MobilePay AppSwitch Implementation Guide

Version 2.2

Table of contents

MobilePay AppSwitch Implementation Guide

1 Change log

Version	Date	Amendment
1.0	01.10.2014	Document created
1.1	08.10.2014	Error scenarios added.
1.2	31.10.2014	More details added by Danske Bank and Trifork.
1.3	28.11.2014	More details added by Danske Bank and Trifork.
1.31	15.01.2015	Chapter 10 updated.
2.0	26.03.2015	TKI: Document updated to support latest AppSwitch release 1.6
2.1	27.05.2015	TKI: Document edited to treat AppSwitch only (SDK)
2.2	11-06-2015	TKI: Document synchronised with GitHub

MobilePay AppSwitch Implementation Guide

2 Purpose of the implementation guide

This implementation guide explains the design and implementation process of MobilePay AppSwitch. This includes descriptions of MobilePay AppSwitch communication, error scenarios, security, and test setup.

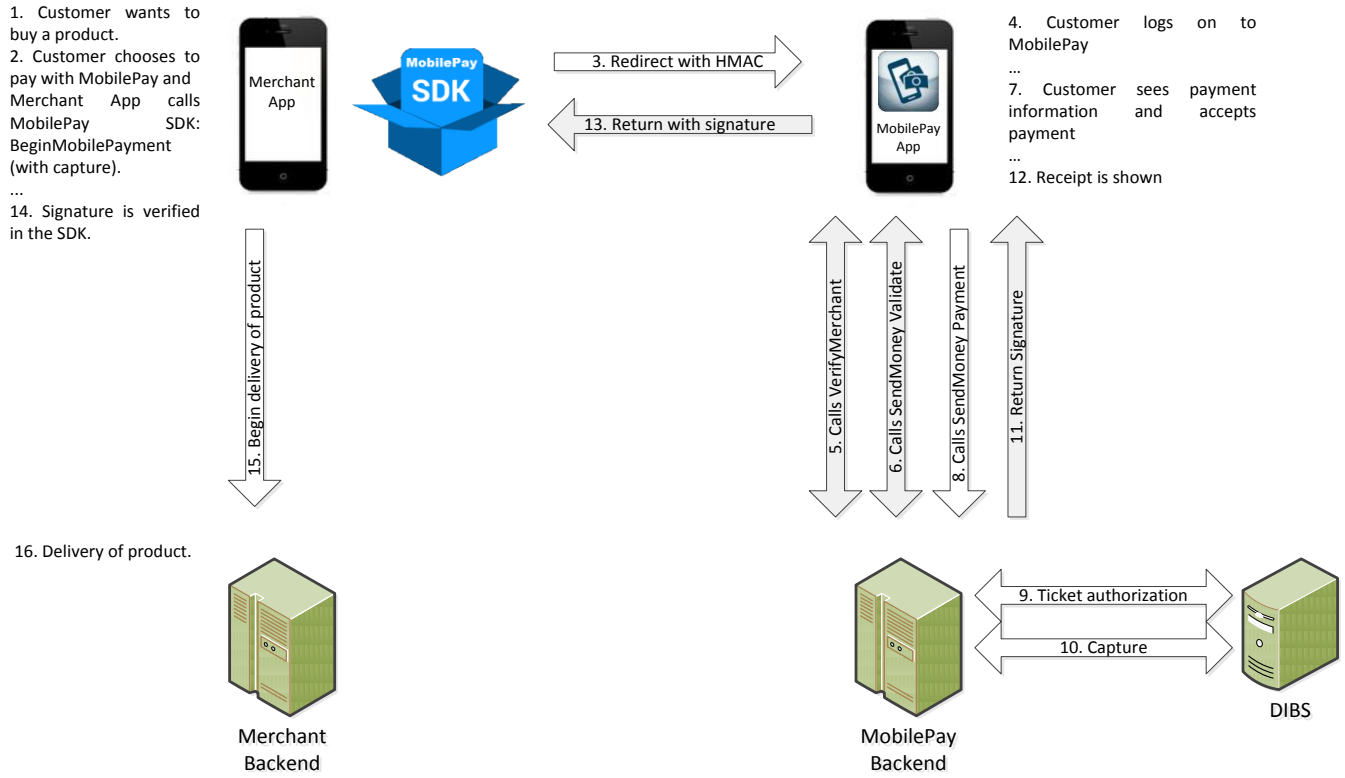
2.1 Target group

The target group is project managers, system architects, and developers.

Implementation details such as coding details and platform specific details (Android, Windows Phone, and iPhone) are not part of this guide. They can be found on GitHub under Danske Bank's AppSwitch SDK repository.

MobilePay AppSwitch Implementation Guide

3 MobilePay AppSwitch



4 Communication

The merchant app communicates with MobilePay AppSwitch SDK and typically also with a merchant backend.

The MobilePay app communicates with MobilePay AppSwitch SDK and MobilePay Backend.

Selection of the MobilePay payment option in the merchant app implies that the MobilePay AppSwitch SDK redirects the end customer to the MobilePay app. This redirection includes verification at the MobilePay backend of the merchant ID (also referred to as AppSwitch ID) and specific secure data (HMAC calculation based upon an agreed key) used for validation of current message content sent from merchant app.

The customer follows the well-known MobilePay steps as logon, payment approval (swipe), and payment completion.

When the requested payment is completed in MobilePay, a signature is returned to the merchant app. The signed information contains the original merchant order ID and the MobilePay transaction ID. The MobilePay AppSwitch SDK validates the signature and ensures that the order ID given to the MobilePay AppSwitch SDK by the merchant app matches the order ID received from MobilePay.

Verification of the digital signature and error handling are both handled in the MobilePay AppSwitch SDK.

The MobilePay app uses different services to communicate with the MobilePay backend at Danske Bank.

The services can be divided into two types; authenticated and non-authenticated services. Authenticated services require a PIN code, installed MobilePay app ID and related MobilePay mobile phone number. Non-authenticated services include services for registration of new users, getting app release information, help texts, contact information etc.

Examples of MobilePay app services:

1. VerifyMerchant: Verifies that the HMAC and the merchant ID are correct.
2. Logon: Enable MobilePay access (authenticated)
3. SendMoneyAppSwitch: Transfers the money from the customer to the merchant account. This step involves creation of a transaction ID (payment ID) which is the signature of a payment. The SendMoneyAppSwitch service has two actions: VALIDATE and PAYMENT. The VALIDATE action verifies whether the customer can make the payment (sufficient funds and card status), but no payment is done. The PAYMENT action executes the actual payment. The VALIDATE action

also verifies whether the current specified order ID (provided from the Merchant App) in the current payment request has already been paid. If already paid, the payment details and signature are returned (authenticated).

4. ShowList: Returns all the customers' transactions in MobilePay (authenticated).
5. ShowDetails: Returns the payment receipt and transaction ID in MobilePay (authenticated).
6. AppVersion: Returns the minimum MobilePay app version required.

All security operations are executed in the MobilePay backend and not in the MobilePay app.

Furthermore, there is no need for any user administration or access control for the merchant to servers at Danske Bank and there is no need for the merchant to host servers that Danske Bank needs access to.

4.1 Communication between the merchant app and MobilePay app

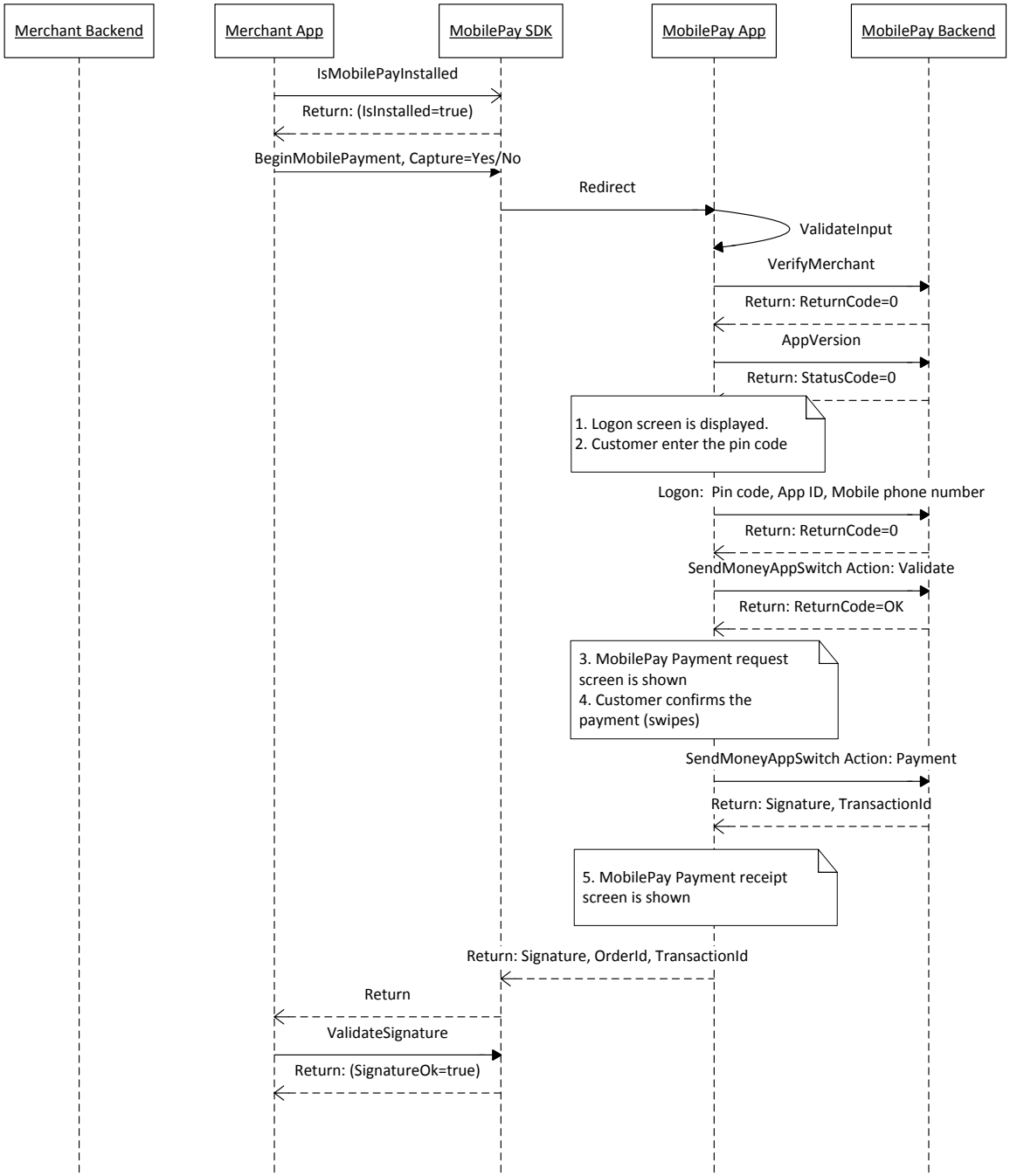
The data exchange between the merchant app and MobilePay app and vice versa is defined in sections 6.1 and 6.2, respectively. The merchant app utilises the MobilePay AppSwitch SDK to initiate a payment that redirects the payment inputs to the MobilePay app.

