoi_gian_ket_thuc
FROM dual;
COMMIT;

```

Query Result | Script Output | Query Result 1 | Query Result 2

All Rows Fetched: 1 in 0.003 seconds

ORDER_ID	CUSTOMER_PHONE	STATUS
1	222 0332880207	Đã hủy

Hình 5.52: Thiết lập thao tác kiểm thử Standard Audit.

- Kết quả:** Hệ thống ghi nhận chính xác trong DBA_AUDIT_TRAIL thông tin về người thực hiện, thời gian.

```

-- Xem tất cả DELETE trên bảng ORDERS trong 1 giờ qua
SELECT
    TO_CHAR(timestamp, 'DD-MON-YY HH24:MI:SS') AS thoi_gian,
    username AS nguoi_thuc_hien,
    os_username AS user_he_dieu_hanh,
    terminal AS may_tinh,
    owner || '.' || obj_name AS bang,
    action_name AS hanh_dong,
    returncode AS ma_ketqua,
    CASE returncode
        WHEN 0 THEN 'Thành công'
        ELSE 'Thất bại (ORA-' || returncode || ')'
    END AS trang_thai,
    SUBSTR(sql_text, 1, 200) AS cau_lenh_sql
FROM DBA_AUDIT_TRAIL
WHERE obj_name = 'ORDERS'
    AND action_name = 'DELETE'
    AND timestamp > SYSTIMESTAMP - INTERVAL '1' HOUR
ORDER BY timestamp DESC;

```

Query Result | Script Output | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5

All Rows Fetched: 14 in 0.008 seconds

THOI_GIAN	NGUOI_THUC_HIEN	USER_HE_DIEU_HANH	MAY_TINH	BANG	HANH_DONG	MA_KET_QUA	TRANG_THAI	CAU_LENH_SQL
1 28-NOV-25 23:15:27 APP	ASUS	unknown	APP.ORDERS	DELETE		0	Thành công	(null)

Hình 5.53: Kết quả truy vấn DBA_AUDIT_TRAIL

5.1.13.2 Kiểm thử Trigger Audit (Lưu vết thay đổi dữ liệu)

Đối với các dữ liệu quan trọng như giá linh kiện, yêu cầu đặt ra là phải biết được giá trị cũ thể trước và sau khi thay đổi để phục vụ giải trình.

- Thao tác: Cập nhật đơn giá linh kiện mã 101 từ 500.000 VNĐ lên 1.000.000 VNĐ.

```
-- thiết lập context để log rõ vai trò
EXEC APP.APP_CTX_PKG.SET_ROLE('ROLE_ADMIN');
EXEC APP.APP_CTX_PKG.SET_EMP('1');
-- Kiểm tra giá hiện tại
SELECT PART_ID, NAME, PRICE
FROM APP.PART
WHERE PART_ID = 312;
-- Cập nhật giá: 100000 -> 99999
UPDATE APP.PART
SET PRICE = 99999
WHERE PART_ID = 312;
COMMIT;
--Ghi nhận thời điểm để đối chiếu
SELECT TO_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH24:MI:SS') AS TS_AFTER_UPDATE FROM dual;
```

Query Result

PART_ID	NAME	PRICE
1	312 TEST	100000

Hình 5.54: Thiết lập thao tác kiểm thử Trigger

- Kết quả: Trigger tự động kích hoạt và ghi vào bảng AUDIT_ALERT_LOG. Log hiển thị rõ cặp giá trị cũ/mới (Old/New Values), đảm bảo tính minh bạch tuyệt đối.

