

1. DB

1.1. HIS_TRANSACTION

- BANK_TRANSACTION_CODE VARCHAR2(12 BYTE) : Mã giao dịch của ngân hàng
 - Đặt unique-constraint
- BANK_TRANSACTION_TIME NUMBER (14,0): Thời gian giao dịch

1.2. HIS_TRANS_REQ

- BANK_MESSAGE NVARCHAR(2000): Thông tin trạng thái thanh toán

2. BACKEND

2.1. Xác nhận thanh toán Vietinbank

2.1.1. Cấu hình

Cấu hình hệ thống:

- *MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO*
 - Mô tả: Cấu hình thông phòng thu ngân, thu ngân, sổ hóa đơn sử dụng khi thanh toán.
<Mã phòng thu ngân>|<tên đăng nhập> - <họ tên>|<Mã sổ biên lai/Hóa đơn>
- *MOS.HIS_TRANSACTION.QR_PAYMENT.DEPOSIT_INFO*.
 - Mô tả: Cấu hình thông phòng thu ngân, thu ngân, sổ hóa đơn sử dụng khi tạm ứng.
<Mã phòng thu ngân>|<tên đăng nhập> - <họ tên>|<Mã sổ biên lai/Hóa đơn>
- *HIS.Desktop.Plugins.PaymentQrCode.VietinbankInfo*:
 - Các tham số phục vụ sinh mã QR do Vietinbank cung cấp

Cấu hình trong web.config MOS:

- <add key="MOS.MANAGER.Vietinbank.CertificatePath" value="" />: Cấu hình đường dẫn chính xác đến vị trí file chứng thư(*.cer) public do bên Vietinbank cung cấp. Ví dụ: C:\\backend\\MOS\\file\\NA_VIETINBANK.cer
- <add key="MOS.MANAGER.Vietinbank.HashAlg" value="SHA256" />: Cấu hình thuật toán mã hóa dữ liệu của khi tích hợp Vietinbank. Nhập giá trị SHA1 hoặc SHA256
- <add key="MOS.MANAGER.Inventec.CertificatePath" value="" />: Cấu hình đường dẫn chính xác đến vị trí file chứng thư(*.cer) private của bệnh viện. Ví dụ: C:\\backend\\MOS\\file\\Bkav_1122121.cer
- <add key="MOS.MANAGER.Inventec.CertificatePass" value="" />: Cấu hình mật khẩu tương ứng của chứng thư.

2.1.2. Api/HisTransReq/QrPaymentVietinbank

- **Input:**

- statusCode: Trạng thái giao dịch (00: Giao dịch thành công phía bank)
- amount : Tổng tiền thanh toán
- terminalId : terminalId được khai báo trong key cấu hình hệ thống
- bankTransactionId : TransactionId của ngân hàng
- requestId : Số hóa đơn của bank
- merchantName : merchantName được khai báo trong key cấu hình hệ thống
- merchantId : Mã merchantId được khai báo trong key cấu hình hệ thống
- transactionDate : Ngày giao dịch
- orderId : TRANS_REQ_CODE (HIS_TRANS_REQ) của HIS
- statusMessage : Thông tin về trạng thái thanh toán, số hóa đơn, số tk, tên tk thanh toán
- productId : productId của sản phẩm
- signature : Dữ liệu mã hóa = sign (requestId + merchantId + orderId + productId)

- **Output:**

- requestId
- paymentStatus
- signature: Dữ liệu mã hóa = sign (requestId + paymentStatus)

- **Validate:**

- Nếu statusCode khác “00” thì dừng không xử lý tiếp thì kết thúc xử lý, trả về output với paymentStatus = “02”

- Kiểm tra thông tin terminalId , merchantName , merchantId so với giá trị của key cấu hình hệ thống *HIS.Desktop.Plugins.PaymentQrCode.VietinbankInfo*:
 - Nếu các thông tin trên không đúng với thông tin cấu hình thì kết thúc xử lý, trả về output với paymentStatus = “02”
- Xử lý kiểm tra thông tin chứng thư.
 - Các key cấu hình trong web.config sẽ bắt buộc phải có dữ liệu trước khi xử lý dữ liệu trên hệ thống HIS
 - Bổ sung log cụ thể thông tin nào chưa được khai báo
 - Kiểm tra hạn chứng thư.
 - Sử dụng hàm GetExpirationDateString để lấy ra ngày hết hạn của chứng thư.
 - Trường hợp ngày hiện tại lớn hơn ngày hết hạn của chứng thư thì dừng xử lý
 - Bổ sung log cụ thể chứng thư nào hết hạn
- Xử lý kiểm tra dữ liệu ký (verify) từ thông tin ký số đi kèm dữ liệu do vietinbank trả về (signature)
 - Xử lý tạo ra chuỗi dữ liệu kiểm tra từ các thông tin do bank gửi theo đúng thứ tự gọi là A
 - requestId: Số hóa đơn
 - merchantId: Mã merchant
 - orderId: Mã yêu cầu thanh toán
 - productId: Mã sản phẩm.
 - Sử dụng file chứng thư của vietinbank và thuật toán mã hóa để so sánh A và signature được gửi kèm
 - Xử lý verify với:
 - Thuật toán sử dụng: lấy từ cấu hình MOS.MANAGER.Vietinbank.HashAlg
 - Public key lấy từ file cer được cấu hình theo MOS.MANAGER.Vietinbank.CertificatePath
 - Nếu xác thực hợp lệ thì tiếp tục xử lý
 - Nếu xác thực không hợp lệ thì trả về output với paymentStatus = “02”
- Xử lý ký dữ liệu (sign) khi gửi trả kết quả lại cho ngân hàng
 - Xử lý tạo ra chuỗi dữ liệu kiểm tra từ các thông tin do bank gửi theo đúng thứ tự gọi là A
 - requestId: Số hóa đơn
 - paymentStatus: Trạng thái xử lý
 - Xử lý ký (sign) với:
 - Thuật toán sử dụng: lấy từ cấu hình MOS.MANAGER.Vietinbank.HashAlg

- Private key lấy từ file cer được cấu hình theo MOS.MANAGER.Inventec.CertificatePath
 - Mật khẩu đọc file cer được cấu hình theo MOS.MANAGER.Inventec.CertificatePass
 - Dữ liệu tạo ra truyền vào trường signature để trả về cho Vietinbank
- Kiểm tra thông tin orderId:
 - Kiểm tra tồn tại thông tin yêu cầu thanh toán
 - Lấy ra bản ghi trong HIS_TRANS_REQ thỏa mãn điều kiện: TRANS_REQ_CODE = orderId, (gọi là A)
 - Nếu không tồn tại A, thì kết thúc xử lý, trả về output với paymentStatus = “03”
 - Kiểm tra trạng thái của yêu cầu thanh toán
 - Nếu TRANS_REQ_STT_ID (A) = 2 (trạng thái "Hoàn thành"), thì kết thúc xử lý, trả về output với paymentStatus = “01”
 - Nếu TRANS_REQ_STT_ID (A) = (3, 4) (trạng thái "Hủy", “Thất bại”), thì kết thúc xử lý, trả về output với paymentStatus = “02”
 - Kiểm tra số tiền của yêu cầu thanh toán
 - Nếu AMOUNT (A) ≠ amount thì:
 - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS_REQ_STT_ID (A) = 3)
 - Trả về output với paymentStatus = “04”
 - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
 - Kết thúc xử lý
 - Nếu TRANS_REQ_TYPE(A) = 1,2 thì:
 - Kiểm tra dịch vụ đã được thanh toán/tạm ứng hay chưa. Cụ thể:
 - B1: Lấy thông tin chi tiết yêu cầu thanh toán. Cụ thể lấy các bản ghi HIS_SESE_TRANS_REQ có TRANS_REQ_ID = ID (A) (gọi là danh sách B)
 - B2: Từ danh sách B, lấy ra danh sách SERE_SERV_ID (gọi là danh sách C)
 - Kiểm tra xem dịch vụ đã thanh toán chưa:
 - Thực hiện truy vấn lấy ra danh sách thanh toán đã thanh toán. Cụ thể, lấy ra HIS_SERE_SERV_BILL thỏa mãn đồng thời:
 - Có SERE_SERV_ID thuộc danh sách C
 - Có IS_CANCEL ≠ 1