The MobilePay app validates the input and requests the user to login. After login, the user gets a payment request once payment validation has taken place (user looked up in DIBS regarding ability to pay with current pay card in MobilePay).

When the user accepts (swipes), the payment is authorised at DIBS. The "Capture" parameter from the SDK can either be 'Y' (yes) or 'N' (no). By default, "Capture" is set to "Y". If set to "N", a reservation takes place – this is a non-supported feature of the current version of AppSwitch.

The request is confirmed in MobilePay backend – a receipt in response is sent to the MobilePay app. The MobilePay app then redirects the customer back to the merchant app with a signature of the payment.

If the transaction is captured, the signature will contain the status of the payment.



If these steps are successful, the order can be delivered.

4.2 How to avoid double payments

The MobilePay app has a built-in validation feature which ensures that the customer will not accidently pay for the same service/product twice.

1. If the merchant app sends a new order ID to the MobilePay app, the customer is then able to pay for the order, and the money will be transferred to the merchant's account.

2. If the merchant app sends an order ID to the MobilePay app, which the customer already has paid for, the payment signature including the existing transaction ID (payment ID) is returned without a new payment taking place.

MobilePay will ensure a one-to-one relation between these artefacts:

- Order ID (from merchant app)
- Actual payment (money transfer)

Normally, there is also a one-to-one relation between order ID and transaction ID:

- Transaction ID (payment ID from DIBS).

The customer is redirected from WIFI to phone network if any network issues occur. The network might resend the transaction, but double payments will not occur, because the second payment will be rejected due to the same order ID not being allowed for payment confirmation.

4.3 How to ensure authentication of the calling merchant app

The merchant is registered in the MobilePay backend with an AppSwitch ID (also referred to as merchant ID). This ID is attached to a Business Online agreement.

The MobilePay backend will verify the merchant in the following steps:

- Is the merchant ID registered in the MobilePay backend database?
- Is it registered as being active?
- Is the HMAC correct (please refer to the section "Security and Certificates" in chapter 6)

4.4 How to ensure authentication of the payment

A MobilePay AppSwitch payment returns a digital signature to the calling merchant app. This signature ensures the authenticity of the

payment, because the Danske Bank private key has been used for signing and therefore the Danske Bank public key can be used for verification at the merchant side. The public key is contained in a certificate and exchanged in a secure manner separately from the payment transaction.

MobilePay AppSwitch Implementation Guide

5 Error codes and error scenarios

The following sections describe the different error codes that can be returned from the MobilePay AppSwitch SDK and general error scenarios.

The error codes from the MobilePay AppSwitch SDK are listed in the table below:

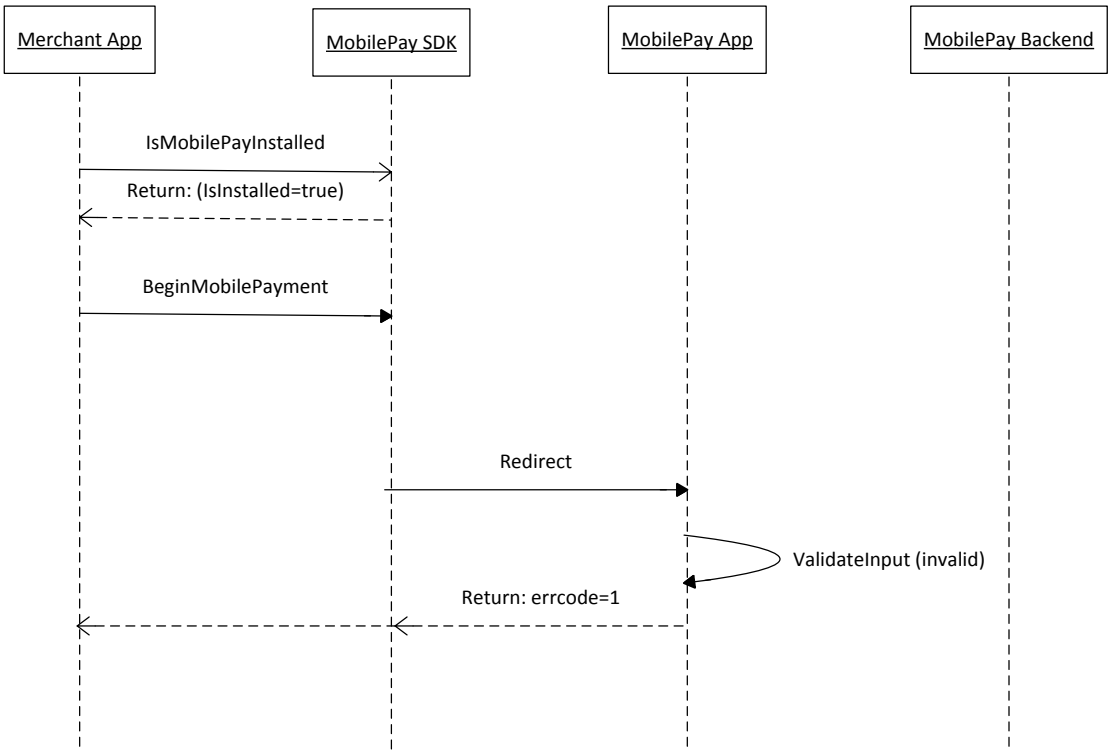
No.	Type
1	Invalid parameters to MobilePay app
2	Validate merchant request fails
3	MobilePay app version is out of date
4	Merchant ID is not valid
5	HMAC parameter is not valid
6	MobilePay timeout
7	MobilePay amount exceeded
8	Merchant app timeout
9	Invalid signature – Danske Bank has not signed it.
10	MobilePay AppSwitch SDK version is outdated
11	The order ID sent to MobilePay has already been used for a confirmed payment by the same merchant.

Most of these errors are technical errors (1,2,4,5,9,10,11) that must be presented as a generic error message without details to the user, where others ***should be handled by the merchant app*** (3,6,7,8) to guide the user in the right direction.

5.1 Invalid parameters to MobilePay app

The MobilePay AppSwitch SDK returns error code no. 1 to the merchant app if the input sent to the MobilePay app is invalid. The input can be invalid, e.g. if the price is lower than 0 or required input is missing.

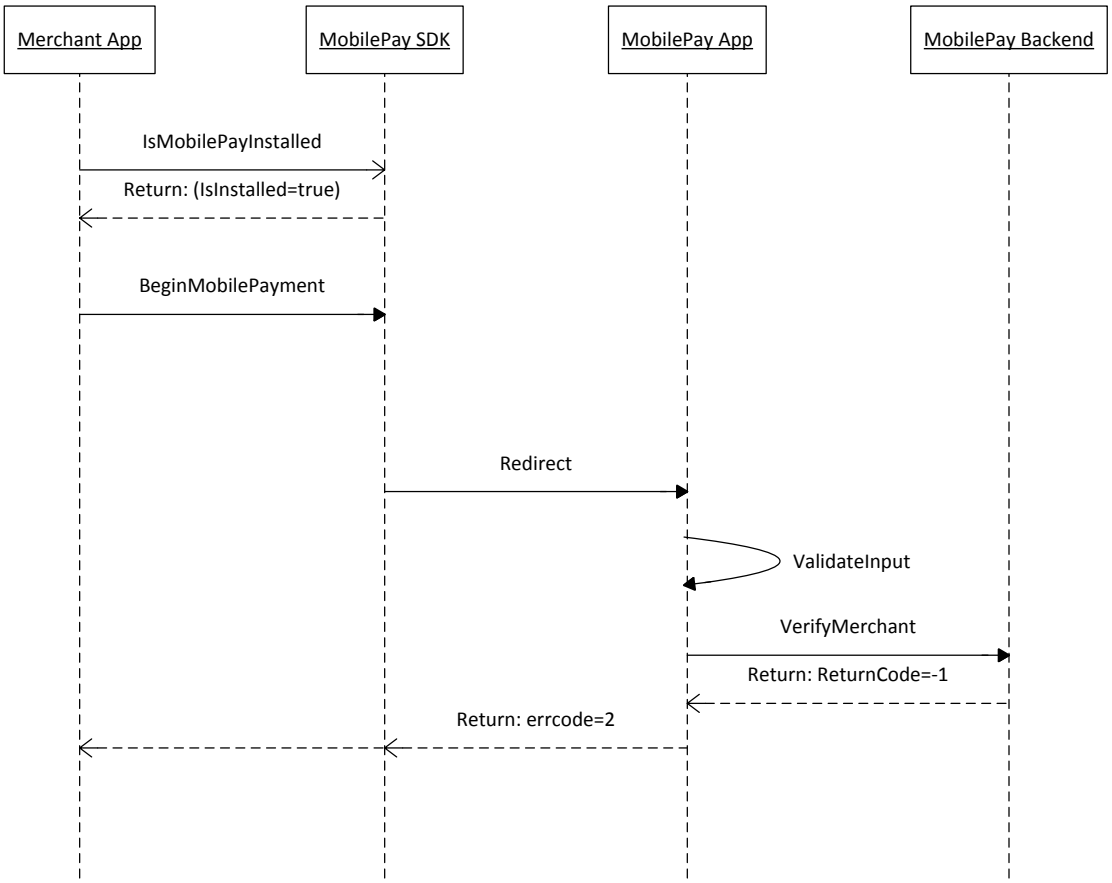
This error code should never be received in the merchant app in production. The error code should be handled in the merchant app by showing a message to the user and creating a log in the merchant backend (if this is possible) that a problem with initiating a payment request has occured.



5.2 Validate merchant request fails

The MobilePay AppSwitch SDK will return error code no. 2 to the merchant app if the validation of the merchant failed due to network failure or timeout. The MobilePay app validates the merchant ID sent to it by making a validation request to the MobilePay backend.