```
-- Xem log mới nhất cho bảng PART
SELECT
    event_ts,
    TO_CHAR(event_ts, 'YYYY-MM-DD HH24:MI:SS') AS event_time,
    changed_columns,
    dml_type,
    old_values,
    new_values
FROM APP.audit_alert_log
WHERE object_name = 'PART'
AND INSTR(changed_columns, 'PRICE') > 0
ORDER BY event_ts DESC
FETCH FIRST 1 ROWS ONLY;
```

Output

EVENT_TS	EVENT_TIME	CHANGED_COLUMNS	DML_TYPE	OLD_VALUES	NEW_VALUES
28-NOV-25 11.36.14.993525000 PM	2025-11-28 23:36:14	PRICE	UPDATE	{"PART_ID":312,"NAME":"TEST",...}	{"PART_ID":312,"NAME":"TEST","MANUFACTURER":"TEST","SER...

Hình 5.55: Kết quả truy vấn log

Giá trị cũ và mới được hiển thị rõ sau khi thay đổi .

OLD_VALUES
{"PART_ID":312,"NAME":"TEST","MANUFACTURER":"TEST","SERIAL":"TEST-1","STATUS":"Đã xuất kho","STOCK_IN_ID":37,"ORDER_ID":84,"PRICE":100000}

Hình 5.56: Giá trị cũ của dữ liệu

NEW_VALUES
{"PART_ID":312,"NAME":"TEST","MANUFACTURER":"TEST","SERIAL":"TEST-1","STATUS":"Đã xuất kho","STOCK_IN_ID":37,"ORDER_ID":84,"PRICE":99999}

Hình 5.57: Giá trị mới sau khi chỉnh sửa

5.1.14. Phục hồi dữ liệu theo thời điểm

Mục tiêu: Kiểm chứng khả năng khôi phục cơ sở dữ liệu về trạng thái ổn định trong quá khứ sau khi xảy ra sự cố (ví dụ: thao tác sai hoặc lỗi dữ liệu).

- Bước 1: Ghi nhận trạng thái dữ liệu gốc (Trước khi Backup) Kiểm tra dữ liệu hiện tại trong bảng A. Tại thời điểm này, bảng có 2 bản ghi (ID 1 và 2).

ID	NAME	CREATED_AT
1	Tên gì đó	17-NOV-25
2	Nguyen Van A	01-DEC-25

Hình 5.58: Trạng thái dữ liệu trước khi sao lưu.

- Bước 2: Thực hiện Sao lưu (Backup) Kích hoạt script sao lưu tự động run_rman_backup.sh. Hệ thống thực hiện sao lưu toàn bộ Database và Archive Logs.

Trạng thái: Backup thành công (Completed successfully).

File log: /home/oracle/backup_web/logs/rman_backup_...

```
[oracle@localhost ~]$ /home/oracle/backup_web/scripts/run_rman_backup.sh
RMAN backup of ORCLPDB1 completed successfully. Log: /home/oracle/backup_web/logs/rman_backup_2025-12-01_221913.log
```

Hình 5.59: Nhật ký thực thi quá trình sao lưu thành công.

- Bước 3: Giả lập thay đổi dữ liệu (Sự cố/Biến động) Sau khi backup xong, thực hiện thêm mới dữ liệu (Insert ID 3) vào hệ thống để mô phỏng các giao dịch phát sinh.

```

INSERT INTO A (ID, NAME, CREATED_AT)
VALUES (3,'Nguyen Van B', TO_DATE('2025-12-01 10:00:00', 'YYYY-MM-DD HH24:MI:SS'));
select * from A;
commit;

```

Hình 5.60: Thực hiện thay đổi dữ liệu sau thời điểm sao lưu.

ID	NAME	CREATED_AT
3	Nguyen Van B	01-DEC-25
1	Tên gì đó	17-NOV-25
2	Nguyen Van A	01-DEC-25

Hình 5.61: Dữ liệu sau khi thay đổi (Có thêm dòng ID 3).

- Bước 4: Thực hiện Phục hồi (Restore) Giả định hệ thống gấp sự cố hoặc cần quay về trạng thái cũ, quản trị viên chạy script run_rman_restore.sh để khôi phục Database về thời điểm đã chọn (Point-in-Time).
 - + Cơ chế: Script tự động đóng PDB (Closing PDB), chạy RMAN Restore và mở lại kết nối.
 - + Kết quả: Quá trình Restore báo trạng thái SUCCESS.

```

[oracle@localhost ~]$ /home/oracle/backup_web/scripts/run_rman_restore.sh "2025-12-01 22:19:13"
==> Closing PDB ORCLPDB1...