- Nếu danh sách có dữ liệu, thì:
 - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS_REQ_STT_ID (A) = 3)
 - Trả về output với paymentStatus = “01”
 - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
 - Kết thúc xử lý
 - Kiểm tra xem dịch vụ đã tạm ứng chưa:
 - Thực hiện truy vấn lấy ra danh sách thanh toán đã thanh toán. Cụ thể, lấy ra HIS_SERE_SERV_DEPOSIT thỏa mãn đồng thời:
 - Có SERE_SERV_ID thuộc danh sách C
 - Có IS_CANCEL ≠ 1
 - Nếu danh sách có dữ liệu, thì:
 - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS_REQ_STT_ID (A) = 3)
 - Trả về output với paymentStatus = “01”
 - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
 - Kết thúc xử lý
- Kiểm tra xem thông tin chi tiết yêu cầu thanh toán có được có khớp với thông tin chi tiết dịch vụ hay không. Cụ thể:
 - B1: Lấy ra danh sách các dịch vụ tương ứng với yêu cầu thanh toán, gọi là danh sách D. Cụ thể, lấy ra các HIS_SERE_SERV có ID thuộc danh sách C
 - B2: Được coi là “khớp” nếu bản ghi HIS_SESE_TRANS_REQ trong danh sách B thì có bản ghi HIS_SERE_SERV tương ứng thuộc danh sách D và ngược lại.
 - Được coi là “tương ứng” nếu thỏa mãn đồng thời:
 - HIS_SESE_TRANS_REQ(SERE_SERV_ID) = HIS_SERE_SERV(ID)
 - HIS_SESE_TRANS_REQ(PRICE) = HIS_SERE_SERV(VIR_TOTAL_PATIENT_PRICE)

- Nếu không khớp thì:
 - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS_REQ_STT_ID (A) = 3)
 - Trả về output với paymentStatus = “04”
 - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
 - Kết thúc xử lý
- Nếu TRANS_REQ_TYPE(A)= 4 thì:
 - Kiểm tra yêu cầu tạm ứng đã được đóng tiền chưa:
 - B1: Lấy thông tin yêu cầu tạm ứng HIS_DEPOSIT_REQ có TRANS_REQ_ID = ID (A)
 - B2: Nếu B1 có dữ liệu thì kiểm tra nếu yêu cầu tạm ứng đã được thanh toán (DEPOSIT_ID(HIS_DEPOSIT_REQ) ≠ null) thì:
 - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS_REQ_STT_ID (A) = 3)
 - Trả về output với paymentStatus = “04”
 - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
 - Kết thúc xử lý
 - Kiểm tra số tiền của yêu cầu tạm ứng:
 - Nếu B1 có dữ liệu thì kiểm tra: Nếu AMOUNT (HIS_DEPOSIT_REQ) ≠ AMOUNT (A) thì:
 - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS_REQ_STT_ID (A) = 3)
 - Trả về output với paymentStatus = “04”
 - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
 - Kết thúc xử lý
- Nếu TRANS_REQ_TYPE(A) = 3:
 - Kiểm tra số tiền của yêu cầu thanh toán:
 - B1: Lấy tổng chi phí bệnh nhân còn thiếu (Gọi là X) = NVL(TOTAL_PATIENT_PRICE,0) - NVL(TOTAL_DEPOSIT_AMOUNT,0) - NVL(TOTAL_DEBT_AMOUNT,0) - NVL(TOTAL_BILL_AMOUNT,0) + NVL(TOTAL_BILL_TRANSFER_AMOUNT,0) + NVL(TOTAL_REPAY_AMOUNT,0) từ V_HIS_TREATMENT_FEE có ID = TREATMENT_ID (HIS_TRANS_REQ) của bản ghi A

- B2: Nếu X khác A thì:
 - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS_REQ_STT_ID (A) = 3)
 - Trả về output với paymentStatus = “04”
 - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
 - Kết thúc xử lý

Mã lỗi	Nội dung	Ghi chú
00	Thành công	
01	Giao dịch đã được thanh toán trước đó	Hoàn tiền
02	Giao dịch không hợp lệ	Hoàn tiền
03	Giao dịch không tìm thấy	Hoàn tiền
04	Số tiền không hợp lệ	Hoàn tiền
05	Giao dịch đã hết hạn thanh toán.	Hoàn tiền
08	Timeout Chưa xác định được	chờ tra soát
09	Bảo trì	Hoàn tiền

- **Xử lý:**
 - Lấy thông tin HIS_TREATMENT tương ứng với TREATMENT_ID (HIS_TRANS_REQ)
 - Tạo giao dịch HIS_TRANSACTION:
 - Nếu HIS_TRANS_REQ có TRANS_REQ_TYPE = 1 hoặc 2:

- ACCOUNT_BOOK_ID = ID sổ cấu hình trong key MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO
- AMOUNT = AMOUNT (HIS_TRANS_REQ)
- BILL_TYPE_ID = BILL_TYPE_ID (HIS_ACCOUNT_BOOK) của sổ thanh toán
- BUYER_ACCOUNT_NUMBER = số tài khoản
- BUYER_ADDRESS = TDL_PATIENT_ACCOUNT_NUMBER (HIS_TREATMENT)
- BUYER_NAME = TDL_PATIENT_NAME (HIS_TREATMENT)
- BUYER_ORGANIZATION = Nếu TDL_PATIENT_WORK_PLACE (HIS_TREATMENT) có thông tin thì lấy TDL_PATIENT_WORK_PLACE. Ngược lại thì lấy thông tin TDL_PATIENT_WORK_PLACE_NAME
- BUYER_TAX_CODE = TDL_PATIENT_TAX_CODE (HIS_TREATMENT)
- BUYER_WORK_PLACE_ID = TDL_PATIENT_WORK_PLACE_ID
- CASHIER_LOGINNAME = Tài khoản thu ngân cấu hình trong key MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO
- CASHIER_ROOM_ID = Phòng thu ngân cấu hình trong key MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO
- CASHIER_USERNAME = Tên thu ngân cấu hình trong key MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO
- PAY_FORM_ID = 8
- TRANS_REQ_ID = ID (HIS_TRANS_REQ)
- TRANSACTION_TIME = thời gian hiện tại
- TRANSACTION_TYPE_ID = 3 (thanh toán)
- TREATMENT_ID = TREATMENT_ID (HIS_TRANS_REQ)
- SERE_SERV_AMOUNT = AMOUNT (HIS_TRANS_REQ)
- IS_DIRECTLY BILLING = 1 (thu trực tiếp)
- Nếu HIS_TRANS_REQ có TRANS_REQ_TYPE = 4:
 - ACCOUNT_BOOK_ID = ID sổ cấu hình trong key MOS.HIS_TRANSACTION.QR_PAYMENT.DEPOSIT_INFO
 - AMOUNT = AMOUNT (HIS_TRANS_REQ)
 - BILL_TYPE_ID = BILL_TYPE_ID (HIS_ACCOUNT_BOOK) của sổ thanh toán
 - BUYER_ACCOUNT_NUMBER = TDL_PATIENT_ACCOUNT_NUMBER (HIS_TREATMENT)