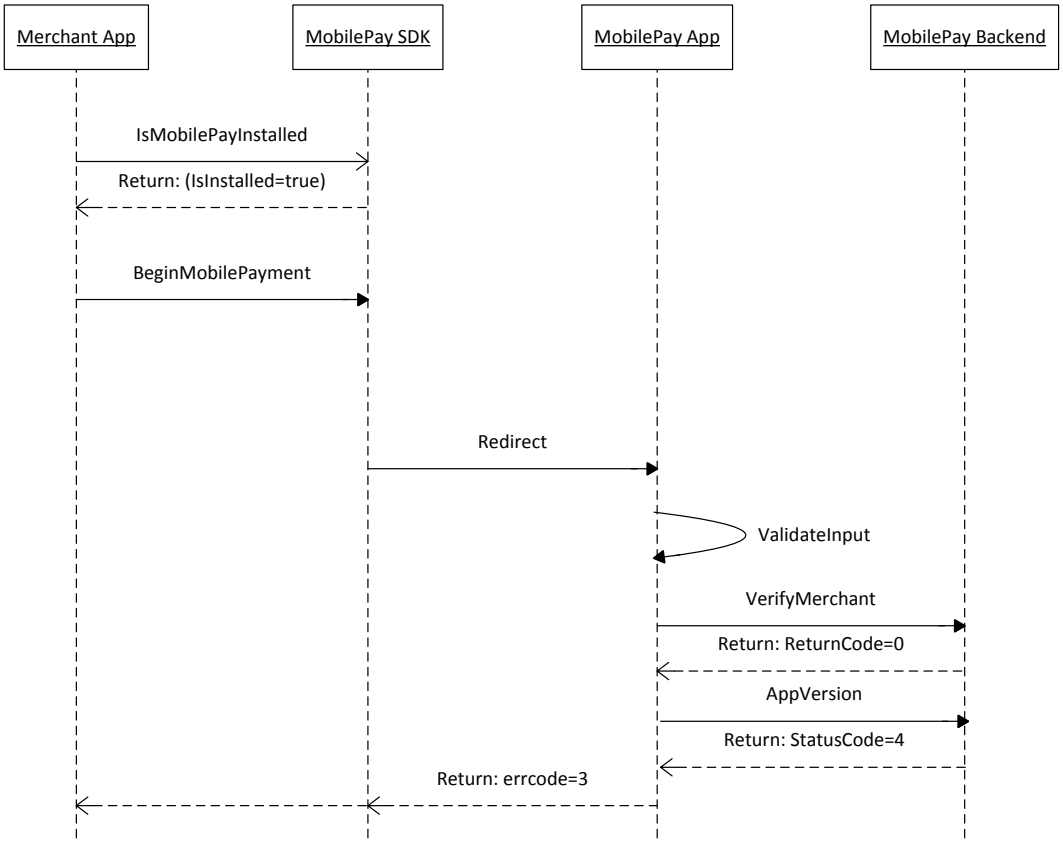
This error code should be handled in the merchant app by showing a message asking the user to check network connectivity and try again.



5.3 MobilePay app is out of date and must be updated

The MobilePay AppSwitch SDK returns error code no. 3 if the MobilePay app must be updated because of security or legal requirements.

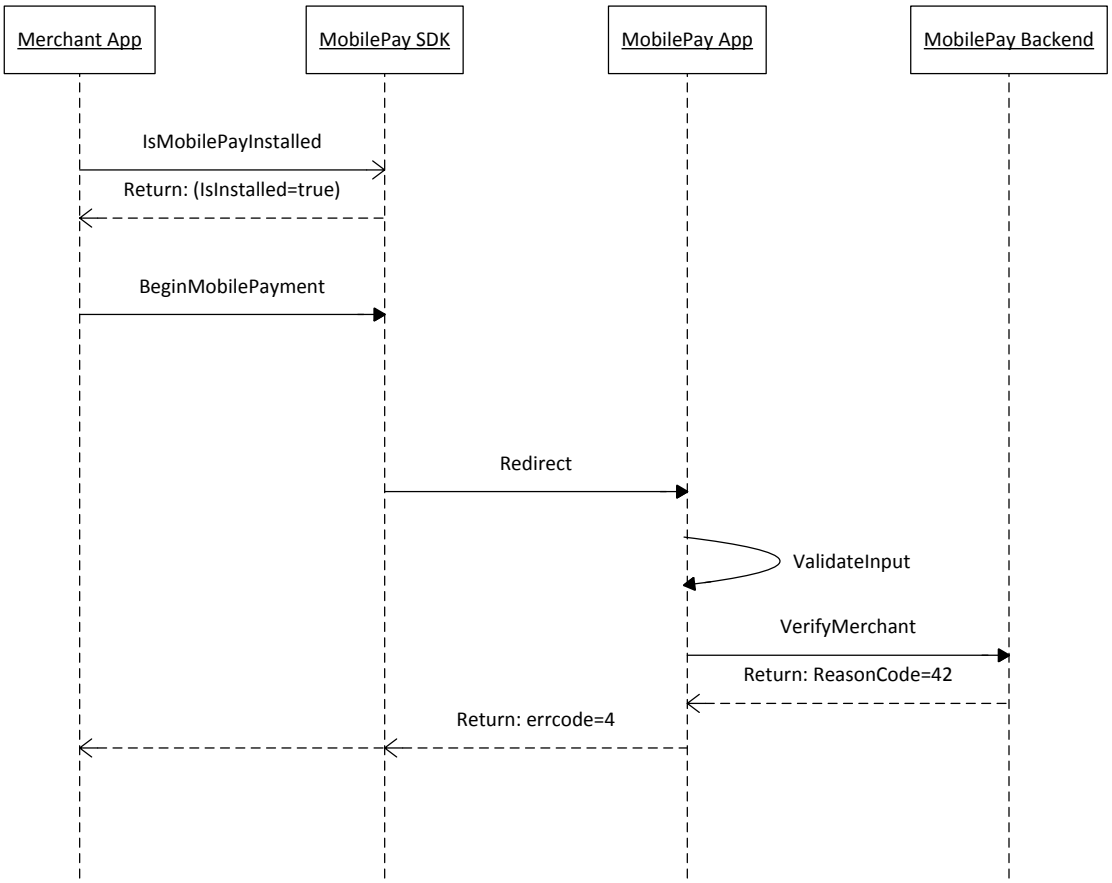
It is the responsibility of the merchant app to inform the user that the MobilePay app must be updated before trying again. The merchant app can show a message to the user and provide a link to an app store to download the latest version of MobilePay.



5.4 MerchantID is not valid

The MobilePay AppSwitch SDK will return error code no. 4 if the merchant ID received by the MobilePay app is invalid. The MobilePay app validates the merchant ID by making a validation request to the MobilePay backend.

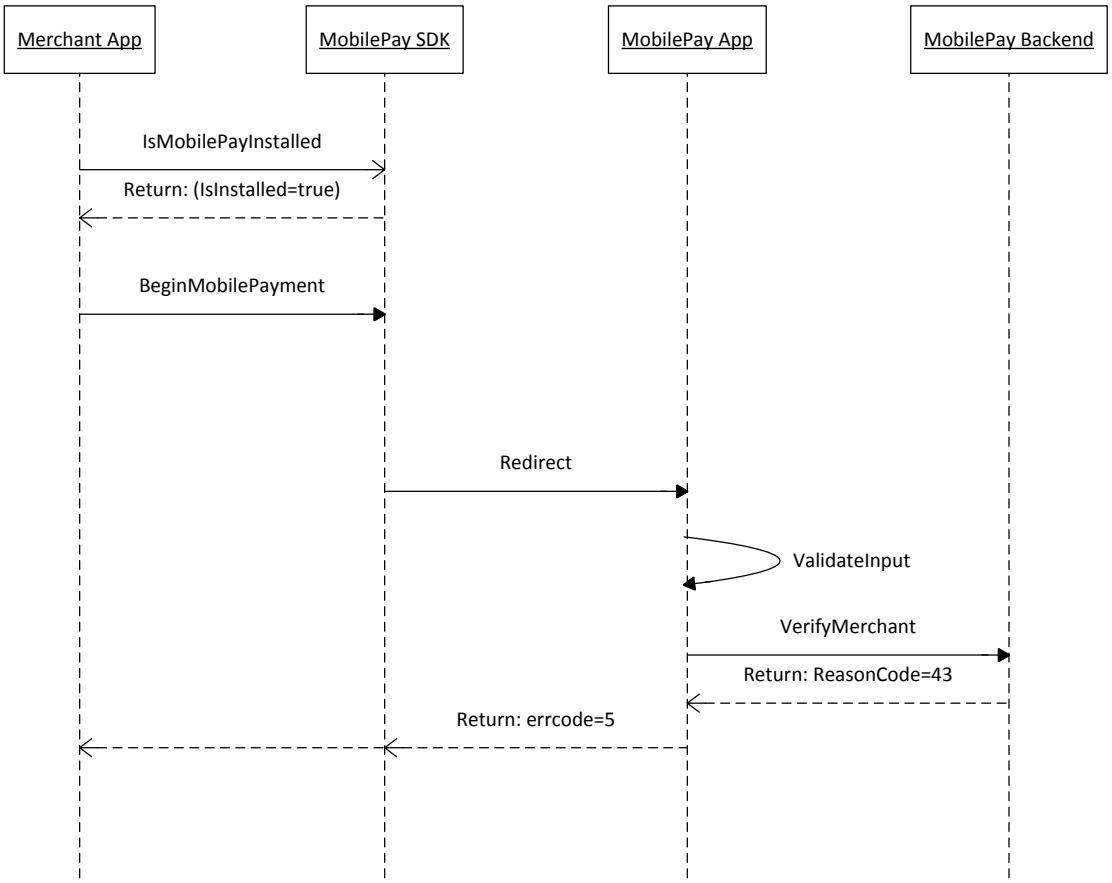
This error code should never be received in the merchant app in production. The error code should be handled in the merchant app, by showing a message to the user and logging it to the merchant backend (if possible).



5.5 HMAC parameter is not valid

The MobilePay AppSwitch SDK will return error code no. 5 to the merchant app, if the HMAC parameter received by the MobilePay app is not valid due to value not matching – if not caused by corrupted data it may be caused by missing synchronisation in the understanding of calculation – calculation method, content of message, and/or wrong key used.

