TECHNICAL SPECIFICATIONS

Redirection

ePOS Paycenter
powered by Piraeus Bank

PIRAEUS BANK

T. +30 210 38 98 954, www.piraeusbank.gr

Change History

Date	Version	Modifications
01/05/2009	1.0	Original version
22/05/2009	1.0.1	▪ Addition of «**AcquirerId**», «**RequestType**», «**CurrencyCode**», «**ExpirePreauth**» and «**parameters**» to ticketing Web Service ▪ Elimination of the above parameters from the form redirecting the user to the payment page ▪ Elimination of the «**RequestType**» parameter from the response sent to the success/failure pages when the «POST» method is chosen ▪ Addition of the «**PackageNo**» and «**AuthStatus**» parameters to the response sent to the success/failure pages when the «GET» method is chosen
01/08/2009	1.0.2	Addition of the option to execute asynchronous transactions (Modification in section 5, «StatusFlag» parameter, and section 9)
23/10/2009	1.0.3	▪ Change of URL where the form is submitted in case of test transactions (section 5) ▪ Change of card number in test cases 8 and 9 (section 7)
01/02/2010	1.0.4	Change of ticketing Web Service URL (section 4) and URL that the form is submitted via POST method (section 5)
21/01/2013	1.0.5	▪ Support of Maestro and American Express cards ▪ Support of Russian language in payment page ▪ Support of various currencies apart from euro
05/08/2013	1.0.6	▪ Support of German language in payment page ▪ Addition of new test case for transactions in USD currency (test case 15 in section 7)
28/04/2014	1.0.7	▪ Support of Discover cards ▪ Addition of new test case for transactions with Discover card (test case 13 in section 7)
22/01/2016	1.0.8	Addition of new currencies and new logo
07/06/2016	1.0.9	Addition of new parameter «**PaymentMethod**»
01/06/2017	1.1.0	New Mastercard/Maestro logos
13/11/2018	1.1.1	New **HashKey** encryption algorithm (HMACSHA256)
02/09/2019	1.1.2	▪ Section 4: Addition of new parameters to Ticketing Web Service: BillAddrCity, BillAddrCountry, BillAddrLine1, BillAddrLine2, BillAddrLine3, BillAddrPostCode, BillAddrState, ShipAddrCity, ShipAddrCountry, ShipAddrLine1, ShipAddrLine2, ShipAddrLine3, ShipAddrPostCode, ShipAddrState, CardholderName, Email, HomePhone, MobilePhone, WorkPhone, RecurringInd, RecurPurchaseDate, RecurFreq, RecurEnd ▪ Section 5 - Response receipt via method POST & GET: Change of the value of AuthStatus and addition of parameter TraceID
20/07/2020	1.1.3	▪ Section 4: Update of the parameters description to Ticketing Web Service: BillAddrCountry, BillAddrLine1, BillAddrPostCode, BillAddrState, ShipAddrCity, ShipAddrCountry, ShipAddrLine1, ShipAddrPostCode, ShipAddrState, CardholderName, Email, HomePhone, MobilePhone, WorkPhone

Contents

1. Introduction

The «*Redirection*» solution of the Piraeus Bank «*ePOS Paycenter*» service is used to process electronic payment transactions, i.e. card debit requests coming from business sites hosting e-shops, where users can pay using their cards.

By means of this solution the user is redirected from the respective merchant site to a Piraeus Bank secure page where they fill out their card details. For the user to be directed to this page, an html form using the «POST» method and hidden parameters for various transaction-specific information is applied. To implement the merchant application, any programming language may be used.

The user enters his/her card details on a secure page (SSL encryption with a 2048-bit key size) which ensures secure transfer of the card details to «ePOS Paycenter».

The information to be filled out on the payment page is as follows:

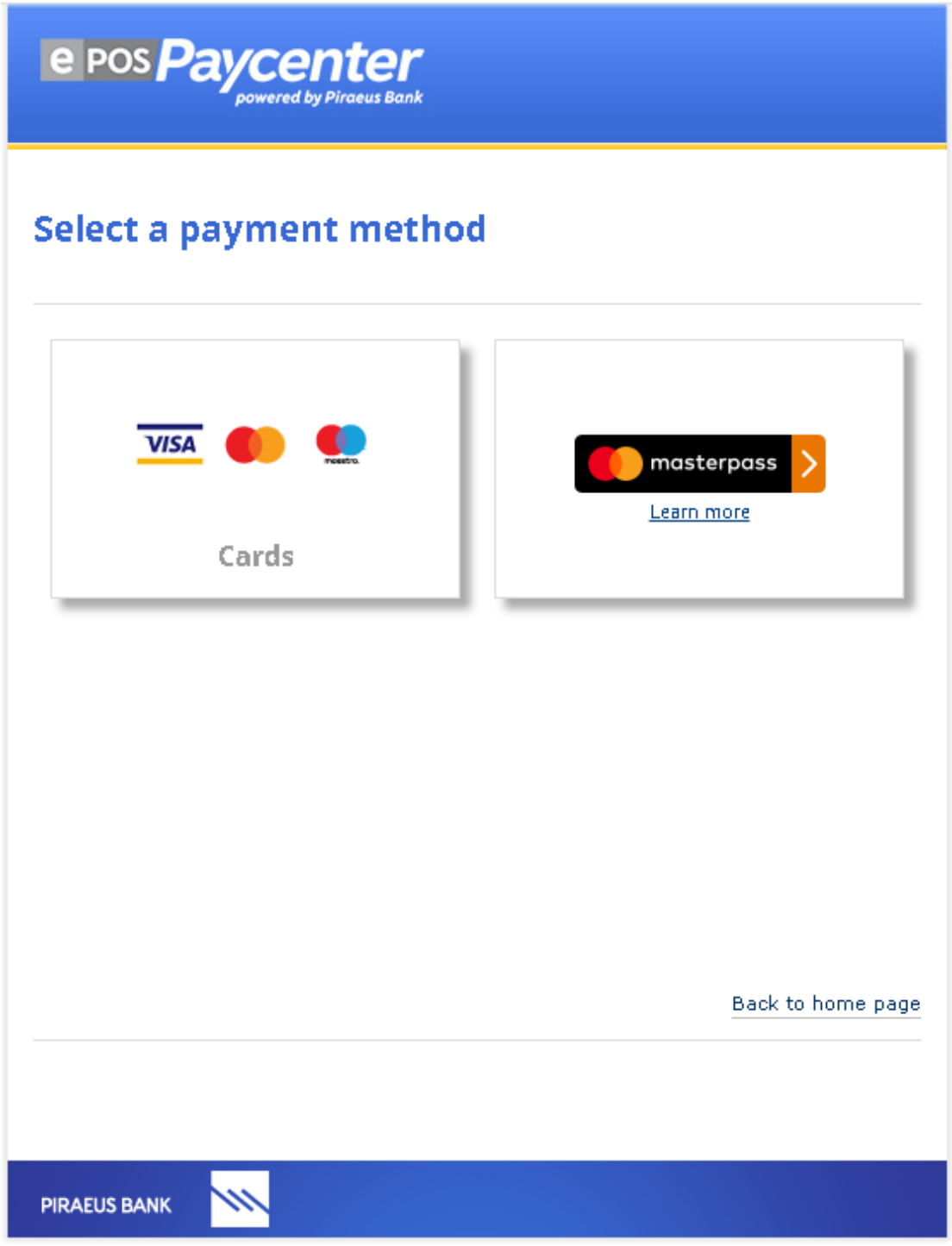
- Card number
- Expiry date
- Card verification code (CVV2/CVC2)
- Email to which a successful card charge confirmation can be sent to the user