- BUYER_ADDRESS = TDL_PATIENT_ADDRESS (HIS_TREATMENT)
- BUYER_NAME = TDL_PATIENT_NAME (HIS_TREATMENT)
- BUYER_ORGANIZATION = Nếu TDL_PATIENT_WORK_PLACE (HIS_TREATMENT) có thông tin thì lấy TDL_PATIENT_WORK_PLACE. Ngược lại thì lấy thông tin TDL_PATIENT_WORK_PLACE_NAME
- BUYER_TAX_CODE = TDL_PATIENT_TAX_CODE (HIS_TREATMENT)
- BUYER_WORK_PLACE_ID = TDL_PATIENT_WORK_PLACE_ID
- CASHIER_LOGINNAME = Tài khoản thu ngân cấu hình trong key *MOS.HIS_TRANSACTION.QR_PAYMENT.DEPOSIT_INFO*
- CASHIER_ROOM_ID = Phòng thu ngân cấu hình trong key *MOS.HIS_TRANSACTION.QR_PAYMENT.DEPOSIT_INFO*
- CASHIER_USERNAME = Tên thu ngân cấu hình trong key *MOS.HIS_TRANSACTION.QR_PAYMENT.DEPOSIT_INFO*
- PAY_FORM_ID = 8
- TRANS_REQ_ID = ID (HIS_TRANS_REQ)
- TRANSACTION_TIME = thời gian hiện tại
- TRANSACTION_TYPE_ID = 1 (tạm ứng)
- TREATMENT_ID = TREATMENT_ID (HIS_TRANS_REQ)
- Nếu HIS_TRANS_REQ có TRANS_REQ_TYPE = 3 (Thanh toán theo tổng chi phí còn thiếu):
 - ACCOUNT_BOOK_ID = ID sổ cấu hình trong key *MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO*
 - AMOUNT = AMOUNT (HIS_TRANS_REQ)
 - BILL_TYPE_ID = BILL_TYPE_ID (HIS_ACCOUNT_BOOK) của sổ thanh toán cấu hình trong key *MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO*
 - BUYER_ACCOUNT_NUMBER = TDL_PATIENT_ACCOUNT_NUMBER (HIS_TREATMENT)
 - BUYER_ADDRESS = TDL_PATIENT_ADDRESS (HIS_TREATMENT)
 - BUYER_NAME = TDL_PATIENT_NAME (HIS_TREATMENT)
 - BUYER_ORGANIZATION = Nếu TDL_PATIENT_WORK_PLACE (HIS_TREATMENT) có thông tin thì lấy TDL_PATIENT_WORK_PLACE. Ngược lại thì lấy thông tin TDL_PATIENT_WORK_PLACE_NAME
 - BUYER_TAX_CODE = TDL_PATIENT_TAX_CODE (HIS_TREATMENT)

- BUYER_WORK_PLACE_ID = TDL_PATIENT_WORK_PLACE_ID
- CASHIER_LOGINNAME = Tài khoản thu ngân cấu hình trong key
MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO
- CASHIER_ROOM_ID = Phòng thu ngân cấu hình trong key
MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO
- CASHIER_USERNAME = Tên thu ngân cấu hình trong key
MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO
- PAY_FORM_ID = 8
- TRANS_REQ_ID = ID (HIS_TRANS_REQ)
- TRANSACTION_TIME = thời gian hiện tại
- TRANSACTION_TYPE_ID = 3 (thanh toán)
- TREATMENT_ID = TREATMENT_ID (HIS_TRANS_REQ)
- SERE_SERV_AMOUNT = AMOUNT (HIS_TRANS_REQ)
- Tự động kết chuyển chi phí đã thanh toán trước đó của hồ sơ: KC_AMOUNT =
NVL(TOTAL_DEPOSIT_AMOUNT,0) + NVL(TOTAL_DEBT_AMOUNT,0) +
NVL(TOTAL_BILL_AMOUNT,0) - NVL(TOTAL_BILL_TRANSFER_AMOUNT,0) -
NVL(TOTAL_REPAY_AMOUNT,0) lấy từ V_HIS_TREATMENT_FEE
- Gọi vào hàm SetTdl (**HIS_TRANSACTION** transaction, **HIS_TREATMENT** treatment) để Lưu các thông tin dư thừa
TDL_...
 - transaction.TDL_TREATMENT_CODE = treatment.TREATMENT_CODE;
 - transaction.TDL_PATIENT_ID = treatment.PATIENT_ID;
 - transaction.TDL_PATIENT_ADDRESS = treatment.TDL_PATIENT_ADDRESS;
 - transaction.TDL_PATIENT_NAME = treatment.TDL_PATIENT_NAME;
 - transaction.TDL_PATIENT_CAREER_NAME = treatment.TDL_PATIENT_CAREER_NAME;
 - transaction.TDL_PATIENT_CODE = treatment.TDL_PATIENT_CODE;
 - transaction.TDL_PATIENT_DISTRICT_CODE = treatment.TDL_PATIENT_DISTRICT_CODE;
 - transaction.TDL_PATIENT_DOB = treatment.TDL_PATIENT_DOB;
 - transaction.TDL_PATIENT_FIRST_NAME = treatment.TDL_PATIENT_FIRST_NAME;

```

transaction.TDL_PATIENT_GENDER_ID = treatment.TDL_PATIENT_GENDER_ID;
transaction.TDL_PATIENT_GENDER_NAME = treatment.TDL_PATIENT_GENDER_NAME;
transaction.TDL_PATIENT_IS_HAS_NOT_DAY_DOB =
treatment.TDL_PATIENT_IS_HAS_NOT_DAY_DOB;
transaction.TDL_PATIENT_LAST_NAME = treatment.TDL_PATIENT_LAST_NAME;
transaction.TDL_PATIENT_MILITARY_RANK_NAME =
treatment.TDL_PATIENT_MILITARY_RANK_NAME;
transaction.TDL_PATIENT_NATIONAL_NAME = treatment.TDL_PATIENT_NATIONAL_NAME;
transaction.TDL_PATIENT_PROVINCE_CODE = treatment.TDL_PATIENT_PROVINCE_CODE;
transaction.TDL_PATIENT_WORK_PLACE = treatment.TDL_PATIENT_WORK_PLACE;
transaction.TDL_PATIENT_WORK_PLACE_NAME = treatment.TDL_PATIENT_WORK_PLACE_NAME;
transaction.TREATMENT_TYPE_ID = treatment.TDL_TREATMENT_TYPE_ID;
transaction.TDL_PATIENT_CLASSIFY_ID = treatment.TDL_PATIENT_CLASSIFY_ID;