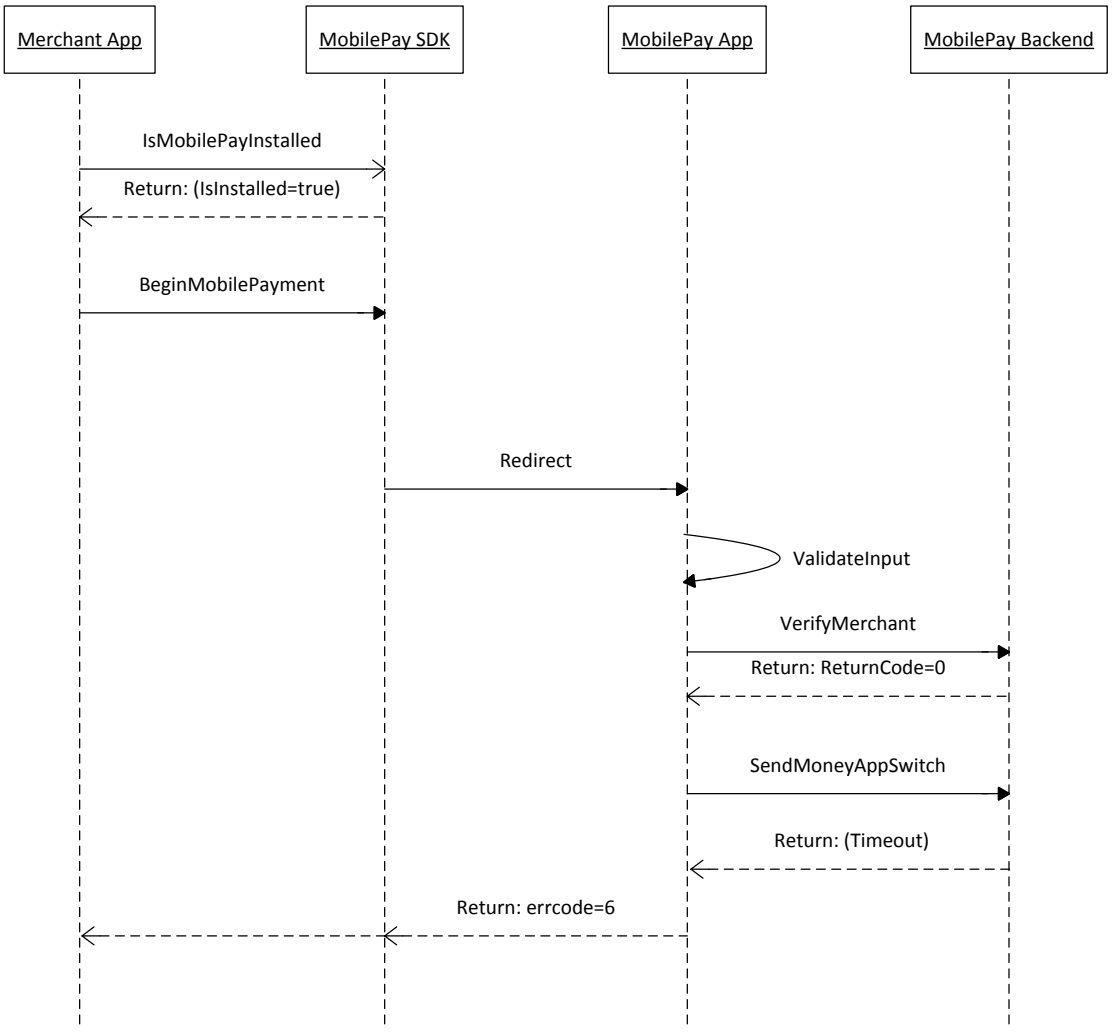
This error code should never be received in the merchant app in production. The error code should be handled in the merchant app by showing a message to the user and log it to the merchant backend (if possible).



5.6 MobilePay timeout

The MobilePay AppSwitch SDK returns error code no. 6 to the merchant app if the call to the MobilePay backend (SendMoneyAppSwitch handles the payment) times out, which by default is set to 5 minutes.

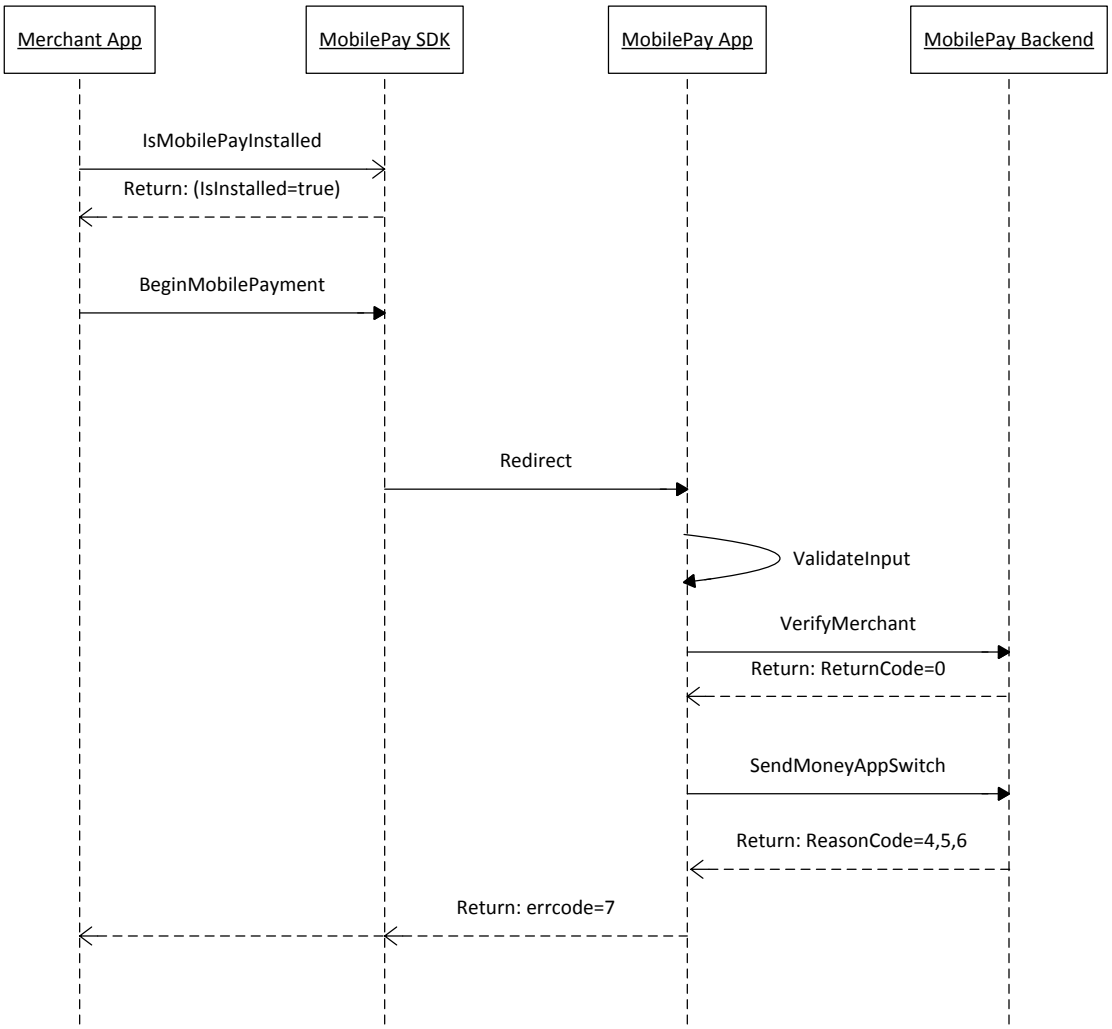
The merchant app should show a message informing the user about the timeout and let the user try the same transaction again. The second attempt will then check if the previous request did succeed at the MobilePay backend and base its reply upon the result of this check.



5.7 MobilePay amount exceeded

The MobilePay AppSwitch SDK returns error code no. 7 to the merchant app if the user's daily or yearly limits are exceeded by the amount requested in the order.

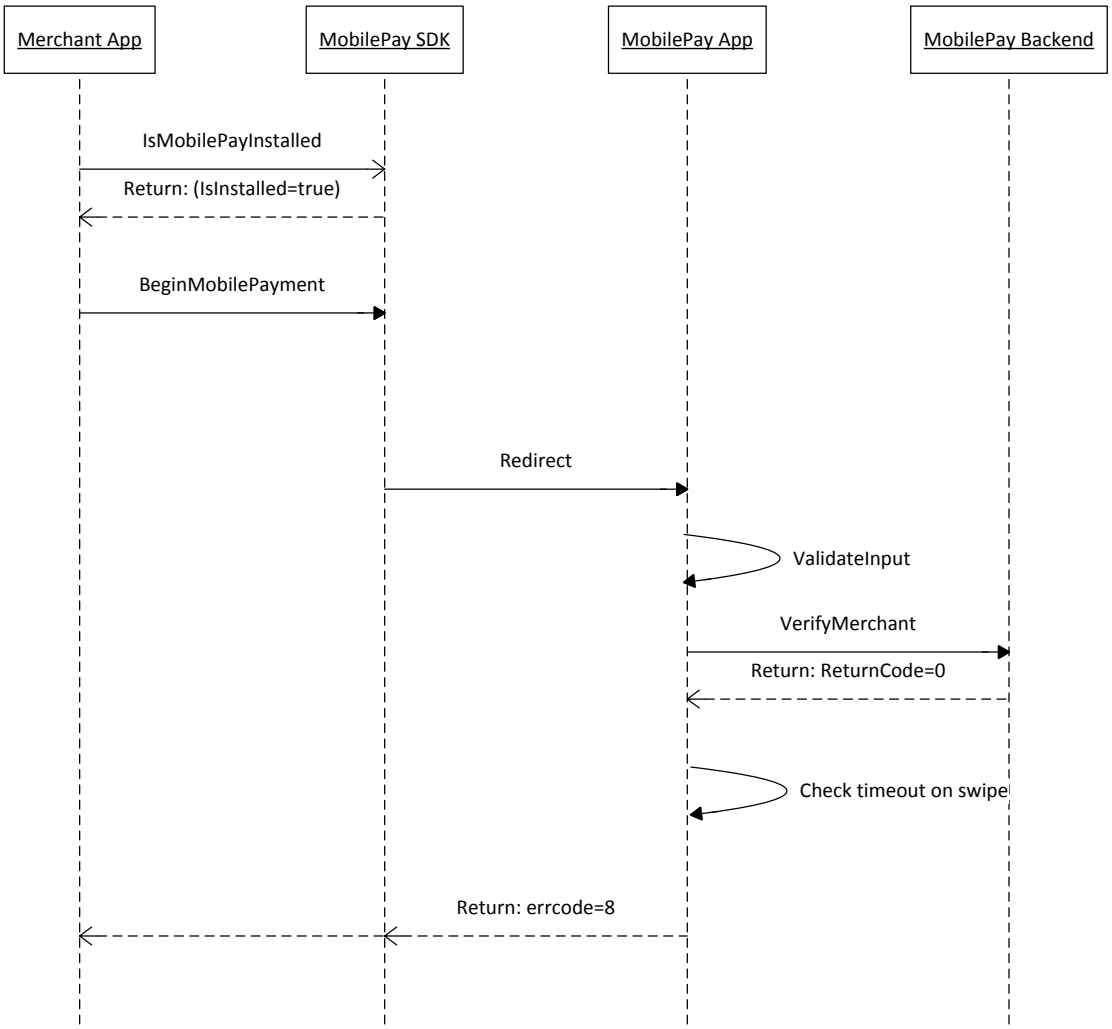
The merchant app should show a message informing the user that the limit has been exceeded and that he or she can view limits under "Beløbsgrænser" in the MobilePay app.