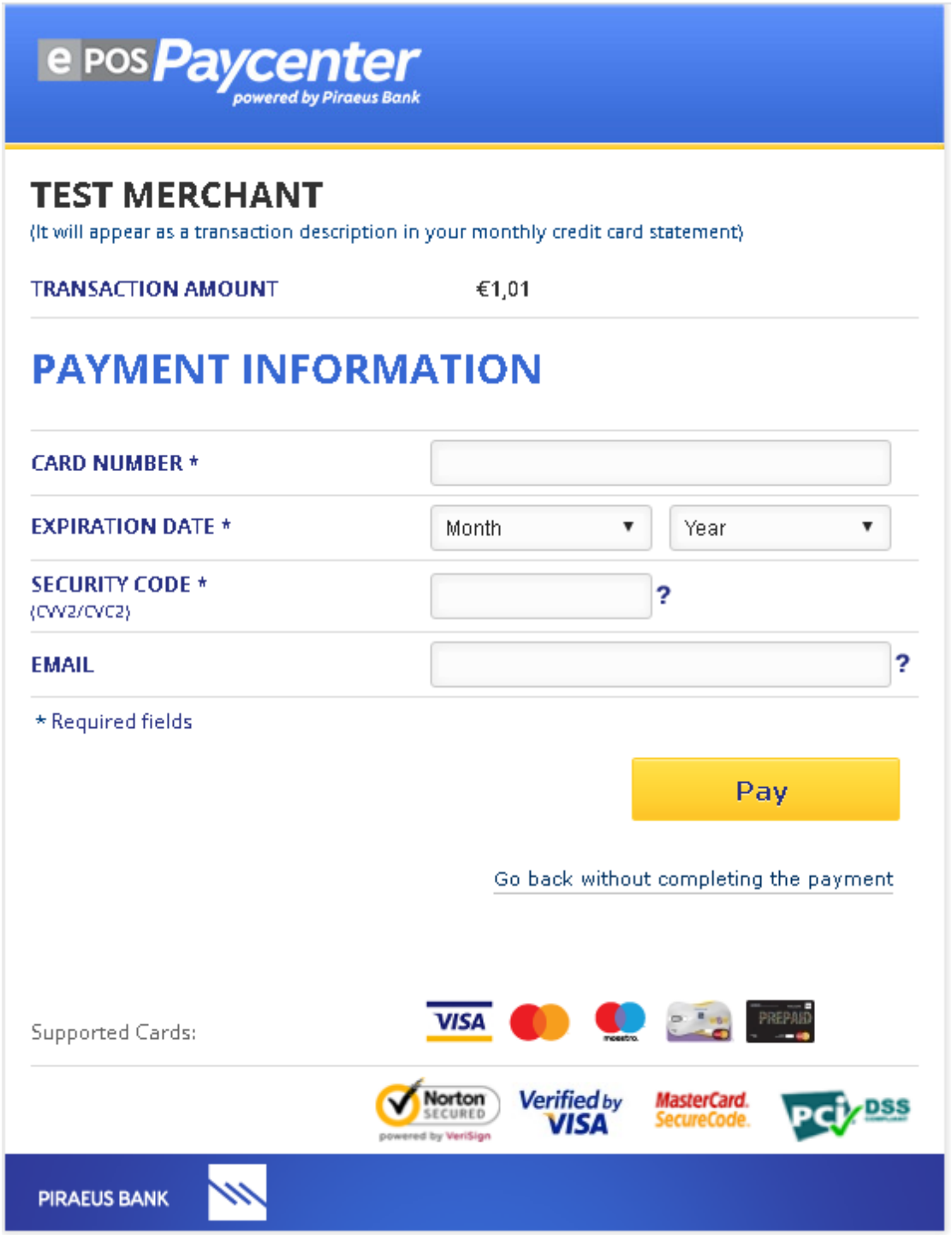
The entry of card details is followed by the process of strong customer authentication (3D Secure), which enhances transaction security level. Once 3D Secure process is completed, card debit transaction is processed and response-specific information is sent to predefined merchant site URLs. Additionally, Piraeus Bank may send a notification email automatically to the merchant, upon request, indicating the transactions performed on its site.