```

- BANK_TRANSACTION_CODE = requestId
- BANK_TRANSACTION_TIME = transactionDate
- Nếu HIS_TRANS_REQ có TRANS_REQ_TYPE = 1 hoặc 2 thì xử lý tạo [HIS_SERE_SERV_BILL](#) tương ứng với các dịch vụ của HIS_SESE_TRANS_REQ:
 - BILL_ID = ID (HIS_TRANSACTION)
 - PRICE = PRICE (HIS_SESE_TRANS_REQ)
 - SERE_SERV_ID = SERE_SERV_ID (HIS_SESE_TRANS_REQ)
 - TDL_BILL_TYPE_ID = BILL_TYPE_ID (HIS_ACCOUNT_BOOK) của sổ thanh toán được cấu hình trong key MOS.HIS_TRANSACTION.QR_PAYMENT.BILL_INFO
 - TDL_TREATMENT_ID = TREATMENT_ID (HIS_TREATMENT)
 - Gọi vào hàm SetTdl([HIS_SERE_SERV_BILL](#) ssBill, [HIS_SERE_SERV](#) sereServ) để cập nhật các thông tin:
 - ssBill.TDL_ADD_PRICE = sereServ.ADD_PRICE;
 - ssBill.TDL_AMOUNT = sereServ.AMOUNT;
 - ssBill.TDL_DISCOUNT = sereServ.DISCOUNT;
 - ssBill.TDL_EXECUTE_DEPARTMENT_ID = sereServ.TDL_EXECUTE_DEPARTMENT_ID;

```
ssBill.TDL_HEIN_LIMIT_PRICE = sereServ.HEIN_LIMIT_PRICE;
ssBill.TDL_HEIN_LIMIT_RATIO = sereServ.HEIN_LIMIT_RATIO;
ssBill.TDL_HEIN_NORMAL_PRICE = sereServ.HEIN_NORMAL_PRICE;
ssBill.TDL_HEIN_PRICE = sereServ.HEIN_PRICE;
ssBill.TDL_HEIN_RATIO = sereServ.HEIN_RATIO;
ssBill.TDL_HEIN_SERVICE_TYPE_ID = sereServ.TDL_HEIN_SERVICE_TYPE_ID;
ssBill.TDL_IS_OUT_PARENT_FEE = sereServ.IS_OUT_PARENT_FEE;
ssBill.TDL_LIMIT_PRICE = sereServ.LIMIT_PRICE;
ssBill.TDL_ORIGINAL_PRICE = sereServ.ORIGINAL_PRICE;
ssBill.TDL_OTHER_SOURCE_PRICE = sereServ.OTHER_SOURCE_PRICE;
ssBill.TDL_OVERTIME_PRICE = sereServ.OVERTIME_PRICE;
ssBill.TDL_PATIENT_TYPE_ID = sereServ.PATIENT_TYPE_ID;
ssBill.TDL_PRICE = sereServ.PRICE;
ssBill.TDL_PRIMARY_PRICE = sereServ.PRIMARY_PRICE;
ssBill.TDL_REQUEST_DEPARTMENT_ID = sereServ.TDL_REQUEST_DEPARTMENT_ID;
ssBill.TDL_SERE_SERV_PARENT_ID = sereServ.PARENT_ID;
ssBill.TDL_SERVICE_CODE = sereServ.TDL_SERVICE_CODE;
ssBill.TDL_SERVICE_ID = sereServ.SERVICE_ID;
ssBill.TDL_SERVICE_NAME = sereServ.TDL_SERVICE_NAME;
ssBill.TDL_SERVICE_TYPE_ID = sereServ.TDL_SERVICE_TYPE_ID;
ssBill.TDL_SERVICE_UNIT_ID = sereServ.TDL_SERVICE_UNIT_ID;
ssBill.TDL_TOTAL_HEIN_PRICE = sereServ.VIR_TOTAL_HEIN_PRICE;
ssBill.TDL_TOTAL_PATIENT_PRICE = sereServ.VIR_TOTAL_PATIENT_PRICE;
ssBill.TDL_TOTAL_PATIENT_PRICE_BHYT = sereServ.VIR_TOTAL_PATIENT_PRICE_BHYT;
ssBill.TDL_USER_PRICE = sereServ.USER_PRICE;
ssBill.TDL_VAT_RATIO = sereServ.VAT_RATIO;
ssBill.TDL_REAL_HEIN_PRICE = sereServ.VIR_HEIN_PRICE;
ssBill.TDL_REAL_PATIENT_PRICE = sereServ.VIR_PATIENT_PRICE;
```

- ```
ssBill.TDL_REAL_PRICE = sereServ.VIR_PRICE;
ssBill.TDL_SERVICE_REQ_ID = sereServ.SERVICE_REQ_ID;
```
- Nếu HIS\_TRANS\_REQ có TRANS\_REQ\_TYPE = 4:
    - Sau khi tạo HIS\_TRANSACTION thành công thì xử lý cập nhật DEPOSIT\_ID (HIS\_DEPOSIT\_REQ) = ID của HIS\_TRANSACTION vừa tạo
  - Nếu HIS\_TRANS\_REQ có TRANS\_REQ\_TYPE = 3 thì xử lý tạo HIS\_SERE\_SERV\_BILL tương ứng với các dịch vụ (HIS\_SERE\_SERV) của hồ sơ chưa được thanh toán hoặc có thanh toán nhưng đã bị hủy:
    - BILL\_ID = ID (HIS\_TRANSACTION)  
PRICE = VIR\_TOTAL\_PATIENT\_PRICE (HIS\_SERE\_SERV)  
SERE\_SERV\_ID = ID (HIS\_SERE\_SERV)  
TDL\_BILL\_TYPE\_ID = BILL\_TYPE\_ID (HIS\_ACCOUNT\_BOOK) của sổ thanh toán được cấu hình trong key MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO  
TDL\_TREATMENT\_ID = TREATMENT\_ID (HIS\_TREATMENT)
    - Gọi vào hàm SetTdl để cập nhật các thông tin như trường hợp TRANS\_REQ\_TYPE = 1 hoặc 2.
  - Sau khi thanh toán thành công, Cập nhật HIS\_TRAN\_REQ:
    - BANK\_JSON\_DATA = chuỗi json bao gồm các thông tin của input
    - BANK\_MESSAGE = statusMessage
    - TRANS\_REQ\_STT\_ID = 2 (Hoàn thành)
  - Trong quá trình thực hiện lưu dữ liệu, nếu xảy ra lỗi (Exception) cần thực hiện try-catch, xử lý rollback dữ liệu, ghi log đầy đủ input, output đồng thời trả về output có paymentStatus = “99”

## 2.2. Xác nhận thanh toán BIDV viện 198

### 2.2.1. Cấu hình

Cấu hình web.config của MOS:

- <add key="MOS.MANAGER.BIDV.SECRET\_KEY" value="" />: Giá trị do BIDV cung cấp, để mã hóa dữ liệu. Ví dụ: uYatzPtZW0CIRFzyHXfBAk3RVRXUuUJS

### Cấu hình hệ thống

- *HIS.Desktop.Plugins.PaymentQrCode.BIDVInfo*:
  - Các tham số phục vụ sinh mã QR do BIDV cung cấp
- *MOS.HIS\_TRANSACTION.QR\_PAYMENT.STATUS\_OPTION*: Trạng thái giao dịch thanh toán Qr:
  - 0: Khóa giao dịch (IS\_ACTIVE = 0). Bệnh nhân sẽ không thực hiện được CLS
  - 1: Không khóa giao dịch (IS\_ACTIVE = 1).
- *MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO*: Thông tin giao dịch thanh toán
- *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*: Thông tin giao dịch tạm ứng
- *MOS.HIS\_TRANSACTION.QR\_PAYMENT.CASHIER\_ROOM\_OPTION*:
  - 1: Lấy thông tin phòng thu ngân theo cấu hình hệ thống (*MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO*, *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*)
  - 2: Lấy thông tin phòng thu ngân dựa theo phòng thu ngân được thiết lập ở phòng tạo yêu cầu thanh toán

### 2.2.2. Api/HisTransReq/QrPaymentBIDV/01043

- Input:

| Tên trường | Kiểu   | Mô tả                           | Yêu Cầu  | Độ Dài  |
|------------|--------|---------------------------------|----------|---------|
| code       | String | Mã Lỗi phản trừ tiền khách hàng | Required | Max(10) |

|           |        |                                                         |          |          |
|-----------|--------|---------------------------------------------------------|----------|----------|
| message   | String | Mô tả mã lỗi trừ tiền khách hàng                        | Required | Max(100) |
| msgType   | String | Loại thanh toán<br><br>1: Thanh toán<br><br>2: Đặt hàng | Required | Max(10)  |
| txnId     | String | Mã đơn hàng, Số hóa đơn trong QRCode                    | Required | Max(20)  |
| qrTrace   | String | Số trace giao dịch : không được trùng nhau              | Required | Max(10)  |
| bankCode  | String | Mã ngân hàng thanh toán                                 | Required | Max(10)  |
| mobile    | String | Số điện thoại khách hàng                                | Optional | Max(20)  |
| accountNo | String | Số tài khoản                                            | Optional | Max(30)  |

|              |        |                                                                             |          |          |
|--------------|--------|-----------------------------------------------------------------------------|----------|----------|
| amount       | String | Số tiền thanh toán                                                          | Required | Max(13)  |
| payDate      | String | Thời hạn thanh toán                                                         | Required | Max(14)  |
| merchantCode | String | Mã code quy định cho VNPAY.                                                 | Required | Max(20)  |
| terminalId   | String | Mã terminal                                                                 | Required | Max(8)   |
| name         | String | Tên người nhận hàng. Tối đa không quá 255 ký tự                             | Optional | Max(100) |
| phone        | String | Số điện thoại nhận hàng (Số này có thể giống hoặc khác với mobile bên trên) | Optional | Max(20)  |
| province_id  | String | ID của tỉnh nhận hàng (Theo thông tin từ VnShop)                            | Optional | Max(14)  |



|             |        |                                                                                                                                                                                                                                                                                                                                                               |          |          |
|-------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------|
| district_id | String | ID của quận/huyện nhận hàng (Theo thông tin từ VnShop)                                                                                                                                                                                                                                                                                                        | Optional | Max(14)  |
| address     | String | Địa chỉ nhận hàng                                                                                                                                                                                                                                                                                                                                             | Optional | Max(100) |
| email       | String | Địa chỉ email                                                                                                                                                                                                                                                                                                                                                 | Optional | Max(100) |
| addData     | String | <p>Json QrCodeItemPayment bao gồm các trường sau:</p> <p>QrCodeItemPayment:</p> <ul style="list-style-type: none"> <li>● String: productId;//Ma san pham</li> <li>● String: amount; //Don gia</li> <li>● String: tipAndFee;//Tien Tip/Phi</li> <li>● String: ccy; //Ma tien te</li> <li>● String: qty;//So luong</li> <li>● String :note;//Ghi chu</li> </ul> | Optional | Free     |

|          |        |                                                                                                                                                                                                                                                                                         |          |         |
|----------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------|
| checksum | String | <p>Dữ liệu mã hóa MD5 của chuỗi sau:</p> <pre>code + " " + msgType + " " + txnId + " " + qrTrace +  " + bankCode + " " + mobile + " " + accountNo +  " + amount + " " + payDate + " " + merchantCode + " " + secretKey</pre> <p>Với secretKey sẽ do bên Merchant cung cấp cho VNPAY</p> | Required | Max(32) |
|----------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------|