5.8 Timeout set in merchant app exceeded

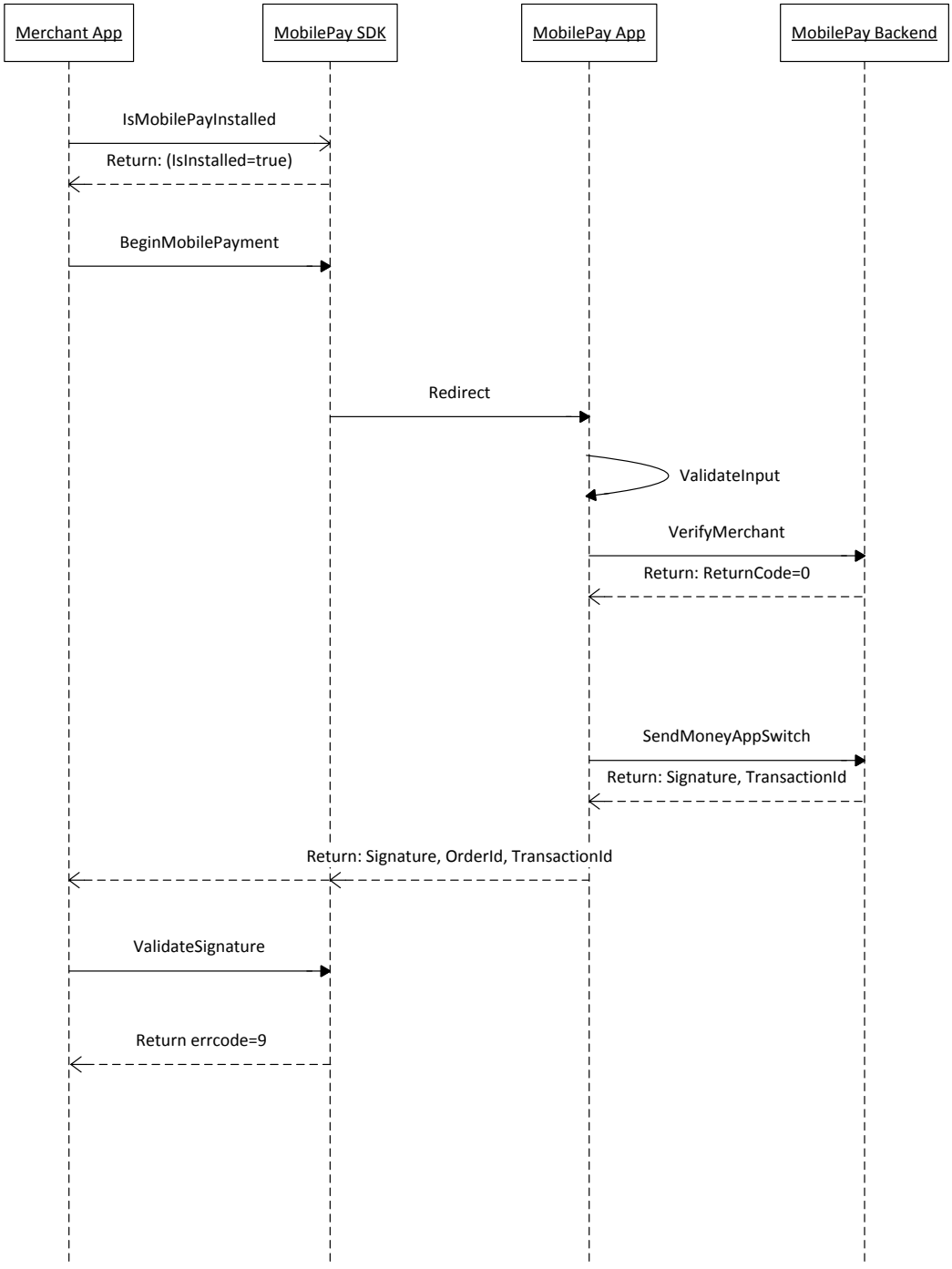
The MobilePay AppSwitch SDK will return error code no. 8 to the merchant app if the purchase takes longer than defined by the merchant app. The default timeout is set to 5 minutes. The timeout will be checked in the MobilePay app when confirming the payment.

The merchant app should show a message informing the user that he or she should try again before the timeout.



5.9 Invalid signature

The MobilePay AppSwitch SDK returns error code no. 9 to the merchant app if the SDK validation of the signature fails due to an invalid signature, if the same transaction ID is sent to the merchant app twice after a single payment.

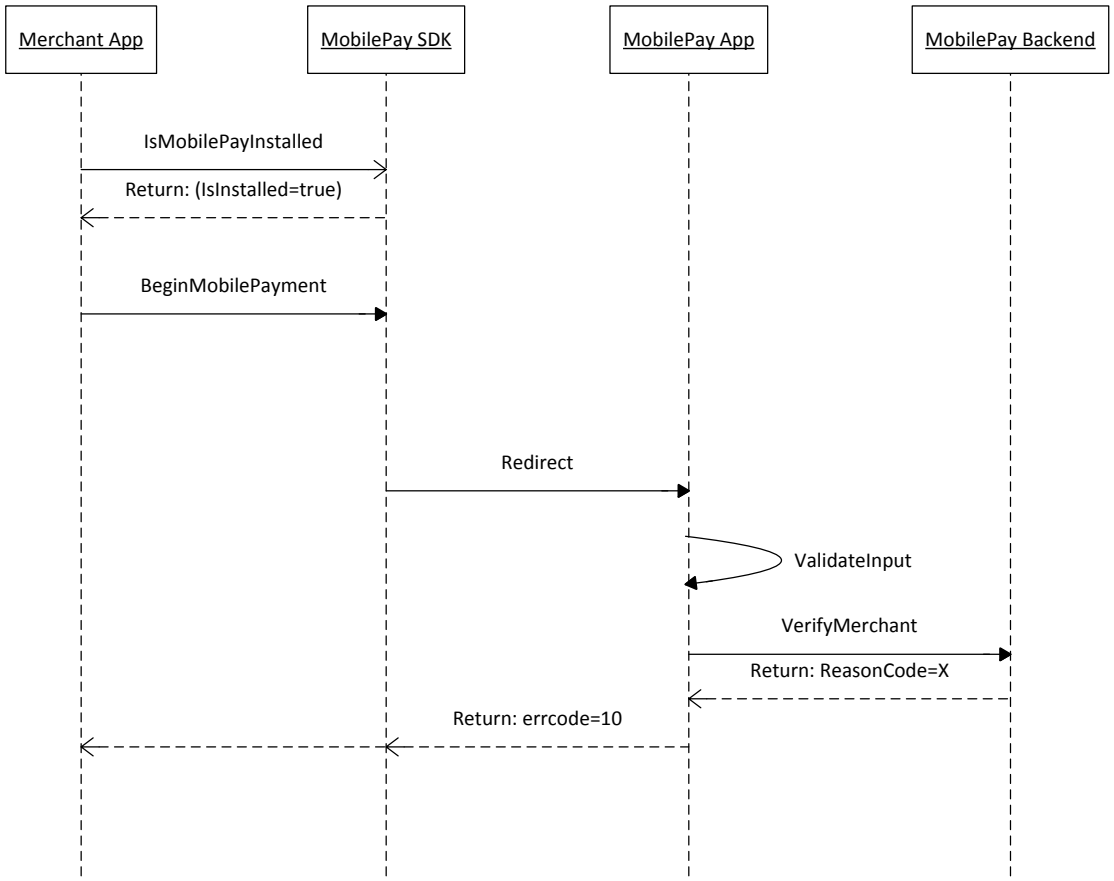


5.10 MobilePay AppSwitch SDK version is outdated

The MobilePay AppSwitch SDK will return error code no. 10 to the merchant app if the API version used by the SDK is declared obsolete by the MobilePay backend.

The merchant app receives this error code if the merchant app is not updated to the minimum version required by the MobilePay AppSwitch SDK.

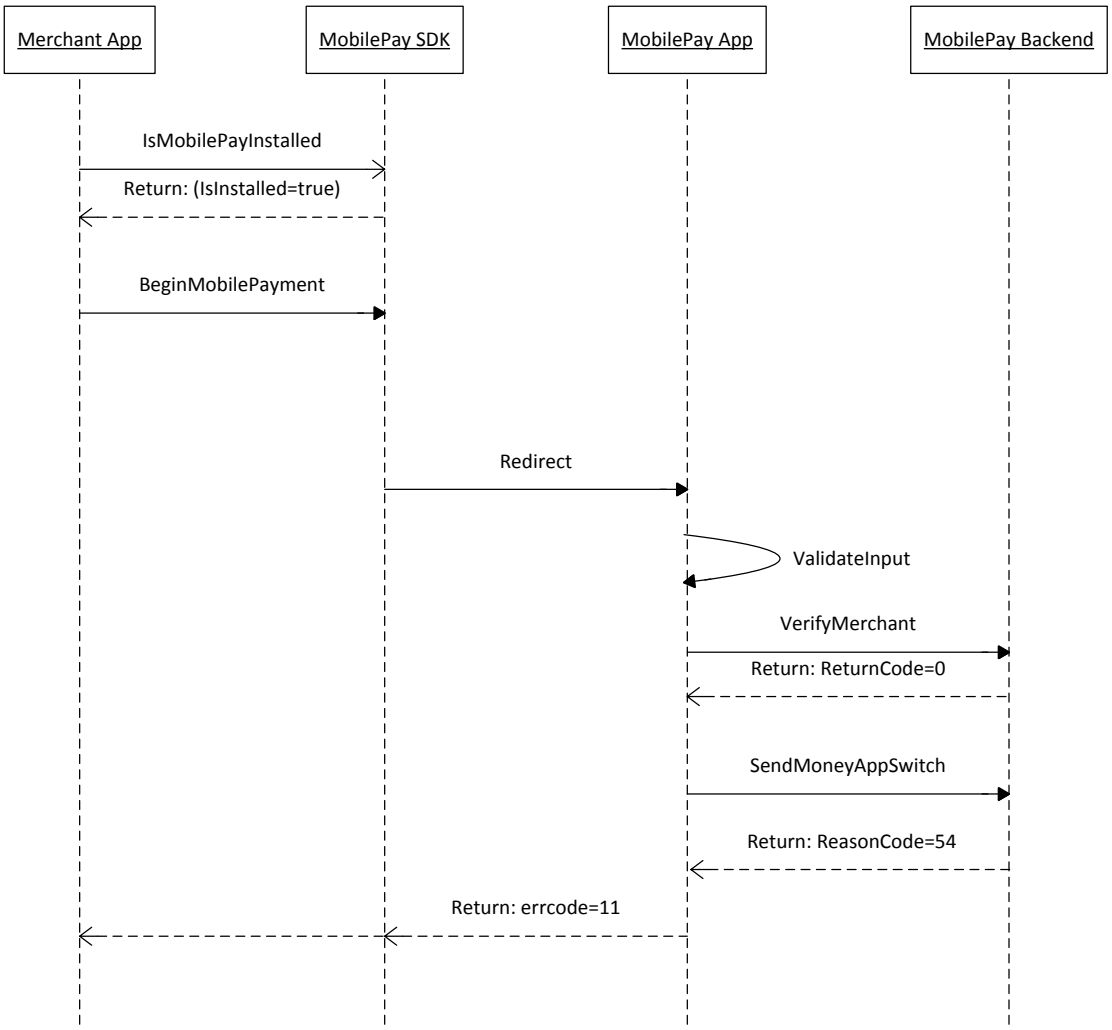
The merchant app should show a message asking the user to update the merchant app.