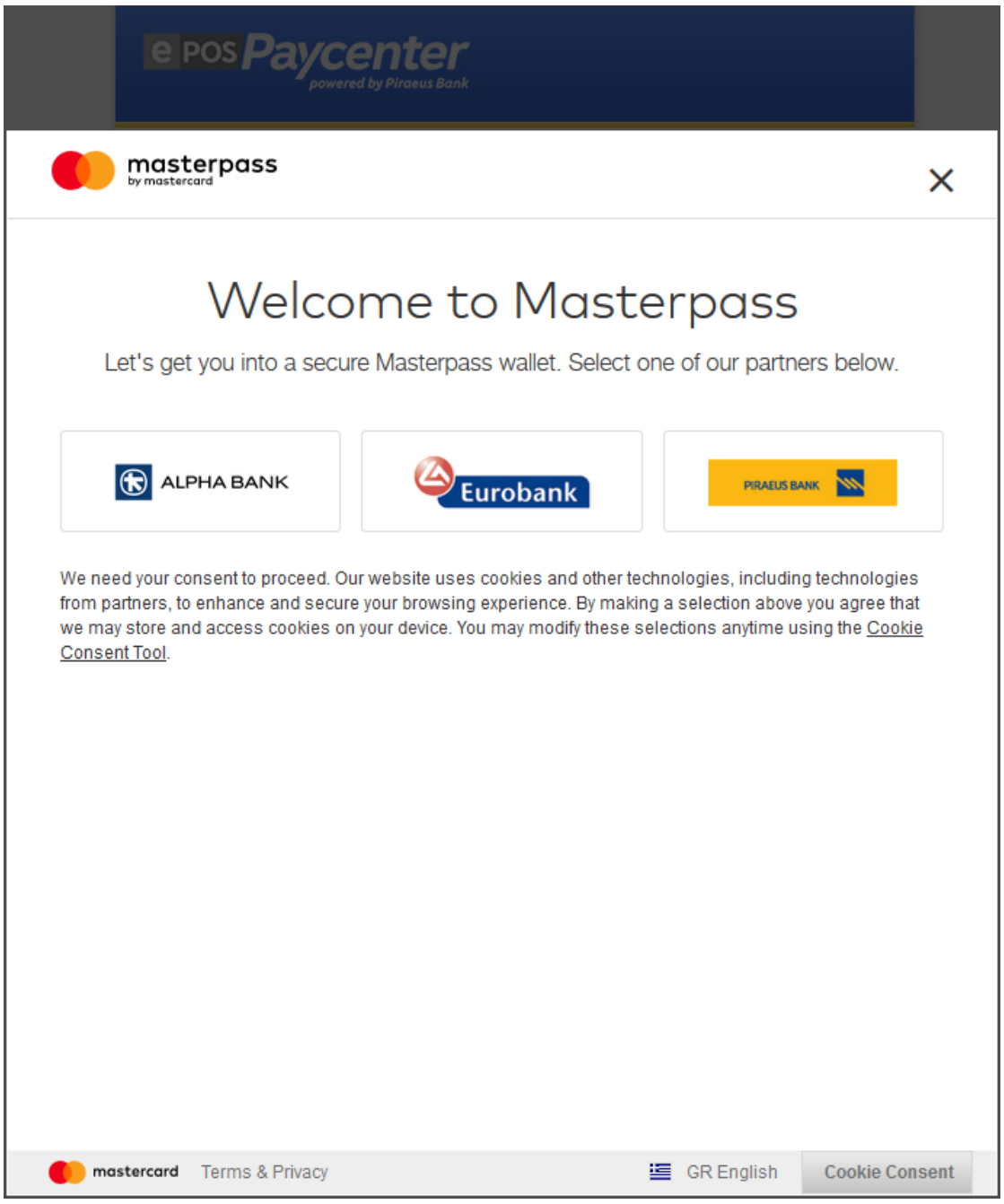
The «Redirection» solution provides the following payment methods:

- Direct entry of card data
- Usage of the digital Masterpass Wallet (see relevant specifications "Masterpass Wallet Acceptance")

Here are samples of the relevant pages:


ePOS Paycenter
powered by Piraeus Bank
Select a payment method
VISA
Cards
masterpass
Learn more
Back to home page
PIRAEUS BANK


ePOS Paycenter
powered by Piraeus Bank
TEST MERCHANT
(It will appear as a transaction description in your monthly credit card statement)
TRANSACTION AMOUNT
€1,01
PAYMENT INFORMATION
CARD NUMBER *
EXPIRATION DATE *
Month
Year
SECURITY CODE *
(CVV2/CVC2)
EMAIL
* Required fields
Pay
Go back without completing the payment
Supported Cards:
VISA
PREPAID
Norton SECURED
powered by VeriSign
Verified by VISA
MasterCard. SecureCode.
PCI DSS
PIRAEUS BANK



Eligible cards are as follows:

- Visa and Mastercard credit cards issued by any Bank
- Visa and Mastercard debit cards issued by any Bank
- Maestro debit cards
- Visa and Mastercard prepaid cards issued by any Bank

Besides, if Diners/Discover or American Express cards are included in the collaboration with a merchant, then they are also eligible.

Attention!

To support Diners/Discover or American Express cards, the merchant should first contact Piraeus Bank in order to be informed about the necessary business process.

In the sections below detailed information is provided on the following:

- **Section 2 → General Architecture:**
  Outline of the «Redirection» solution overall structure.

- **Section 3 → Details for the Creation of a Test Account:**
  The details required to be submitted to Piraeus Bank so as to create a *test account* to perform test transactions.

- **Section 4 → Ticketing Mechanism:**
  Description of the «Ticketing Mechanism», a process that must precede any transaction in order to state the details of the transaction to be executed (e.g. transaction type, amount, number of installments, etc.).

- **Section 5 → Data Submission – Transaction Response Receipt:**
  Description of the parameters sent via the html form to redirect the user to the payment page as well as parameters included in the response to the merchant site.

- **Section 6 → Merchant Application Action Flow:**
  Chart illustration of the algorithm to be implemented by the merchant application so that a transaction may be executed.

- **Section 7 → Test Cases:**
  Description of the test cases to be performed in the framework of test transactions.

- **Section 8 → Use of Icons:**
  Material about the mandatory and optional icons to be posted on the merchant site.

- **Section 9 → Tips:**
  Tips and remarks about key points to consider.

- **Section 10 → Implementation Checklist:**
  A list of actions to be performed by the Technical Manager, in order to conclude the agreement with the merchant.

2. General Architecture

The chart below illustrates the general architecture of the «Redirection» service.