Pluggable database altered.

==> Running RMAN PITR restore...
V RESTORE SUCCESS (PITR) - log: /home/oracle/backup_web/logs/rman_restore_2025-12-01_222332.log
[oracle@localhost ~]$ 

```

Hình 5.62: Nhật ký thực thi quá trình phục hồi dữ liệu.

ID	NAME	CREATED_AT
1	2 Nguyen Van A	01-DEC-25
2	1 Tên gì đó	17-NOV-25

Hình 5.63: Dữ liệu sau khi restore.

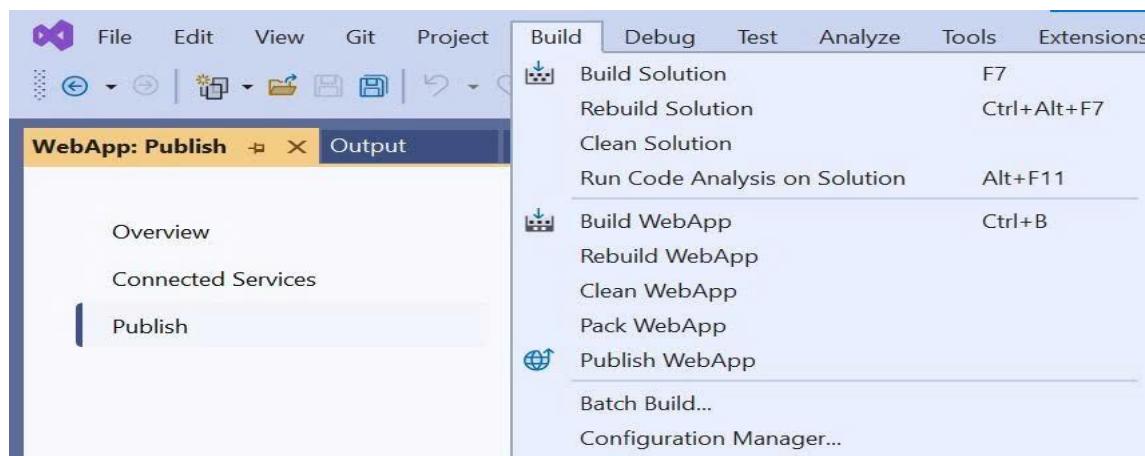
5.2. Đóng gói ứng dụng

Thực hiện đóng gói ứng dụng bằng công cụ Visual Studio 2022

<input checked="" type="checkbox"/> Mobile	12/3/2025 12:47 PM	File folder
WebAPI	12/3/2025 11:58 AM	File folder
WebAPP	12/3/2025 12:00 PM	File folder

Hình 5.64: Thư mục mã nguồn.

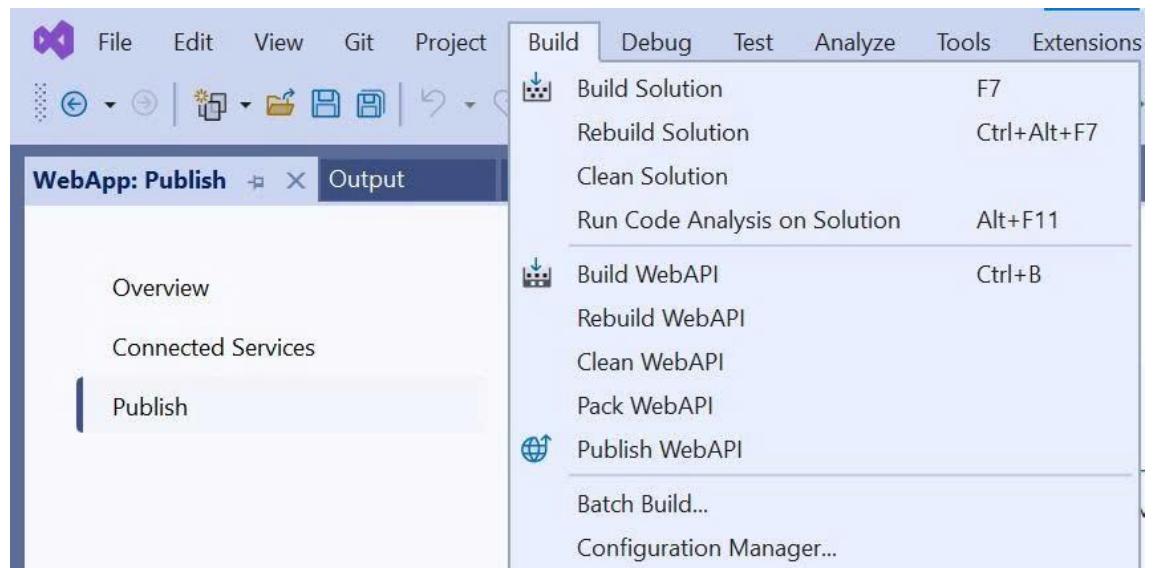
Dùng công cụ Visual Studio 2022 mở thư mục mã nguồn. Trên thanh công cụ tiện ích, chọn Publish lần lượt các thành phần như WebApp, WebAPI.



Hình 5.65: Đóng gói WebApp.

WebApp.deps.json	12/3/2025 12:00 PM	JSON Source File	86 KB
WebApp.dll	12/3/2025 12:00 PM	Application extension	1,643 KB
WebApp.exe	12/3/2025 12:00 PM	Application	136 KB
WebApp.ndb	12/3/2025 12:00 PM	Program Debug Data	376 KB

Hình 5.66: WebApp sau khi đóng gói.



Hình 5.67: Đóng gói WebAPI.

	WebAPI.deps.json	12/3/2025 11:58 AM
	WebAPI.dll	12/3/2025 11:58 AM
	WebAPI.exe	12/3/2025 11:58 AM
	WebAPI.pdb	12/3/2025 11:58 AM
	WebAPI.runtimeconfig.json	12/3/2025 11:56 AM

Hình 5.68: WebAPI sau khi đóng gói.

	app-release.apk	12/3/2025 12:47 PM	APK File	64,019 KB
	app-release.apk.sha1	12/3/2025 12:46 PM	SHA1 File	1 KB

Hình 5.69: Đóng gói apk

KẾT LUẬN

Đồ án đã hoàn thành các mục tiêu đề ra trong đề cương chi tiết:

- Về mặt nghiệp vụ: Đã khảo sát và mô hình hóa thành công quy trình nghiệp vụ thực tế tại cửa hàng, từ khâu tiếp nhận, sửa chữa, quản lý kho linh kiện đến bảo hành và thanh toán. Xây dựng được ứng dụng hoàn chỉnh trên nền tảng Web và Mobile đáp ứng nhu cầu quản lý.