5.1.1 OrderID already used

Error code no. 11 is returned to the merchant app if the order ID sent to MobilePay already has been used for a confirmed payment by the same merchant but not the current customer.

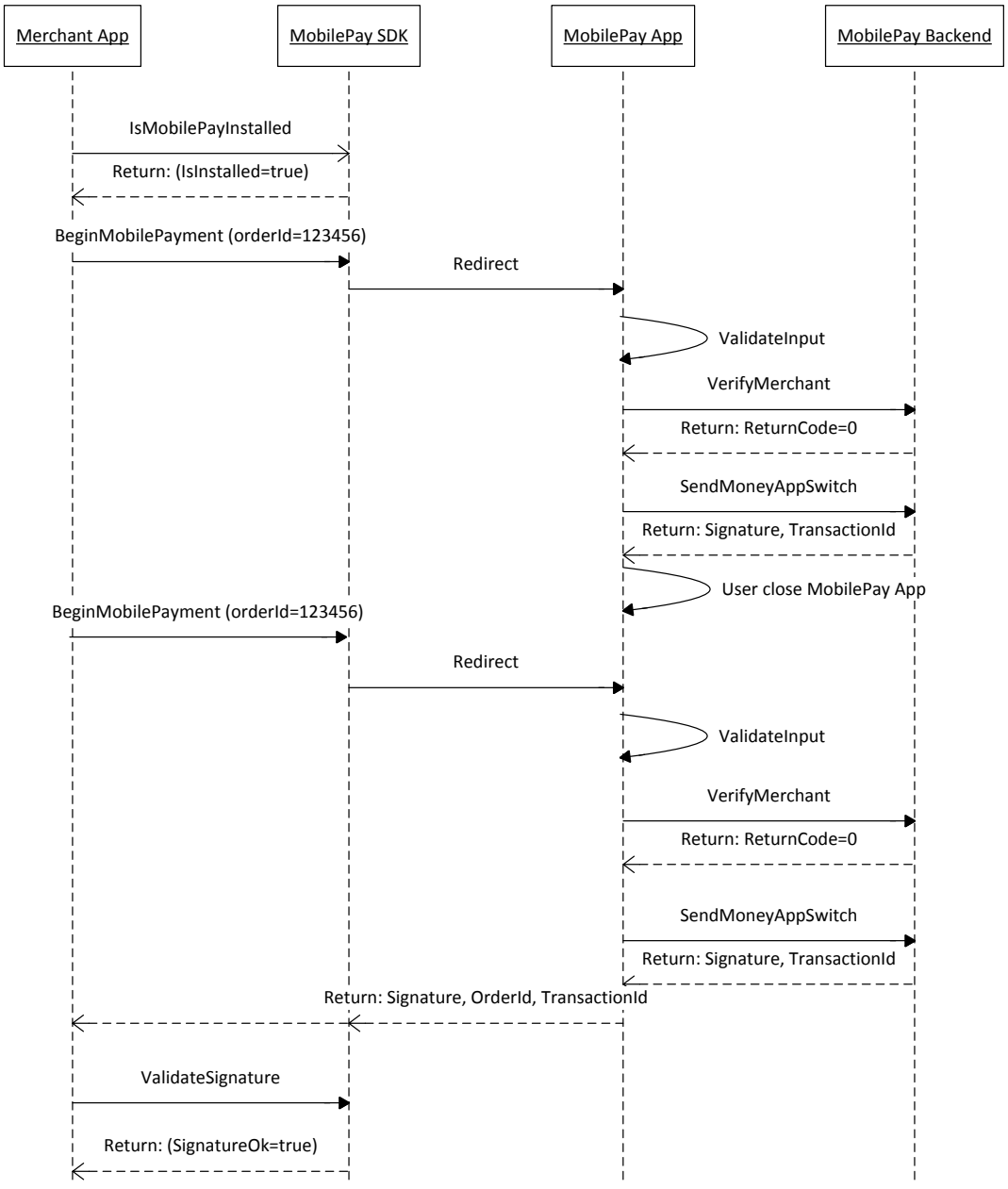
The merchant app must show a message to the user and create a new order ID in order to attempt making the payment again.



5.12 Abandoned payment scenarios - MobilePay app is closed down while doing payment

In this case, the merchant app does not receive a reply and this problem can be handled as a normal timeout scenario.

The merchant app can choose to resend the same order ID to MobilePay, which will ensure that the customer will not pay twice for the same order. MobilePay will always return the payment information, including the signature.

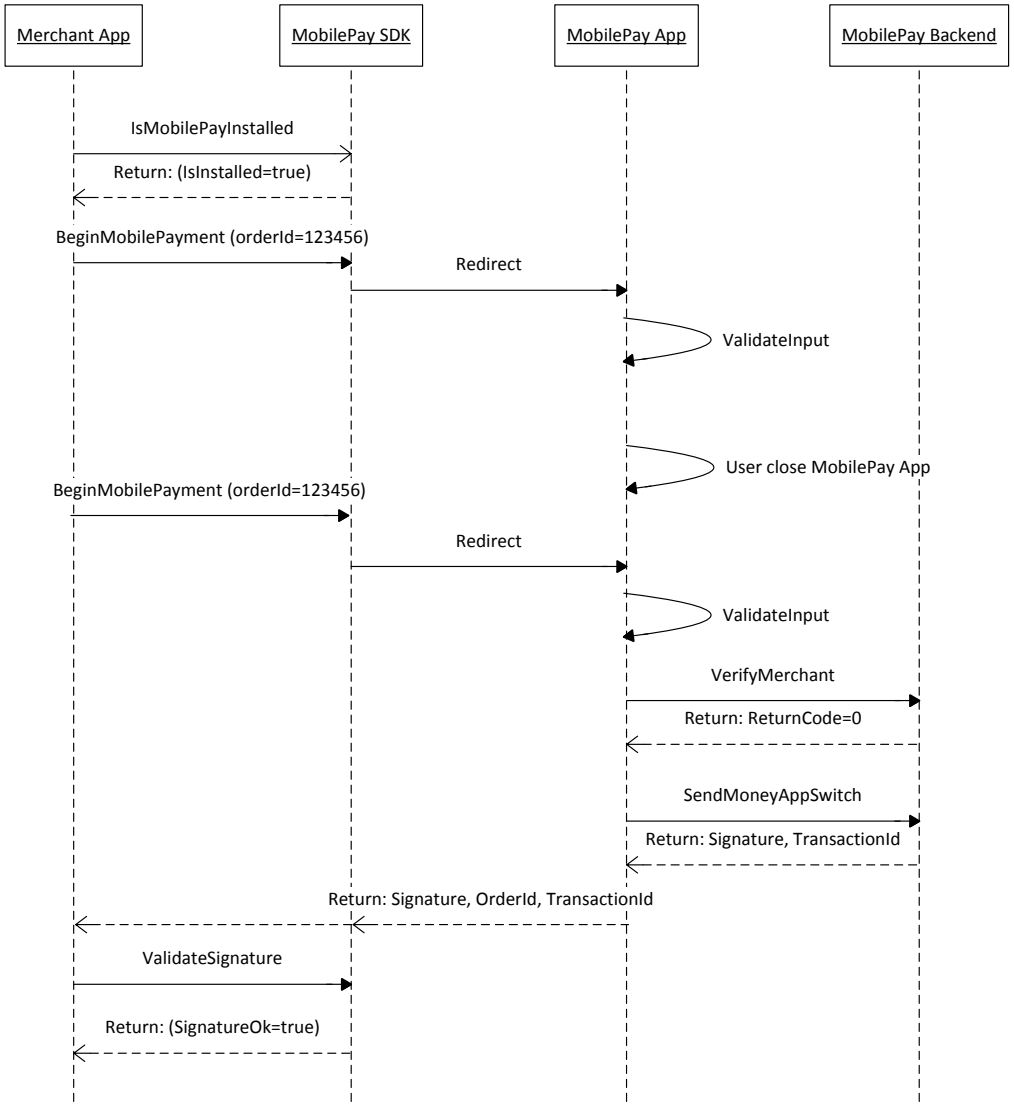


5.13 Abandoned payment scenarios - customer navigates away from MobilePay

The MobilePay app can oprate in two different modes, namely AppSwitch mode and normal payment mode. MobilePay is in AppSwitch mode when an AppSwitch payment is initiated and in normal payment mode when used e.g. in relation to P2P payments.

If the customer navigates away from MobilePay during an AppSwitch payment (e.g. by answering the phone) the payment flow is cancelled, the MobilePay app mode changes from AppSwitch mode to normal payment mode, and the user is logged out.

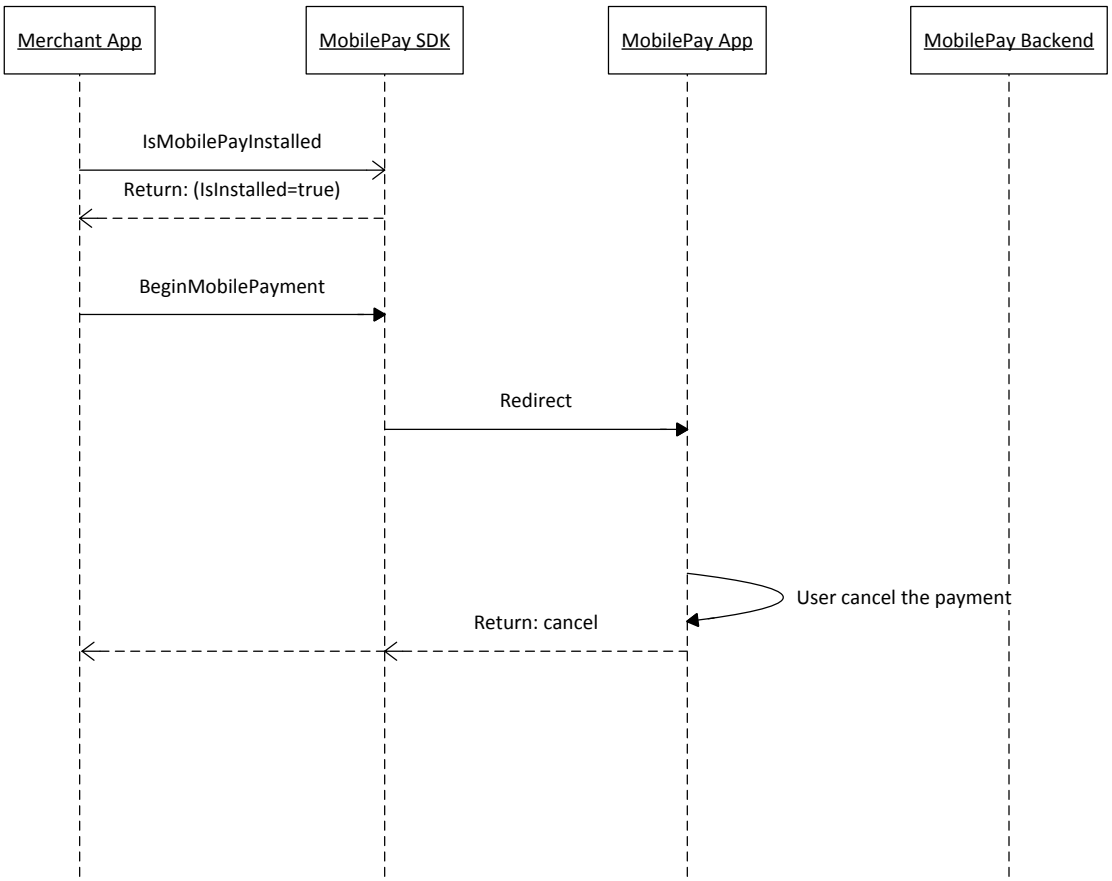
If the customer wants to pay after the phone call is completed, the customer has to restart the payment from the merchant app.