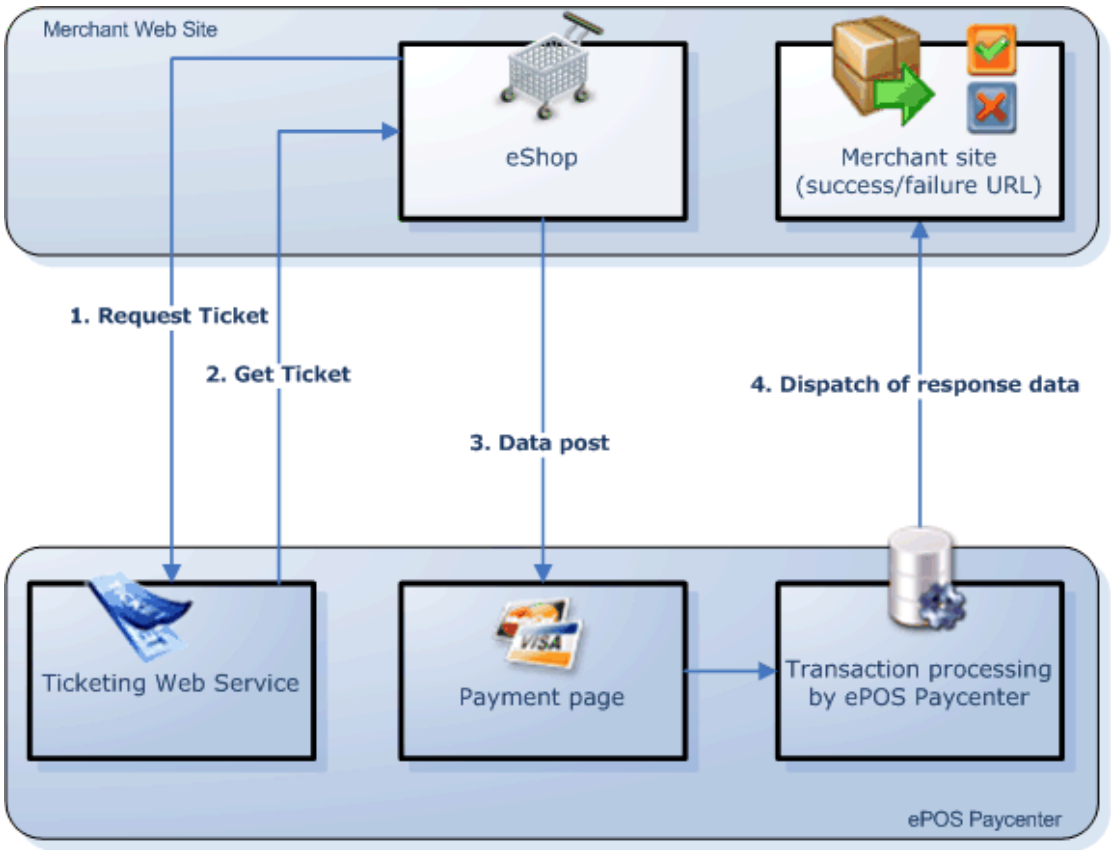


Chart 1: *General Architecture*

For any transaction to be executed, the «*Ticketing Mechanism*» is first used to send transaction-specific information (e.g. transaction type, amount, number of installments, etc.) from the merchant application. Then a «*Transaction ticket*» is received, which is necessary to authenticate the response to be received by the site.

In particular, the merchant system calls a SOAP Web Service (step 1) to communicate to the Piraeus system («*ePOS Paycenter*») data about the transaction to be executed and acquire a «*transaction ticket*» (step 2) which will be required to authenticate the response to be received from the Paycenter. This mechanism is detailed in section 4 «*Ticketing Mechanism*». Then, an HTML form having various information as hidden parameters is used and submitted to a specified URL with the «POST» method (step 3) – see section 5.

The user is directed to the Piraeus Bank payment page, fills out his/her card details and the transaction is processed by the Paycenter. Finally, a response is sent to a predefined URL of the merchant site (step 4) – see section 5.

3. Details for the Creation of a Test Account

The data to be submitted to Piraeus Bank in order to provide the necessary technical info (*test account*) for test transactions are as follows (all mandatory):

- **Details of the Technical Manager**
  - Name of the technical manager
  - Telephone of the technical manager
  - Email address of the technical manager
  - Company where the technical manager is employed

- **Details of the merchant owning the site:**
  - Distinctive title of the merchant owning the site
  - Tax Registration Number of the merchant owning the site
  - Domain name of the merchant live site

- **Technical data:**
  - **Website URL**: The website URL from where the test transactions will be run.
  - **Referrer URL**: The page URL from which test transaction data will be sent.
  - **Success URL**: The page URL to which the test transaction success response will be sent.
  - **Failure URL**: The page URL to which the test transaction failure response will be sent.
  - **Backlink URL**: The URL to which the user is returned when the «Cancel» button is pressed.
  - **IP address**: The IP address of the server sending the requests to the ticketing mechanism – see section 4.
  - **Response method (GET or POST)**: It concerns the way in which responses will be sent to the success/failure URLs – see section 5.
  - **Installments support (YES/NO)**: Declare if you are going to use installments in test transactions.
  - **Payment method selection page** (see relevant specifications "Masterpass Wallet Acceptance")**:**
    - Show
    - Don't show

The *test account* details provided by Piraeus Bank, once the above information is sent, are as follows:

- AcquirerId
- MerchantId
- PosId
- Username
- Password

Information about the use of these details is provided in the following sections.

4. Ticketing Mechanism

The «*Ticketing mechanism*» is used to report via a SOAP Web Service to the Paycenter the details of the transaction to be sent with a specific reference («*MerchantReference*») as well as to acquire a «*transaction ticket*» that will be required for the merchant system to authenticate the response received from the Paycenter.

In particular, via the «*ticketing Web Service*»:

- The merchant system reports the details of the transaction to be sent with a specific reference («***MerchantReference***»).
- A «*transaction ticket*» is received to be used when the transaction is executed in order for the merchant site to authenticate the response received from the Paycenter (for detailed information, see section 5 - *«Hash Key» Verification*).

Note:

The reference («***MerchantReference***») is a code generated by the merchant site uniquely identifying a transaction (e.g. order number, contract number etc.).

Attention!

- The «Ticketing mechanism» must precede **each transaction**.
- The details reported by the «*ticketing Web Service*» have a defined validity time (30 min.) within which the transaction has to be submitted.
- If the ticketing Web Service is called more than once using the same «MerchantReference» (e.g. due to transaction failure), a different «transaction ticket» will be returned each time. The details of the last call are always the ones in effect.
- The Web Service call must be made through Server. **Cross-origin HTTP requests are not allowed through scripts**.

The «*ticketing Web Service*» URL is:

https://paycenter.piraeusbank.gr/services/tickets/issuer.asmx

The following tables provide description for the request and response parameters of Ticketing Web Service. Each parameter has one of the following indications:

- **M (Mandatory)**: The parameter must have a value
- **O (Optional)**: A parameter value is not required.