- Output:

- code (String): Mã lỗi trả về
- message (String): Mô tả mã lỗi đính kèm
- data (Json): Chi tiết lỗi – Không bắt buộc
- checksum: Dữ liệu mã hóa MD5 của chuỗi sau: code + secretKey (dữ liệu lấy từ value key="MOS.MANAGER.BIDV.SECRET\_KEY" của web.config)

| STT | Mã Lỗi | Mô tả mã lỗi |
|-----|--------|--------------|
|-----|--------|--------------|

|    |        |                                                                                                                                                                                                                                                             |
|----|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01 | String | <p>Thiếu hàng trong đơn hàng</p> <p>Thông tin message sẽ hiển thị dạng List Json các sản phẩm thiếu &lt;Products&gt; với nội dung cụ thể bao gồm:</p> <pre>Data{     • String: productid;// Mã sản phẩm     • String: qty;// Số lượng còn trong kho }</pre> |
| 02 | String | <p>Hết toàn bộ hàng trong đơn hàng</p> <p>Thông tin message sẽ hiển thị dạng String : “Kho hàng đã hết sản phẩm”</p>                                                                                                                                        |
| 03 | String | <p>Đơn hàng đã được thanh toán</p> <p>Thông tin message sẽ hiển thị dạng String: “Đơn hàng đã được thanh toán”.</p> <pre>Data{     • String: txnId; // Mã đơn hàng }</pre>                                                                                  |

|    |        |                                                                                                                                                                                                          |
|----|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 04 | String | <p>Lỗi tạo đơn hàng</p> <p>Thông tin message sẽ hiển thị dạng String mô tả mã lỗi exception.</p>                                                                                                         |
| 05 | String | <p>Thông tin message sẽ hiển thị dạng String : “đơn hàng đang được xử lí”</p> <p>Data{</p> <ul style="list-style-type: none"> <li>• String: txnId; // Mã đơn hàng</li> </ul> <p>}</p>                    |
| 06 | String | <p>Thông tin message sẽ hiển thị dạng String : “sai thông tin xác thực”</p>                                                                                                                              |
| 07 | String | <p>Thông tin message sẽ hiển thị dạng String : “số tiền không chính xác”</p> <p>Data{</p> <ul style="list-style-type: none"> <li>• String: amount; // Số tiền chính xác của đơn hàng</li> </ul> <p>}</p> |

|    |        |                                      |
|----|--------|--------------------------------------|
| 08 | String | Giao dịch timeout                    |
| 09 | String | QR hết hạn thanh toán                |
| 11 | String | Dữ liệu đầu vào không đúng định dạng |
| 14 | String | IP bị khóa                           |
| 96 | String | System is maintaining                |
|    |        |                                      |
| 88 | String | Dont post to Merchant                |

|    |        |                                                                                                                                                                  |
|----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00 | String | Thông tin message sẽ hiển thị dạng String : “đặt hàng thành công”<br><br>Data{ <ul style="list-style-type: none"> <li>String: txnId; // Mã đơn hàng</li> </ul> } |
|----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **Validate:**

- Với các trường Input có Yêu cầu là Required thì không được để trống, bắt buộc có thông tin. Nếu không có thông tin thì trả về Output với code = 11
- Nếu code(input) khác “00” thì đừng không xử lý tiếp thì kết thúc xử lý, trả về output với code= 11
- Kiểm tra thông tin terminalId , merchantId so với giá trị của key cấu hình hệ thống MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO tương ứng với 2 thông tin TerminalLabel, MerchantCode :
  - Nếu các thông tin trên không đúng với thông tin cấu hình thì kết thúc xử lý, trả về output với code = “11”
- Checksum:
  - Xử lý tạo ra chuỗi dữ liệu kiểm tra từ các thông tin do bank gửi theo đúng thứ tự gọi là A (dữ liệu bank gửi được mã hóa MD5):
    - code|msgType|txnId|qrTrace|bankCode|mobile| accountNo|amount|payDate|merchantCode|secretKey
  - Từ chuỗi có được, thực hiện kiểm tra thông tin với dữ liệu đầu vào và secretKey (dữ liệu lấy từ value key="MOS.MANAGER.BIDV.SECRET\_KEY" của web.config). Nếu không khớp thì trả về output với code = “06”
- Kiểm tra thông tin txnId:
  - Kiểm tra tồn tại thông tin yêu cầu thanh toán
    - Lấy ra bản ghi trong HIS\_TRANS\_REQ thỏa mãn điều kiện: TRANS\_REQ\_CODE = txnId (**gọi là A**)
    - Nếu không tồn tại A, thì kết thúc xử lý, trả về output với code= “07”
  - Kiểm tra trạng thái của yêu cầu thanh toán
    - Nếu TRANS\_REQ\_STT\_ID (A) = 2 (trạng thái "Hoàn thành"), thì kết thúc xử lý, trả về output với code= “03”

- Nếu TRANS\_REQ\_STT\_ID (A) = (3, 4) (trạng thái "Hủy", "Thất bại"), thì kết thúc xử lý, trả về output với code= "02"
- Kiểm tra Hồ sơ đã khóa viện phí hay chưa:
  - HIS\_TREATMENT có ID = TREATMENT\_ID (A) IS\_ACTIVE = 0 thì kết thúc xử lý
    - Trả về output với code = "09"
    - Cập nhật yêu cầu thanh toán về trạng thái "Thất bại" (TRANS\_REQ\_STT\_ID (A) = 3)
    - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
    - Kết thúc xử lý
- Kiểm tra số tiền của yêu cầu thanh toán
  - Nếu AMOUNT (A) ≠ amount thì:
    - Cập nhật yêu cầu thanh toán về trạng thái "Thất bại" (TRANS\_REQ\_STT\_ID (A) = 3)
    - Trả về output với code= "07"
    - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
    - Kết thúc xử lý
- Nếu TRANS\_REQ\_TYPE(A) = 1,2 thì:
  - Kiểm tra dịch vụ đã được thanh toán/tạm ứng hay chưa. Cụ thể:
    - B1: Lấy thông tin chi tiết yêu cầu thanh toán. Cụ thể lấy các bản ghi HIS\_SESE\_TRANS\_REQ có TRANS\_REQ\_ID = ID (A) (gọi là danh sách B)
    - B2: Từ danh sách B, lấy ra danh sách SERE\_SERV\_ID (gọi là danh sách C)
      - Kiểm tra xem dịch vụ đã thanh toán chưa:
        - Thực hiện truy vấn lấy ra danh sách thanh toán đã thanh toán. Cụ thể, lấy ra HIS\_SERE\_SERV\_BILL thỏa mãn đồng thời:
          - Có SERE\_SERV\_ID thuộc danh sách C
          - Có IS\_CANCEL ≠ 1
        - Nếu danh sách có dữ liệu, thì:
          - Cập nhật yêu cầu thanh toán về trạng thái "Thất bại" (TRANS\_REQ\_STT\_ID (A) = 3)
          - Trả về output với code= "03"

- Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
  - Kết thúc xử lý
- Kiểm tra xem dịch vụ đã tạm ứng chưa:
  - Thực hiện truy vấn lấy ra danh sách thanh toán đã thanh toán. Cụ thể, lấy ra HIS\_SERE\_SERV\_DEPOSIT thỏa mãn đồng thời:
    - Có SERE\_SERV\_ID thuộc danh sách C
    - Có IS\_CANCEL  $\neq$  1
  - Nếu danh sách có dữ liệu, thì:
    - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS\_REQ\_STT\_ID (A) = 3)
    - Trả về output với code= “03”
    - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
    - Kết thúc xử lý
- Kiểm tra xem thông tin chi tiết yêu cầu thanh toán có được có khớp với thông tin chi tiết dịch vụ hay không. Cụ thể:
  - B1: Lấy ra danh sách các dịch vụ tương ứng với yêu cầu thanh toán, gọi là danh sách D. Cụ thể, lấy ra các HIS\_SERE\_SERV có ID thuộc danh sách C
  - B2: Được coi là “khớp” nếu bản ghi HIS\_SESE\_TRANS\_REQ trong danh sách B thì có bản ghi HIS\_SERE\_SERV tương ứng thuộc danh sách D và ngược lại.
    - Được coi là “tương ứng” nếu thỏa mãn đồng thời:
      - HIS\_SESE\_TRANS\_REQ(SERE\_SERV\_ID) = HIS\_SERE\_SERV(ID)
      - HIS\_SESE\_TRANS\_REQ(PRICE) = HIS\_SERE\_SERV(VIR\_TOTAL\_PATIENT\_PRICE)
    - Nếu không khớp thì:
      - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS\_REQ\_STT\_ID (A) = 3)
      - Trả về output với code= “07”



- Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
  - Kết thúc xử lý
- Nếu TRANS\_REQ\_TYPE(A)= 4 thì:
  - Kiểm tra yêu cầu tạm ứng đã được đóng tiền chưa:
    - B1: Lấy thông tin yêu cầu tạm ứng HIS\_DEPOSIT\_REQ có TRANS\_REQ\_ID = ID (A)
    - B2: Nếu B1 có dữ liệu thì kiểm tra nếu yêu cầu tạm ứng đã được thanh toán (DEPOSIT\_ID(HIS\_DEPOSIT\_REQ) ≠ null) thì:
      - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS\_REQ\_STT\_ID (A) = 3)
      - Trả về output với code= “07”
      - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
      - Kết thúc xử lý
  - Kiểm tra số tiền của yêu cầu tạm ứng:
    - Nếu B1 có dữ liệu thì kiểm tra: Nếu AMOUNT (HIS\_DEPOSIT\_REQ) ≠ AMOUNT (A) thì:
      - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS\_REQ\_STT\_ID (A) = 3)
      - Trả về output với code= “07”
      - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
      - Kết thúc xử lý
- Nếu TRANS\_REQ\_TYPE(A) = 3:
  - Kiểm tra số tiền của yêu cầu thanh toán:
    - B1: Lấy tổng chi phí bệnh nhân còn thiếu (Gọi là X) = Round(NVL(TOTAL\_PATIENT\_PRICE,0) - NVL(TOTAL\_DEPOSIT\_AMOUNT,0) - NVL(TOTAL\_DEBT\_AMOUNT,0) - NVL(TOTAL\_BILL\_AMOUNT,0) + NVL(TOTAL\_BILL\_TRANSFER\_AMOUNT,0) + NVL(TOTAL\_REPAY\_AMOUNT,0)) từ V\_HIS\_TREATMENT\_FEE có ID = TREATMENT\_ID (HIS\_TRANS\_REQ) của bản ghi A
    - B2: Nếu X khác A thì:
      - Cập nhật yêu cầu thanh toán về trạng thái “Thất bại” (TRANS\_REQ\_STT\_ID (A) = 3)
      - Trả về output với code= “07”

- Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
    - Kết thúc xử lý
  - Lỗi cấu hình hệ thống không đúng thông tin để tạo giao dịch thanh toán:
    - Trường hợp cấu hình sai thông tin người thu, sổ thu. Ví dụ thông tin không có trong danh mục nhân viên,... thì trả về output với code = “04”
    - Nếu cấu hình có thông tin sổ thì kiểm tra nếu sổ được check “Không tự động tạo sổ”  
IS\_NOT\_GEN\_TRANSACTION\_ORDER (HIS\_ACCOUNT\_BOOK) = 1 thì trả về output với code = “04”.
    - Ghi log hệ thống (log dưới dạng info), nội dung ghi rõ lý do thất bại và output trả về ngân hàng.
    - Kết thúc xử lý
  - Kiểm tra thiết lập phòng thu ngân:
    - Lấy ra thông tin phòng (HIS\_ROOM) tương ứng với REQUEST\_ROOM\_ID(HIS\_TRANS\_REQ), gọi là A
    - Nếu DEFAULT\_CASHIER\_ROOM\_ID(A) = NULL:
      - Trả về mã lỗi là “04” (Lỗi tạo đơn hàng)
      - Ghi log info có dạng “Phòng tạo yêu cầu thanh toán (XXX - YYYY) chưa thiết lập thông tin phòng thu ngân mặc định”.
        - Trong đó: XXX: Là mã phòng ROOM\_CODE (A)
        - Trong đó: YYYY: Là tên phòng ROOM\_NAME (A)
    - Kết thúc xử lý
- **Xử lý:**
  - Lấy thông tin HIS\_TREATMENT tương ứng với TREATMENT\_ID (HIS\_TRANS\_REQ)
  - Tạo giao dịch HIS\_TRANSACTION:
    - Thông tin CASHIER\_ROOM\_ID:
      - Nếu cấu hình MOS.HIS\_TRANSACTION.QR\_PAYMENT.CASHIER\_ROOM\_OPTION = 1:
        - CASHIER\_ROOM\_ID = ID phòng thu ngân theo thông tin được thiết lập trong cấu hình hệ thống  
*MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO* (với trường hợp là giao dịch tạm ứng) *hoặc*  
*MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO* (với giao dịch thanh toán)
      - Nếu cấu hình MOS.HIS\_TRANSACTION.QR\_PAYMENT.CASHIER\_ROOM\_OPTION = 2:
        - CASHIER\_ROOM\_ID = DEFAULT\_CASHIER\_ROOM\_ID (A)

- Thông tin BEFORE\_UNLOCK\_CASHIER\_ROOM\_ID:
  - Trong trường hợp MOS.HIS\_TRANSACTION.QR\_PAYMENT.STATUS\_OPTION = 1 thì lưu:
    - BEFORE\_UL\_CASHIER\_ROOM\_ID: NULL
    - BEFORE\_UL\_CASHIER\_LOGINNAME: NULL
    - BEFORE\_UL\_CASHIER\_USERNAME: NULL
  - Trong trường hợp MOS.HIS\_TRANSACTION.QR\_PAYMENT.STATUS\_OPTION = 0 thì lưu các thông tin như sau:
    - BEFORE\_UL\_CASHIER\_ROOM\_ID = CASHIER\_ROOM\_ID
    - BEFORE\_UL\_CASHIER\_LOGINNAME = CASHIER\_LOGINNAME
    - BEFORE\_UL\_CASHIER\_USERNAME = CASHIER\_USERNAME
- Nếu HIS\_TRANS\_REQ có TRANS\_REQ\_TYPE = 1 hoặc 2:
  - ACCOUNT\_BOOK\_ID = ID sổ cấu hình trong key *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*
  - AMOUNT = AMOUNT (HIS\_TRANS\_REQ)
  - BILL\_TYPE\_ID = BILL\_TYPE\_ID (HIS\_ACCOUNT\_BOOK) của sổ thanh toán
  - BUYER\_ACCOUNT\_NUMBER = TDL\_PATIENT\_ACCOUNT\_NUMBER (HIS\_TREATMENT)
  - BUYER\_ADDRESS = TDL\_PATIENT\_ADDRESS (HIS\_TREATMENT)
  - BUYER\_NAME = TDL\_PATIENT\_NAME (HIS\_TREATMENT)
  - BUYER\_ORGANIZATION = Nếu TDL\_PATIENT\_WORK\_PLACE (HIS\_TREATMENT) có thông tin thì lấy TDL\_PATIENT\_WORK\_PLACE. Ngược lại thì lấy thông tin TDL\_PATIENT\_WORK\_PLACE\_NAME
  - BUYER\_TAX\_CODE = TDL\_PATIENT\_TAX\_CODE (HIS\_TREATMENT)
  - BUYER\_WORK\_PLACE\_ID = TDL\_PATIENT\_WORK\_PLACE\_ID
  - CASHIER\_LOGINNAME = Tài khoản thu ngân cấu hình trong key *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*
  - CASHIER\_USERNAME = Tên thu ngân cấu hình trong key *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*
  - PAY\_FORM\_ID = 8
  - TRANS\_REQ\_ID = ID (HIS\_TRANS\_REQ)
  - TRANSACTION\_TIME = thời gian hiện tại