5.14 Abandoned payment scenarios - payment is cancelled in MobilePay

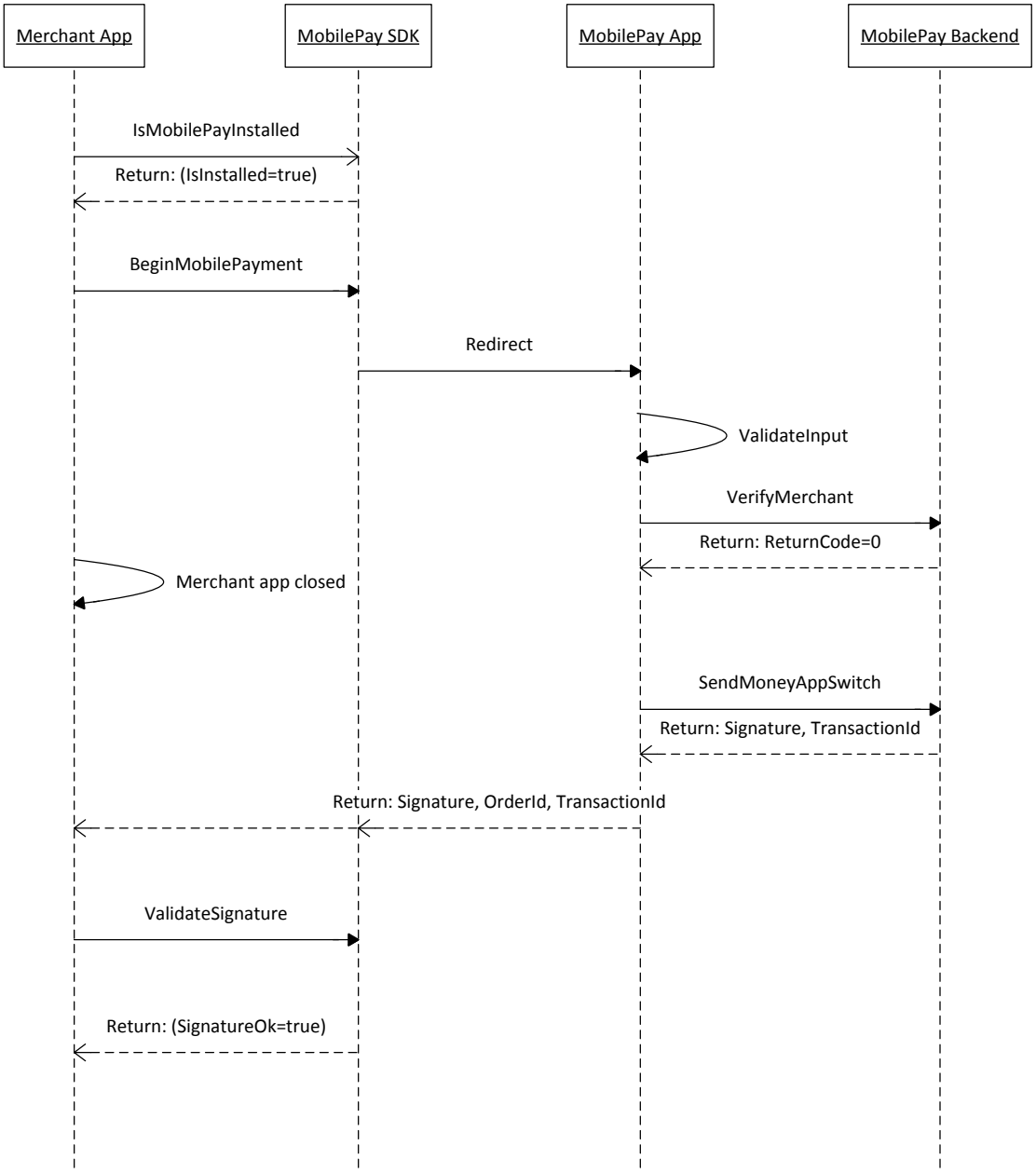
If the customer chooses to cancel the payment in the MobilePay app, the merchant app will be notified of the cancellation.

The merchant app can show a message to the user stating that the payment was cancelled and let the user continue shopping in the merchant app.



5.15 Abandoned payment scenarios - customer closes merchant app

If the customer closes the merchant app while making a payment in the MobilePay app (e.g. by using the multitask feature of the OS), the merchant app should also respond when receiving the call-back from MobilePay.



5.16 Abandoned payment scenarios - same order ID is sent to MobilePay twice

In case of network errors, slow network etc. the merchant app may resend the pending order (same order ID) to MobilePay, for which the customer has already paid for.

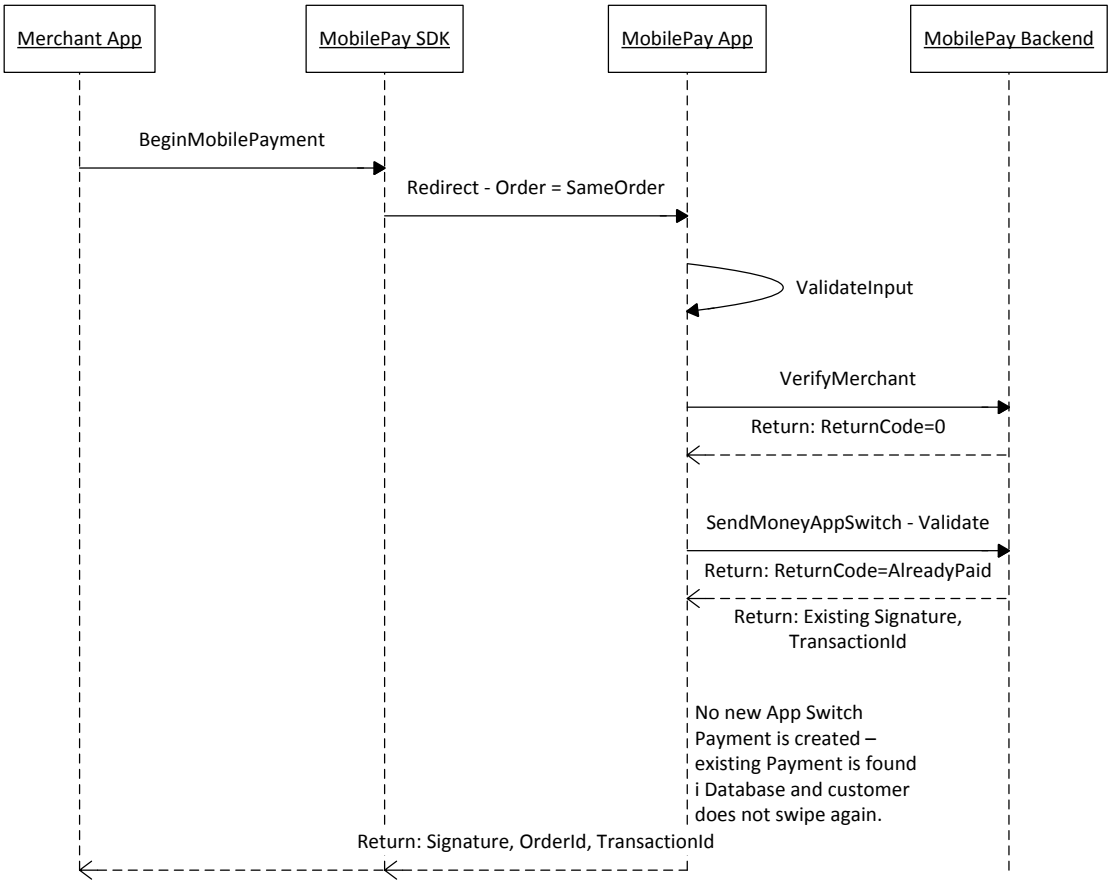
This error scenario is handled in MobilePay by returning the payment information to the merchant app and letting the merchant app complete the order.

MobilePay uses a unique ID for the payment, which consists of the merchant ID and the order ID. This ID is used to identify the payment transaction at DIBS, and DIBS will reject the ticket authorisation the second time if the same ID is used.

This means the customer will not be able to swipe for payment, but the MobilePay app will immediately show the receipt.

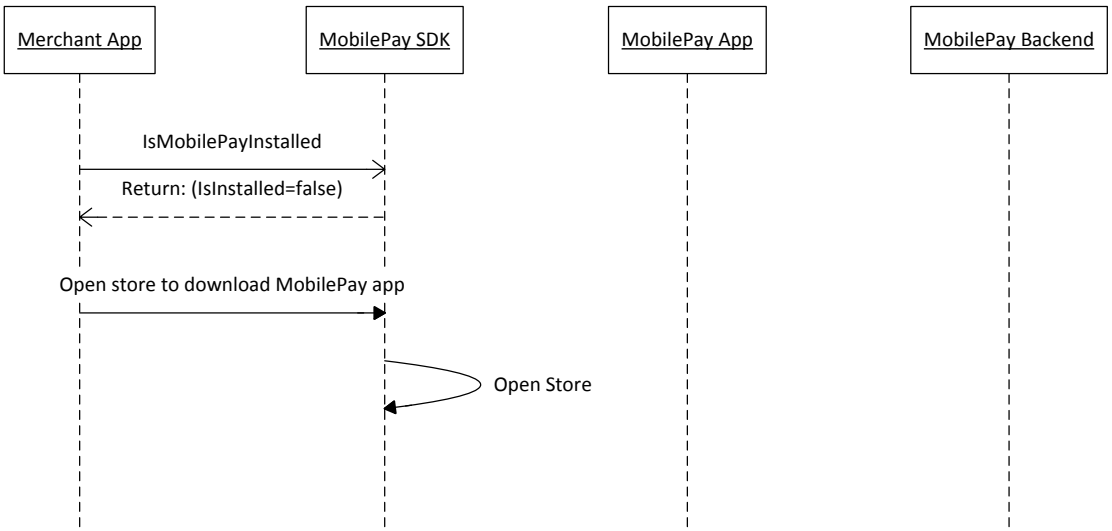
Please note that:

- Two redirects to MobilePay with the same order ID return signatures in both cases, but the customer has only paid once for this order. It must be ensured at merchant side that it will not be possible to receive two services or products using the same order ID.