Below there is a list of the information required to call the «*ticketing Web Service*»:

REQUEST PARAMETERS		
Parameter name	**Description**	**Type**
AcquirerId (M)	The acquirer identification. Value will be provided by Piraeus Bank.	Integer
MerchantId (M)	The merchant identification number. Value will be provided by Piraeus Bank.	Integer
PosId (M)	The POS identification. Provided by Piraeus Bank.	Integer
Username (M)	The user name. Value will be provided by Piraeus Bank.	String (max. 50 characters)
Password (M)	The user password encrypted with the MD5 hashing algorithm. Value will be provided by Piraeus Bank (in non-encrypted form).	String (max. 50 characters)
RequestType (M)	The transaction type to be executed. Possible values: ▪ **00**: Preauthorization → The amount will be simply committed and, later the preauthorization will have to be completed (through either the AdminTool or a Web Service call with RequestType = «SETTLE» [*]) so as to be settled. **Attention!** Preauthorization is available after Piraeus Bank's approval. Please contact Piraeus Bank for activating this type of transaction. **Note:** [*] In case of interest in preauthorization completion though a Web Service call, the technical specifications should be requested from Piraeus Bank. ▪ **02**: Sale → Transaction to be immediately settled in the current batch.	String (2 characters)

CurrencyCode (M)	The code of the transaction currency. It is **978** for debits in euro. **Attention!** For every different currency, a new MerchantId and PosId value will be provided by Piraeus Bank. **Note:** The supported currency codes are shown in **Annex 5**.	Integer
MerchantReference (M)	The reference of the transaction to be executed using the details designated in the other parameters. ▪ «MerchantReference» contains only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+- ▪ It has to be unique to each successful transaction. **Attention!** ▪ This value must be the same as that of the «*MerchantReference*» field in the form redirecting the user to the payment page. ▪ If a sale/preauthorisation transaction is not successful, the transaction may be re-submitted with the same «MerchantReference». ▪ When a sale/preauthorisation transaction has been approved, even if it is refunded, it is impossible to re-use the «MerchantReference» of that transaction in any future transaction (see section 6).	String (max. 50 characters)
Amount (M)	The amount of the transaction with reference «*MerchantReference*». **Attention!** ▪ When the transaction contains installments the parameter includes **total amount**. ▪ The value must be stored in order to be known to the merchant system	Decimal with 2 decimal digits
Installments (M)	The number of installments of the transaction with the «*MerchantReference*». ▪ To support installments, the merchant	Unsigned Byte

	must state it to Piraeus Bank. ▪ For non-installment transactions, the value should be 0 or 1. **Attention!** In the event of installments, their number must be stored in order to be known to the merchant system. **Note:** Piraeus Bank provides the «BIN Web Service» which can be used in order to check if a card supports installments without sending a charge transaction. In case of interest, the technical specifications should be requested from Piraeus Bank.	
ExpirePreauth (M)	It concerns preauthorization requests (RequestType =«00») and contains the number of days in which the preauthorization may be settled. Maximum value: 30 days. In the event of sale (RequestType =«02»), the value should be 0. **Note:** If preauthorizations are used, the «**ExpirePreauth**» parameter must take a value higher than 1.	Integer
Bnpl (M)	For future use (the value is always 0).	Unsigned Byte
Parameters (M)	The content of «parameters» variable is returned to the success/failure pages. It may be used to pass various information items that will be returned to the site along with the other parameters.	String (max. 512 characters)
BillAddrCity (M)	Billing address city (For 3D Secure process purposes) The parameter contains only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-	String (max. 50 characters)
BillAddrCountry (M)	ISO 3166-1 numeric country code, corresponding to Billing address country. E.g. 300 for Greece. (For 3D Secure process purposes)	String (max. 3 characters)
BillAddrLine1 (M)	Additional line 1 of the billing address (For 3D Secure process purposes) The parameter contains only Greek or Latin lowercase & uppercase alphanumeric	String (max. 50 characters)

	characters, space, or the following special characters /:_().,+-	
BillAddrLine2 (O)	Additional line 2 of the billing address (For 3D Secure process purposes) The parameter contains only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-	String (max. 50 characters)
BillAddrLine3 (O)	Additional line 3 of the billing address (For 3D Secure process purposes) The parameter contains only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-	String (max. 50 characters)
BillAddrPostCode (M)	Post code of the billing address (For 3D Secure process purposes)	String (max. 16 characters)
BillAddrState (M)	ISO 3166 country subdivision code without the country name code, corresponding to Billing address State (if available) Below are the values for the administrative regions in Greece A Eastern Macedonia and Thrace B Central Macedonia C Western Macedonia D Epirus E Thessaly F Ionian Islands G Western Greece H Central Greece I Attica J Peloponnese K Northern Aegean L Southern Aegean M Crete (For 3D Secure process purposes)	String (max. 3 characters)
ShipAddrCity (M)	Shipping address city (For 3D Secure process purposes) The parameter contains only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-	String (max. 50 characters)
ShipAddrCountry (M)	ISO 3166-1 numeric country code, corresponding to Shipping address country. E.g. 300 for Greece. (For 3D Secure process purposes)	String (max. 3 characters)
ShipAddrLine1 (M)	Additional line 1 of the shipping address (For 3D Secure process purposes)	String (max. 50 characters)

	The parameter contains only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-	
ShipAddrLine2 (O)	Additional line 2 of the shipping address (For 3D Secure process purposes) The parameter contains only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-	String (max. 50 characters)
ShipAddrLine3 (O)	Additional line 3 of the shipping address (For 3D Secure process purposes) The parameter contains only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-	String (max. 50 characters)
ShipAddrPostCode (M)	Post code of the shipping address (For 3D Secure process purposes)	String (max. 16 characters)
ShipAddrState (M)	ISO 3166 country subdivision code without the country name code, corresponding to Shipping address State (if available) Below are the values for the administrative regions in Greece A Eastern Macedonia and Thrace B Central Macedonia C Western Macedonia D Epirus E Thessaly F Ionian Islands G Western Greece H Central Greece I Attica J Peloponnese K Northern Aegean L Southern Aegean M Crete (For 3D Secure process purposes)	String (max. 3 characters)
CardholderName (M)	Name of the card holder (For 3D Secure process purposes) The parameter contains only Latin (**not Greek**) lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-	String (min. 2, max. 45 characters)
Email (M)	E-mail of the card holder (For 3D Secure process purposes) The parameter shall meet requirements of Section 3.4 of IETF RFC 5322.	String (max. 254 characters)