- TRANSACTION\_TYPE\_ID = 1 (tạm ứng)
  - TREATMENT\_ID = TREATMENT\_ID (HIS\_TRANS\_REQ)
  - TDL\_SERE\_SERV\_DEPOSIT\_COUNT = số bản ghi dịch vụ tạm thu (HIS\_SESE\_TRANS\_REQ tương ứng)
  - IS\_ACTIVE = giá trị của cấu hình MOS.HIS\_TRANSACTION.QR\_STATUS\_OPTION
- Nếu HIS\_TRANS\_REQ có TRANS\_REQ\_TYPE = 4:
- ACCOUNT\_BOOK\_ID = ID sổ cấu hình trong key *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*
  - AMOUNT = AMOUNT (HIS\_TRANS\_REQ)
  - BILL\_TYPE\_ID = BILL\_TYPE\_ID (HIS\_ACCOUNT\_BOOK) của sổ thanh toán
  - BUYER\_ACCOUNT\_NUMBER = TDL\_PATIENT\_ACCOUNT\_NUMBER (HIS\_TREATMENT)
  - BUYER\_ADDRESS = TDL\_PATIENT\_ADDRESS (HIS\_TREATMENT)
  - BUYER\_NAME = TDL\_PATIENT\_NAME (HIS\_TREATMENT)
  - BUYER\_ORGANIZATION = Nếu TDL\_PATIENT\_WORK\_PLACE (HIS\_TREATMENT) có thông tin thì lấy TDL\_PATIENT\_WORK\_PLACE. Ngược lại thì lấy thông tin TDL\_PATIENT\_WORK\_PLACE\_NAME
  - BUYER\_TAX\_CODE = TDL\_PATIENT\_TAX\_CODE (HIS\_TREATMENT)
  - BUYER\_WORK\_PLACE\_ID = TDL\_PATIENT\_WORK\_PLACE\_ID
  - CASHIER\_LOGINNAME = Tài khoản thu ngân cấu hình trong key *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*
  - CASHIER\_ROOM\_ID = Phòng thu ngân cấu hình trong key *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*
  - CASHIER\_USERNAME = Tên thu ngân cấu hình trong key *MOS.HIS\_TRANSACTION.QR\_PAYMENT.DEPOSIT\_INFO*
  - PAY\_FORM\_ID = 8
  - TRANS\_REQ\_ID = ID (HIS\_TRANS\_REQ)
  - TRANSACTION\_TIME = thời gian hiện tại
  - TRANSACTION\_TYPE\_ID = 1 (tạm ứng)
  - TREATMENT\_ID = TREATMENT\_ID (HIS\_TRANS\_REQ)
  - IS\_ACTIVE = giá trị của cấu hình MOS.HIS\_TRANSACTION.QR\_STATUS\_OPTION
- Nếu HIS\_TRANS\_REQ có TRANS\_REQ\_TYPE = 3 (Thanh toán theo tổng chi phí còn thiếu):

- ACCOUNT\_BOOK\_ID = ID sổ cấu hình trong key MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO
- AMOUNT = AMOUNT (HIS\_TRANS\_REQ) + KC\_AMOUNT
- BILL\_TYPE\_ID = BILL\_TYPE\_ID (HIS\_ACCOUNT\_BOOK) của sổ thanh toán cấu hình trong key MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO
- BUYER\_ACCOUNT\_NUMBER = TDL\_PATIENT\_ACCOUNT\_NUMBER (HIS\_TREATMENT)
- BUYER\_ADDRESS = TDL\_PATIENT\_ADDRESS (HIS\_TREATMENT)
- BUYER\_NAME = TDL\_PATIENT\_NAME (HIS\_TREATMENT)
- BUYER\_ORGANIZATION = Nếu TDL\_PATIENT\_WORK\_PLACE (HIS\_TREATMENT) có thông tin thì lấy TDL\_PATIENT\_WORK\_PLACE. Ngược lại thì lấy thông tin TDL\_PATIENT\_WORK\_PLACE\_NAME
- BUYER\_TAX\_CODE = TDL\_PATIENT\_TAX\_CODE (HIS\_TREATMENT)
- BUYER\_WORK\_PLACE\_ID = TDL\_PATIENT\_WORK\_PLACE\_ID
- CASHIER\_LOGINNAME = Tài khoản thu ngân cấu hình trong key MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO
- CASHIER\_ROOM\_ID = Phòng thu ngân cấu hình trong key MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO
- CASHIER\_USERNAME = Tên thu ngân cấu hình trong key MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO
- PAY\_FORM\_ID = 8
- TRANS\_REQ\_ID = ID (HIS\_TRANS\_REQ)
- TRANSACTION\_TIME = thời gian hiện tại
- TRANSACTION\_TYPE\_ID = 3 (thanh toán)
- TREATMENT\_ID = TREATMENT\_ID (HIS\_TRANS\_REQ)
- SERE\_SERV\_AMOUNT = AMOUNT
- Tự động kết chuyển chi phí đã tạm ứng, hoàn ứng, kết chuyển trước đó của hồ sơ:  $KC\_AMOUNT = NVL(TOTAL\_DEPOSIT\_AMOUNT,0) - NVL(TOTAL\_REPAY\_AMOUNT,0) - NVL(TOTAL\_BILL\_TRANSFER\_AMOUNT,0)$
- IS\_ACTIVE = giá trị của cấu hình MOS.HIS\_TRANSACTION.QR\_STATUS\_OPTION

- Gọi vào hàm SetTdl (**HIS\_TRANSACTION** transaction, **HIS\_TREATMENT** treatment) để Lưu các thông tin dư thừa TDL\_...
  - transaction.TDL\_TREATMENT\_CODE = treatment.TREATMENT\_CODE;  
 transaction.TDL\_PATIENT\_ID = treatment.PATIENT\_ID;  
 transaction.TDL\_PATIENT\_ADDRESS = treatment.TDL\_PATIENT\_ADDRESS;  
 transaction.TDL\_PATIENT\_NAME = treatment.TDL\_PATIENT\_NAME;  
 transaction.TDL\_PATIENT\_CAREER\_NAME = treatment.TDL\_PATIENT\_CAREER\_NAME;  
 transaction.TDL\_PATIENT\_CODE = treatment.TDL\_PATIENT\_CODE;  
 transaction.TDL\_PATIENT\_DISTRICT\_CODE = treatment.TDL\_PATIENT\_DISTRICT\_CODE;  
 transaction.TDL\_PATIENT\_DOB = treatment.TDL\_PATIENT\_DOB;  
 transaction.TDL\_PATIENT\_FIRST\_NAME = treatment.TDL\_PATIENT\_FIRST\_NAME;  
 transaction.TDL\_PATIENT\_GENDER\_ID = treatment.TDL\_PATIENT\_GENDER\_ID;  
 transaction.TDL\_PATIENT\_GENDER\_NAME = treatment.TDL\_PATIENT\_GENDER\_NAME;  
 transaction.TDL\_PATIENT\_IS\_HAS\_NOT\_DAY\_DOB =  
 treatment.TDL\_PATIENT\_IS\_HAS\_NOT\_DAY\_DOB;  
 transaction.TDL\_PATIENT\_LAST\_NAME = treatment.TDL\_PATIENT\_LAST\_NAME;  
 transaction.TDL\_PATIENT\_MILITARY\_RANK\_NAME =  
 treatment.TDL\_PATIENT\_MILITARY\_RANK\_NAME;  
 transaction.TDL\_PATIENT\_NATIONAL\_NAME = treatment.TDL\_PATIENT\_NATIONAL\_NAME;  
 transaction.TDL\_PATIENT\_PROVINCE\_CODE = treatment.TDL\_PATIENT\_PROVINCE\_CODE;  
 transaction.TDL\_PATIENT\_WORK\_PLACE = treatment.TDL\_PATIENT\_WORK\_PLACE;  
 transaction.TDL\_PATIENT\_WORK\_PLACE\_NAME = treatment.TDL\_PATIENT\_WORK\_PLACE\_NAME;  
 transaction.TREATMENT\_TYPE\_ID = treatment.TDL\_TREATMENT\_TYPE\_ID;  
 transaction.TDL\_PATIENT\_CLASSIFY\_ID = treatment.TDL\_PATIENT\_CLASSIFY\_ID;
  - BANK\_TRANSACTION\_CODE = qrTrace của input
  - BANK\_TRANSACTION\_TIME = payDate của input
- Nếu HIS\_TRANS\_REQ có TRANS\_REQ\_TYPE = 1 hoặc 2 thì xử lý tạo **HIS\_SERE\_SERV\_DEPOSIT** tương ứng với các dịch vụ của HIS\_SESE\_TRANS\_REQ, HIS\_SERE\_SERV:

- DEPOSIT\_ID = ID (HIS\_TRANSACTION)
  - AMOUNT = PRICE (HIS\_SESE\_TRANS\_REQ)
  - SERE\_SERV\_ID = SERE\_SERV\_ID (HIS\_SESE\_TRANS\_REQ)
  - TDL\_AMOUNT = AMOUNT (HIS\_SERE\_SERV)
  - TDL\_TREATMENT\_ID = TREATMENT\_ID (HIS\_TREATMENT)
  - TDL\_EXECUTE\_DEPARTMENT\_ID = TDL\_EXECUTE\_DEPARTMENT\_ID (HIS\_SERE\_SERV)
  - TDL\_HEIN\_LIMIT\_PRICE = HEIN\_LIMIT\_PRICE (HIS\_SERE\_SERV)
  - TDL\_HEIN\_PRICE = HEIN\_PRICE (HIS\_SERE\_SERV)
  - TDL\_HEIN\_SERVICE\_TYPE\_ID = TDL\_HEIN\_SERVICE\_TYPE\_ID (HIS\_SERE\_SERV)
  - TDL\_IS\_EXPEND = IS\_EXPEND (HIS\_SERE\_SERV)
  - TDL\_IS\_OUT\_PARENT\_FEE = IS\_OUT\_PARENT\_FEE (HIS\_SERE\_SERV)
  - TDL\_PATIENT\_TYPE\_ID = PATIENT\_TYPE\_ID (HIS\_SERE\_SERV)
  - TDL\_REQUEST\_DEPARTMENT\_ID = TDL\_REQUEST\_DEPARTMENT\_ID (HIS\_SERE\_SERV)
  - TDL\_SERE\_SERV\_PARENT\_ID = PARENT\_ID (HIS\_SERE\_SERV)
  - TDL\_SERVICE\_CODE = TDL\_SERVICE\_CODE (HIS\_SERE\_SERV)
  - TDL\_SERVICE\_ID = SERVICE\_ID (HIS\_SERE\_SERV)
  - TDL\_SERVICE\_NAME = TDL\_SERVICE\_NAME (HIS\_SERE\_SERV)
  - TDL\_SERVICE\_REQ\_ID = SERVICE\_REQ\_ID (HIS\_SERE\_SERV)
  - TDL\_SERVICE\_TYPE\_ID = TDL\_SERVICE\_TYPE\_ID (HIS\_SERE\_SERV)
  - TDL\_SERVICE\_UNIT\_ID = TDL\_SERVICE\_UNIT\_ID (HIS\_SERE\_SERV)
  - IS\_ACTIVE = giá trị của cấu hình MOS.HIS\_TRANSACTION.QR\_STATUS\_OPTION
- Nếu HIS\_TRANS\_REQ có TRANS\_REQ\_TYPE = 4:
    - Sau khi tạo HIS\_TRANSACTION thành công thì xử lý cập nhật DEPOSIT\_ID (HIS\_DEPOSIT\_REQ) = ID của HIS\_TRANSACTION vừa tạo
  - Nếu HIS\_TRANS\_REQ có TRANS\_REQ\_TYPE = 3 thì xử lý tạo HIS\_SERE\_SERV\_BILL tương ứng với các dịch vụ (HIS\_SERE\_SERV) của hồ sơ chưa được thanh toán hoặc có thanh toán nhưng đã bị hủy:
    - BILL\_ID = ID (HIS\_TRANSACTION)
    - PRICE = VIR\_TOTAL\_PATIENT\_PRICE (HIS\_SERE\_SERV)

SERE\_SERV\_ID = ID (HIS\_SERE\_SERV)

TDL\_BILL\_TYPE\_ID = BILL\_TYPE\_ID (HIS\_ACCOUNT\_BOOK) của sổ thanh toán được cấu hình trong key MOS.HIS\_TRANSACTION.QR\_PAYMENT.BILL\_INFO

TDL\_TREATMENT\_ID = TREATMENT\_ID (HIS\_TREATMENT)

○ IS\_ACTIVE = giá trị của cấu hình MOS.HIS\_TRANSACTION.QR\_STATUS\_OPTION

○ Gọi vào hàm SetTdl([HIS\\_SERE\\_SERV\\_BILL](#) ssBill, [HIS\\_SERE\\_SERV](#) sereServ) để cập nhật các thông tin:

- ssBill.TDL\_ADD\_PRICE = sereServ.ADD\_PRICE;  
ssBill.TDL\_AMOUNT = sereServ.AMOUNT;  
ssBill.TDL\_DISCOUNT = sereServ.DISCOUNT;  
ssBill.TDL\_EXECUTE\_DEPARTMENT\_ID = sereServ.TDL\_EXECUTE\_DEPARTMENT\_ID;  
ssBill.TDL\_HEIN\_LIMIT\_PRICE = sereServ.HEIN\_LIMIT\_PRICE;  
ssBill.TDL\_HEIN\_LIMIT\_RATIO = sereServ.HEIN\_LIMIT\_RATIO;  
ssBill.TDL\_HEIN\_NORMAL\_PRICE = sereServ.HEIN\_NORMAL\_PRICE;  
ssBill.TDL\_HEIN\_PRICE = sereServ.HEIN\_PRICE;  
ssBill.TDL\_HEIN\_RATIO = sereServ.HEIN\_RATIO;  
ssBill.TDL\_HEIN\_SERVICE\_TYPE\_ID = sereServ.TDL\_HEIN\_SERVICE\_TYPE\_ID;  
ssBill.TDL\_IS\_OUT\_PARENT\_FEE = sereServ.IS\_OUT\_PARENT\_FEE;  
ssBill.TDL\_LIMIT\_PRICE = sereServ.LIMIT\_PRICE;  
ssBill.TDL\_ORIGINAL\_PRICE = sereServ.ORIGINAL\_PRICE;  
ssBill.TDL\_OTHER\_SOURCE\_PRICE = sereServ.OTHER\_SOURCE\_PRICE;  
ssBill.TDL\_OVERTIME\_PRICE = sereServ.OVERTIME\_PRICE;  
ssBill.TDL\_PATIENT\_TYPE\_ID = sereServ.PATIENT\_TYPE\_ID;  
ssBill.TDL\_PRICE = sereServ.PRICE;  
ssBill.TDL\_PRIMARY\_PRICE = sereServ.PRIMARY\_PRICE;  
ssBill.TDL\_REQUEST\_DEPARTMENT\_ID = sereServ.TDL\_REQUEST\_DEPARTMENT\_ID;  
ssBill.TDL\_SERE\_SERV\_PARENT\_ID = sereServ.PARENT\_ID;  
ssBill.TDL\_SERVICE\_CODE = sereServ.TDL\_SERVICE\_CODE;  
ssBill.TDL\_SERVICE\_ID = sereServ.SERVICE\_ID;



```

ssBill.TDL_SERVICE_NAME = sereServ.TDL_SERVICE_NAME;
ssBill.TDL_SERVICE_TYPE_ID = sereServ.TDL_SERVICE_TYPE_ID;
ssBill.TDL_SERVICE_UNIT_ID = sereServ.TDL_SERVICE_UNIT_ID;
ssBill.TDL_TOTAL_HEIN_PRICE = sereServ.VIR_TOTAL_HEIN_PRICE;
ssBill.TDL_TOTAL_PATIENT_PRICE = sereServ.VIR_TOTAL_PATIENT_PRICE;
ssBill.TDL_TOTAL_PATIENT_PRICE_BHYT = sereServ.VIR_TOTAL_PATIENT_PRICE_BHYT;
ssBill.TDL_USER_PRICE = sereServ.USER_PRICE;
ssBill.TDL_VAT_RATIO = sereServ.VAT_RATIO;
ssBill.TDL_REAL_HEIN_PRICE = sereServ.VIR_HEIN_PRICE;
ssBill.TDL_REAL_PATIENT_PRICE = sereServ.VIR_PATIENT_PRICE;
ssBill.TDL_REAL_PRICE = sereServ.VIR_PRICE;
ssBill.TDL_SERVICE_REQ_ID = sereServ.SERVICE_REQ_ID;

```

- Sau khi thanh toán thành công, Cập nhật HIS\_TRAN\_REQ:
  - BANK\_JSON\_DATA = chuỗi json bao gồm các thông tin của input
  - BANK\_MESSAGE = statusMessage
  - TRANS\_REQ\_STT\_ID = 2 (Hoàn thành)
- Trong quá trình thực hiện lưu dữ liệu, nếu xảy ra lỗi (Exception) cần thực hiện try-catch, xử lý rollback dữ liệu, ghi log đầy đủ input, output đồng thời trả về output có code = “04”, message là nội dung của Exception