- Về mặt hạ tầng: Triển khai thành công hệ thống trên môi trường Oracle Linux 8 và Hệ quản trị cơ sở dữ liệu Oracle Database 21c theo kiến trúc đa thuê bao (Multitenant Architecture - CDB/PDB). Giải quyết triệt để bài toán kết nối bảo mật từ xa thông qua giải pháp mạng riêng ảo ZeroTier VPN.
- Về mặt giải pháp bảo mật (Trọng tâm):
 - + Kiểm soát truy cập (DAC & RBAC): Hiện thực hóa cơ chế phân quyền dựa trên vai trò (Role-Based Access Control) kết hợp với mô hình DAC, đảm bảo nguyên tắc đặc quyền tối thiểu cho từng nhóm người dùng (Admin, Thủ kho, Kỹ thuật viên, Khách hàng).
 - + Bảo mật mức dòng (VPD): Áp dụng thành công kỹ thuật Virtual Private Database (VPD) để phân quyền dữ liệu đa cấp độ dựa trên ngữ cảnh người dùng. Đây là giải pháp thay thế linh hoạt cho mô hình MAC (Oracle Label Security), giúp bảo vệ dữ liệu nhạy cảm (như đơn hàng, lịch sử sửa chữa) của từng khách hàng một cách biệt lập.
 - + Mã hóa và Xác thực: Triển khai cơ chế mã hóa lai (Hybrid Encryption) kết hợp giữa RSA và AES để bảo vệ dữ liệu trên đường truyền. Tích hợp chữ ký số và xác thực toàn vẹn dữ liệu cho hóa đơn điện tử (PDF) và dữ liệu trong Database, đảm bảo tính chống chối bỏ.
 - + Kiểm toán (Auditing): Thiết lập hệ thống giám sát chặt chẽ thông qua Standard Audit và Trigger Audit, ghi nhận đầy đủ lịch sử truy cập và các biến động dữ liệu quan trọng (lưu vết giá trị cũ/mới).

Hạn chế:

Bên cạnh những kết quả đạt được, đồ án vẫn còn một số hạn chế:

- Về tính năng FGA (Fine-Grained Auditing): Do hạn chế về tài nguyên phần cứng của máy chủ thử nghiệm và vấn đề hiệu năng, nhóm chưa triển khai kiểm toán FGA mà tập trung tối ưu hóa Standard Audit kết hợp Trigger để đảm bảo yêu cầu giám sát.
- Giao diện ứng dụng trên Mobile cần được tối ưu thêm về trải nghiệm người dùng (UX) khi xử lý các tác vụ phức tạp.

Hướng phát triển

- Trong tương lai, nhóm đề xuất các hướng phát triển để hoàn thiện hệ thống:
- Nâng cấp hạ tầng để triển khai FGA và Oracle Label Security (OLS) nhằm phục vụ các mô hình bảo mật đa cấp phức tạp hơn.
- Tích hợp các giải pháp sao lưu dự phòng trên nền tảng đám mây (Cloud Backup) để tăng cường an toàn dữ liệu.
- Phát triển thêm các module phân tích dữ liệu (Data Analytics) để hỗ trợ cửa hàng ra quyết định kinh doanh dựa trên lịch sử sửa chữa và bảo hành.

TÀI LIỆU THAM KHẢO

- [1] Khoa CNTT - ĐH Công Thương TP.HCM, *Bài Giảng Phân tích hệ thống thông tin*.
- [2] Khoa CNTT - ĐH Công Thương TP.HCM, *Bài Giảng Công nghệ phần mềm*.
- [3] Phạm Hữu Khang, *Kỹ thuật lập trình ứng dụng chuyên nghiệp C#*, NXB Lao Động Xã Hội, 2006.
- [4] Lê Phúc, *Bài giảng Bảo mật Hệ thống thông tin*, Học viện Công nghệ Bưu chính Viễn thông, 2007.
- [5] Trần Văn Dũng, *Giáo trình An toàn và Bảo mật thông tin*, Trường Đại học Giao thông Vận tải, 2007.
- [6] Dương Anh Đức, Trần Minh Triết, *Mã hóa và ứng dụng*, NXB Đại học Quốc gia TP.HCM, 2005.
- [7] Phạm Nguyễn Cương, Nguyễn Trần Minh Thư, Hồ Bảo Quốc, *Giáo trình Phân tích thiết kế hệ thống thông tin theo hướng đối tượng*, NXB Khoa học và Kỹ thuật, 2016.
- [8] Nguyễn Thái Nghe, Trần Ngân Bình, Đặng Quốc Việt, *Giáo trình Hệ quản trị cơ sở dữ liệu*, NXB Đại học Cần Thơ, 2014.
- [9] Padmaja Potinei, *Oracle Database Backup and Recovery User's Guide, 18c*, Oracle, 2018.
- [10] Lance Ashdown, Tom Kyte, Joe McCormack, *Oracle Database Concepts, 18c*, Oracle, 2018.
- [11] Rajesh Bhatiya, Padmaja Potineni, *Database Administrator's Guide 18c*, Oracle, 2020.
- [12] https://github.com/huyDemo-us/db_management_oracle/tree/main, truy cập ngày 02/11/2025.
- [13] https://github.com/sanguyeenx96/NetCore6_spareparts_management_ver2, truy cập ngày 02/11/2025.
- [14] <https://www.chirags.in/oracle-database-21c-installation-on-oracle-linux-8-and-connect-with-sql-developer/>, truy cập ngày 02/11/2025.