HomePhone (M)	Home phone number of the card holder, in (..3-..15) format, namely (up to 3 characters dash up to 15 characters). E.g. 210-3288000 (For 3D Secure process purposes)	String (max. 19 characters)
MobilePhone (M)	Mobile number of the card holder, in (..3-..15) format, namely (up to 3 characters dash up to 15 characters). E.g. +30-6972222222 (For 3D Secure process purposes)	String (max. 19 characters)
WorkPhone (M)	Work phone number of the card holder, in (..3-..15) format, namely (up to 3 characters dash up to 15 characters). E.g. 210-3288000 (For 3D Secure process purposes)	String (max. 19 characters)
RecurringInd (O)	The parameter is only used when it concerns the first transaction of a recurring payment (i.e. standing order) performed online by the cardholder. **Attention!** If the transaction does not concern a recurring payment, either the parameter will not be sent, or null value should be sent (after serialization, parameter will be displayed as follows: <RecurringInd xsi:nil="true" />) Potential values: ▪ R, for recurring transactions (transactions performed at regular intervals) ▪ C, for unscheduled recurring transactions (transactions performed at irregular intervals)	String (1 character)
RecurPurchaseDate (O)	If the parameter RecurringInd has value, it contains the current date in YYYYMMDDHHMMSS format.	String (max. 14 characters)
RecurFreq (O)	If the parameter RecurringInd has value, it contains the recurrence frequency of the transaction. It is an integer that expresses number of days.	String (max. 4 characters)
RecurEnd (O)	If the parameter RecurringInd has value, it contains the expiry date of the recurring payments in YYYYMMDD format.	String (max. 8 characters)

Attention:
Parameters concerning the 3D Secure process are mandatory, as Issuers may soon be rejecting transactions lacking this information.

Below there is a list of the information received from the «*ticketing Web Service*»:

RESPONSE PARAMETERS		
Parameter name	**Description**	**Type**
ResultCode	The result code indicating whether the call was successful. More specifically: ▪ **Value = 0**: The transaction details were successfully reported and the response contains a value in the «*TranTicket*» parameter. ▪ **Value ≠ 0:** Neither transaction details were successfully reported nor was a value in the «*TranTicket*». The description of the problem encountered is in the «*RedultDescription*» field.	String (max. 5 characters)
ResultDescription	Description of the «*ResultCode*» in the event of a problem.	String (max. 1024 characters)
TranTicket	The transaction ticket value to be used to authenticate the response to be received by the site when the transaction has been executed (see **section 5 - *«Hash Key» Verification***) ⚠ **Attention!** ▪ This value must be temporarily stored in a secure manner because it will be used to authenticate the response to be received from the Paycenter. ▪ In no case **may this value be visible to the user** (e.g. not to be transferred via hidden parameters to an html form).	String (32 characters)
Timestamp	The date and time when the request details became valid.	DateTime
MinutesToExpiration	The number of minutes during which the request details and transaction ticket are valid.	Integer

5. Data Submission – Transaction Response Receipt

When the ticketing mechanism is completed and transaction details have been reported successfully, the user is redirected to the payment page by submitting a form where the transaction information is sent with hidden parameters and using the POST method.

Attention!

- The values of all the parameters (both on the form via which the user is redirected and in the response returned to the site) are in **UTF-8 encoding**.
- For the proper operation of the page where the user enters the card details, javascript must be activated in the browser.

The POST URL is indicated below:

https://paycenter.piraeusbank.gr/redirection/pay.aspx

A sample form is included in **Annex 1**.

When the user has been redirected to the payment page, has filled out the card details and the transaction has been sent to the Paycenter, the details are then processed. A page appears on the browser informing the user about the transaction success/failure. Then the user is directed back to the merchant site (either to the success page if the transaction was successfully completed, or otherwise to the failure page). The flow of user actions is illustrated in the following chart:

Chart 2: *User actions*

(*): *For information about the possible ways to send a response to the site (POST or GET method), see below under response description.*

In **Annex 2** there are sample screens of the payment and information pages that are displayed after a successful transaction.

Note:
The Piraeus Bank page where the user enters his card details may be partly modified (e.g. merchant logo display). See **Annex 3**.

Below there is an analysis of the parameters sent via the form to the Paycenter as well as those sent as a response from the Paycenter to the site.

FORM PARAMETERS		
Parameter name	**Description**	**Type**
AcquirerId (M)	The acquirer identification. Provided by Piraeus Bank.	Integer
MerchantId (M)	The merchant identification. Provided by Piraeus Bank.	Integer
PosId (M)	The POS identification. Provided by Piraeus Bank.	Integer
User (M)	User name. Provided by Piraeus Bank.	String (max. 50 characters)
LanguageCode (M)	The code of the language in which the payment page will be displayed. Possible values: ▪ **el-GR**: Greek ▪ **en-US**: English ▪ **ru-RU**: Russian ▪ **de-DE:** German	String (max. 5 characters)
MerchantReference (M)	The reference of the transaction the details of which have already been reported via the ticketing Web Service. ▪ «MerchantReference» can be up to 50 characters long, containing only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+- ▪ It has to be unique to each successful transaction. **Note:** If a transaction is not successful, the transaction may be resent with the same MerchantReference provided that the «ticketing mechanism» is repeated.	String (max. 50 characters)
ParamBackLink (O)	The variable content is used as a parameter ('query string') in the URL returned to the user when the «Cancel» button is pressed. Example: If http://www.site.gr/cancel has been designated as backlink URL and the ParamBackLink parameter value is «*p1=v1&p2=v2*», then the «Cancel»	String (up to 0.5 Kb)

	button will direct the user to the URL: http://www.site.gr/cancel?p1=v1&p2=v2. **Note:** The parameter value must not include the character «?» as a prefix.	

Attention!

All the transaction details (e.g. type, amount, number of installments) must be reported in advance through the «ticketing mechanism» (see section 4)

Once the details sent have been processed, the response is sent to the success page URL if the transaction was successfully executed, or the failure page URL if the transaction was not successfully executed.

The ePOS Paycenter response may be sent to the site in 2 ways:

- Either detailed information about the process result will be sent to the merchant site (success/failure URL) through the POST method;
- Or only key information will be sent to the merchant site through the GET method (i.e. as parameters in the URL). If detailed information is required for the transaction, the merchant system may call the «*follow-up*» Web Service later to get detailed information about all the response parameters.

Note:

In the event of interest in the **«follow-up» Web Service**, the technical specifications should be requested from Piraeus Bank.