Merchant App
MobilePay SDK
MobilePay App
MobilePay Backend
BeginMobilePayment
Redirect - Order = SameOrder
ValidateInput
VerifyMerchant
Return: ReturnCode=0
SendMoneyAppSwitch - Validate
Return: ReturnCode=AlreadyPaid
Return: Existing Signature, TransactionId
No new App Switch Payment is created – existing Payment is found i Database and customer does not swipe again.
Return: Signature, OrderId, TransactionId

5.17 Installation issues - MobilePay is not downloaded

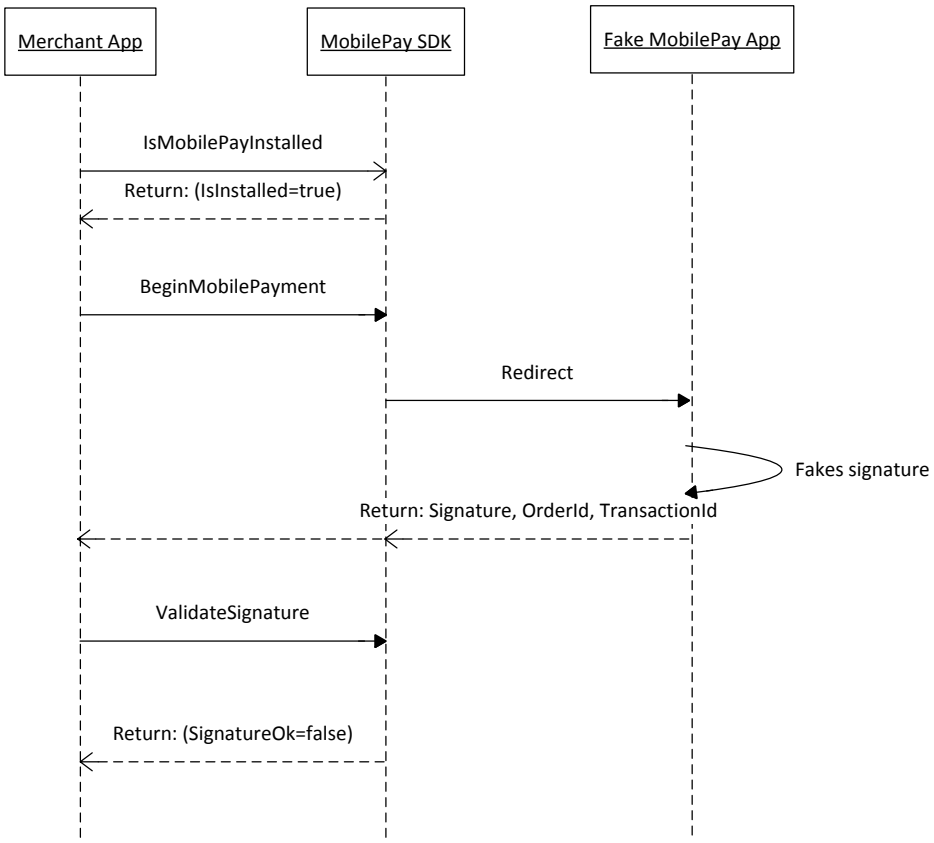
If MobilePay is not installed on the customer's phone, the merchant app can be setup to display an appropriate message. The SDK in GitHub has a method for checking whether MobilePay is installed or not.



5.18 Installation issues - fake MobilePay app installed

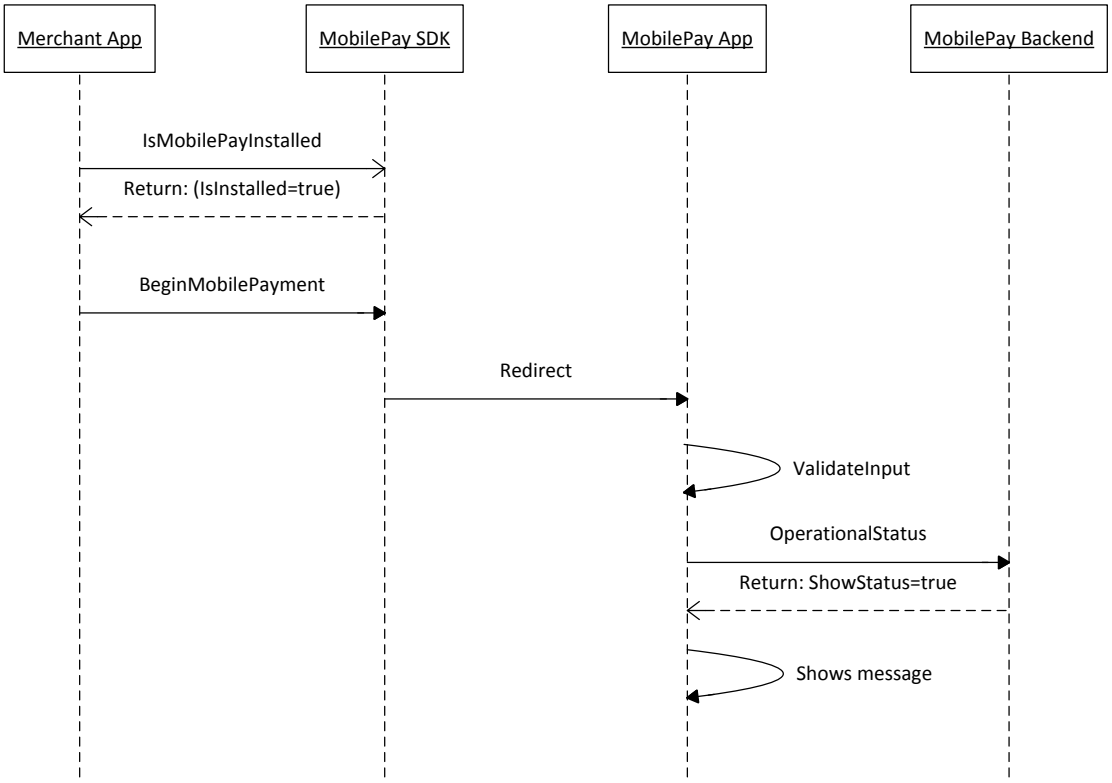
A fake MobilePay app will not be able to generate a valid signature.

Please also refer to section 4.4 “How to ensure authentication of payment”.



5.19 Installation issues - MobilePay is out of service

If MobilePay is out of service a notice is displayed in the MobilePay app. In this case the customer can choose to cancel the transaction in MobilePay and the flow will then continue as explained in 5.14.



6 Security

This section describes how the communication back and forth between the merchant app and the MobilePay app is secured.

Communication and security are both ensured by the MobilePay AppSwitch SDK on GitHub.

6.1 From merchant app to MobilePay app

The merchant app delivers the following data to the MobilePay app. Current version of the SDK:

Merchant ID	Char(60)
Order ID	Char(50)
Product name	Char(40)
Product price	Decimal
Receipt message	Char(66)
SDK version	Char(20)
Signature version	Decimal
Success URL	Char(100)
Failure URL	Char(100)
Cancel URL	Char(100)
Capture (Y/N)	Char(1)
HMAC (data)	Char(64)

HMAC (data) is a SHA-256 hash value of all the preceding data. The MobilePay app asks the MobilePay backend to verify the HMAC on this data. The key for the HMAC calculation is stored in the merchant app (MobilePay AppSwitch SDK part) and in Danske Bank Backend (HMAC key agreed upon between merchant and Danske Bank).

6.2 From MobilePay app to merchant app

If the Capture parameter specified in the input is 'Y', the MobilePay app will send the following data to the merchant app:

Order ID	Char(50)
Merchant ID	Char(60)
Transaction ID (Payment ID)	Char(20)
Signature	Char(2500)

And the signature will contain these data:

Signature version 2.0	
Order ID	Char(50)
Merchant ID	Char(60)

Transaction ID (Payment ID)	Char(20)
Amount	Dec(15,2)
Currency	Char(3)
Country	Char(2)

6.3 Security from MobilePay app to MobilePay backend at Danske Bank

The data sent from the MobilePay app to the MobilePay backend is sent over HTTP with SSL.

When a user logs on the MobilePay app the user will have a session with the MobilePay backend. This session is created using the AES and RSA cryptographic encryption algorithms.

6.4 Data at Rest

The input that the MobilePay app receives from the merchant app is stored temporarily, i.e. no data persists in the MobilePay app.

All payment and customer data is stored in the Danske Bank MobilePay backend (DB2 database).

MobilePay AppSwitch Implementation Guide

7 MobilePay AppSwitch SDK updates

The SDK updates according to the following scheme: {MAJOR}.{MINOR}.{PATCH}. Within a given version number category, each number is assigned in ascending order which means that version 1.4.3 is newer than 1.4.2.

The {MAJOR} number is increased when there are significant jumps in functionality or changes to the framework which could cause incompatibility with interfacing systems.

The {MINOR} number is increased when minor features or significant fixes have been added.

The {PATCH} number is increased when minor bugs are fixed.

8 Test setup

It is not possible for merchants to communicate with the Danske Bank test environment. Testing must therefore be done in the production environment.

The test merchant ID "APPDK0000000000" can be used for testing purposes. When the test merchant ID is used, it is possible to complete the payment flow without transferring any money. This means the merchant is able to test the security setup, SDK etc. without creating any payments.

When the merchant wants to test reconciliation files, the merchant must use the production merchant ID in order to create real payments. However, small amounts of e.g. 0.01 DKK can be used in this test setup.

9 Key Terms & Definitions

Terms	Definitions
AES	Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S
Alias	See Merchant ID
DIBS	Dansk Internet Betalings System
Data at Rest	Data at Rest is used as a complement to the terms Data in Use and Data in Motion which together define the three states of digital Data.
GitHub	GitHub - Code sharing service. It is a Git repository web-based hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as own features.
HMAC	Hash Message Authentication Code is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key.
HMAC Message	The message on which the HMAC calculation is based
HMAC Key	Agreed upon (between sender and receiver) key used for HMAC calculation
HTTP	Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web
Merchant App	Synonym for an app that involves payment transactions - typically related to selling of goods or services.
Merchant ID	Merchant identification number - A unique merchant ID provided by Danske Bank. Also referred to as AppSwitch ID.
Order ID	Order identification number - here referring to the unique provided Order ID (shop's order ID) sent along with a payment request from a merchant (app) to MobilePay (app)
Payment ID	See Transaction ID
P2P	Peer-to-peer payment
REST	Representational state transfer (REST) is a simple stateless architecture that generally runs over HTTP. REST is used between the MobilePay app and the MobilePay backend.
RSA	RSA is one of the first practicable public-key cryptosystems and is widely used for secure data transmission. RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key.
SDK	Software Development Kit – MobilePay AppSwitch SDK is designed to be embedded in a merchant app.
SIM	Subscriber Identification Module
SSL	Secure Sockets Layer – a standard cryptographic protocol designed to provide communication security over the Internet.
Transaction ID	DIBS provided payment transaction ID
UI	User Interface