1) Response Receipt in the POST method

The following parameters are sent with the POST method:

RESPONSE PARAMETERS (POST METHOD)		
Parameter name	**Description**	**Type**
SupportReferenceID	Reference id of the submitted request. There is a different value per request (even if the transaction failed). **Note:** It is necessary to store the value, so as to be used as a reference in the communication with Piraeus Bank, as required.	Integer
ResultCode	The request result code indicating whether there was any technical problem in the transaction processing. Specifically: ▪ **Value = 0**: There was no problem; the transaction was executed. Then, the «StatusFlag» parameter must be checked to verify that the transaction was approved. ▪ **Value ≠ 0**: There was a transaction data problem or technical problem at the Paycenter, so the transaction failed and the card was not debited. The «ResultDescription» parameter contains the problem description. **Note:** The most frequent «ResultCode» values are shown in **Annex 6**.	String (max. 5 characters)
ResultDescription	The description corresponding to the «ResultCode» parameter value. **Note:** ▪ This information should not be displayed to the user. ▪ If the request is rejected due to anti-fraud checks (ResultCode=7001, see **Annex 6**), the «ResultDescription» parameter contains the code of the rule that was fired-up. The zero value (0) means that the card number is included in a black list. If special anti-fraud rules	String (max. 1024 characters)

	have been agreed with the merchant, Piraeus Bank will provide the relevant rule codes that may be returned.	
StatusFlag	The parameter value indicating whether the transaction was approved. Possible values: ▪ **Success**: Transaction approved by the Issuer. ▪ **Failure**: Transaction declined by the Issuer.	String (7 characters)
ResponseCode	When a transaction has been executed, it contains a response code. The response codes for approved transaction are: **00, 08, 10, 16**. **Note:** The most frequent «ResponseCode» values are shown in **Annex 6**.	String (2 characters)
ResponseDescription	The description corresponding to the «ResponseCode» parameter value.	String (max. 120 characters)
LanguageCode	The language code sent with the request. Possible values: ▪ **el-GR**: Greek ▪ **en-US**: English ▪ **ru-RU**: Russian ▪ **de-DE:** German	String (5 characters)
MerchantReference	The transaction reference sent with the request.	String (max. 50 characters)
TransactionDateTime	▪ If no transaction has been executed (i.e. when ResultCode ≠ 0), the value is a dash «-» . ▪ If a transaction has been executed (i.e. when ResultCode = 0), the value is the transaction execution date and time in the **DD/MM/YYYY HH24:MM:SS** format.	String (19 characters)
TransactionId	If a transaction has been executed, it takes a unique transaction id generated by ePOS Paycenter.	Integer
CardType	The type of the card used for the transaction. Possible values: ▪ **1:** Visa ▪ **2:** Mastercard ▪ **3:** Maestro ▪ **4:** American Express (supported	Integer

	only if included in the agreement with the merchant) ▪ **5:** Diners or Discover (supported only if included in the agreement with the merchant) ⚠ **Attention!** Diners/Discover and American Express transactions use a **different MerchantId and PosId value** than the Visa / Mastercard / Maestro transactions.	
PackageNo	If a transaction has been executed (i.e. when ResultCode=0), it takes the number of the package that includes this transaction.	Integer
ApprovalCode	If a successful transaction has been executed (i.e. when ResultCode=0 and StatusFlag=Success), it takes the transaction approval code.	String (max. 6 characters)
RetrievalRef	If a transaction has been executed (i.e. when ResultCode=0), it takes the Retrieval Reference Number generated by the acquiring system.	String (max. 12 characters)
AuthStatus	It concerns the result of the strong customer authentication process (3D-Secure) which is applied to Visa, Mastercard and Maestro transactions. Possible values: ▪ **01**: Successful completion of the strong customer authentication process ▪ **03**: Failure of the strong customer authentication process	String (2 characters)
Parameters	The content of «parameters» variable that had been sent with the request is returned to the success/failure pages.	String (max. 512 characters)
HashKey	If the transaction is successful (i.e. when ResultCode=0 and StatusFlag=Success), it takes a value that will be used by the merchant system to authenticate the response (see below – ***«Hash Key» Verification***). If the transaction failed, it is blank.	String (64 characters)
PaymentMethod	The payment method that was used in a completed transaction. Possible values: ▪ **Card:** The payment was made through direct data entry ▪ **MasterPass:** The payment was	String

	made using the digital MasterPass Wallet ▪ **IRIS:** The payment was made using IRIS online payments.	
TraceID	Transaction reference that is generated by Visa/Mastercard and is recommended to be stored to the merchant's system. Use at recurring transactions: When the Ticketing Web Service is called including a value in RecurringInd parameter, the TraceID value should be stored in order to be used in the next recurring payments (through Web Service or Batch File solution).	String (max. 50 characters)

2) Response Receipt in the GET method

The parameters sent through the GET method (i.e. included in the success or failure URL), are as follows:

RESPONSE PARAMETERS (GET METHOD)		
Parameter name	**Description**	**Type**
SupportReferenceID	Reference id of the submitted request. There is a different value per request (even if the transaction failed). **Note:** It is necessary to store the value, so as to be used as a reference in the communication with Piraeus Bank, as required.	Integer
ResultCode	The transaction result code indicating whether there was any technical problem in the transaction processing. Specifically: ▪ **Value = 0**: There was no technical problem; the transaction was executed. Then, the «StatusFlag» parameters must be checked to verify that the transaction was approved. ▪ **Value ≠ 0**: There was a transaction data problem or technical problem at the Paycenter, so the transaction failed and the card was not debited. The «ResultDescription» parameter contains the problem description. **Note:** The most frequent «ResultCode»	String (max. 5 characters)

	values are shown in **Annex 6**.	
StatusFlag	If the ResultCode takes a 0 value (i.e. the transaction was executed), its value indicates whether the transaction was approved. Possible values: ▪ **Success**: Transaction approved by the Issuer. ▪ **Failure**: Transaction not approved by the Issuer.	String (7 characters)
ResponseCode	When a transaction has been executed, it contains a response code. The response codes for approved transactions are: **00, 08, 10, 16**. **Note:** The most frequent «ResponseCode» values are shown in **Annex 6**.	String (2 characters)
MerchantReference	The transaction reference sent with the request.	String (max. 50 characters)
TransactionId	If the transaction has been executed, it takes a unique transaction id generated by ePOS Paycenter.	Integer
PackageNo	If a transaction has been executed (i.e. when ResultCode=0), it takes the number of the package that includes this transaction.	Integer
ApprovalCode	If a successful transaction has been executed (i.e. when ResultCode=0 and StatusFlag=Success), it takes the transaction approval code.	String (max. 6 characters)
AuthStatus	It concerns the result of the strong customer authentication process (3D-Secure) which is applied to Visa, Mastercard and Maestro transactions. Possible values: ▪ **01**: Successful completion of the strong customer authentication process ▪ **03**: Failure of the strong customer authentication process	String (2 characters)
Parameters	The content of «parameters» variable that had been sent via the ticketing Web Service is returned to the success/failure pages.	String (max. 512 characters)
HashKey	If the transaction is successful (i.e. when ResultCode=0 and StatusFlag=Success), it	String (64

	takes a value that will be used by the merchant system to authenticate the response (see below – ***«Hash Key» Verification***). If the transaction failed, it is blank.	characters)
PaymentMethod	The payment method that was used in a completed transaction. Possible values: ▪ **Card:** The payment was made through direct data entry ▪ **MasterPass:** The payment was made using the digital MasterPass Wallet ▪ **IRIS:** The payment was made using IRIS online payments.	String
TraceID	Transaction reference that is generated by Visa/Mastercard and is recommended to be stored to the merchant's system. Use at recurring transactions: When the Ticketing Web Service is called including a value in RecurringInd parameter, the TraceID value should be stored in order to be used in the next recurring payments (through Web Service or Batch File solution).	String (max. 50 characters)

Note:
When a response is sent in the GET method, if the merchant system needs detailed information about the transaction, it may call the **«follow-up» Web Service** later. Then the latter will send all the information about a «MerchantReference»-specific request.

In the event of interest, the technical specifications should be requested from Piraeus Bank.

«Hash Key» Verification

If a transaction has been successfully executed (regardless of the response receipt method – GET or POST), the «HashKey» response parameter contains a value that will be used to verify the response received. Specifically, the following must be performed:

A) Calculation of the expected Hash Key:

The merchant system should calculate the «HashKey» expected to receive in the response for a «MerchantReference»-specific transaction:

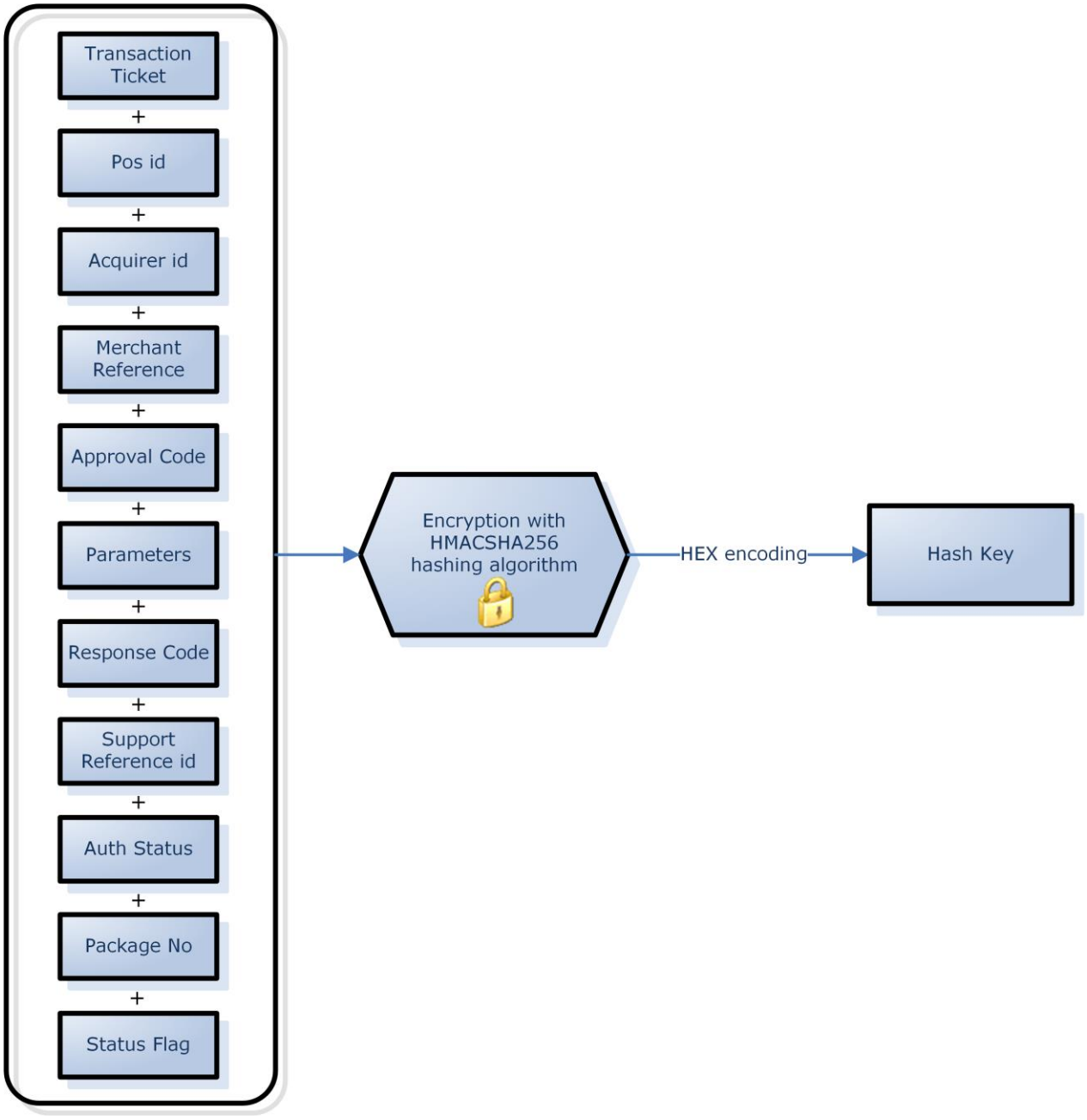


Chart 3: *«Hash Key» Calculation*

Namely:

- An alpharithmetic string is generated bringing together the following in the following order:
  - **Transaction Ticket** («TranTicket» parameter of the ticketing Web Service response)
  - **Pos id** («PosId» parameter of the ticketing Web Service response)
  - **Acquirer id** («AcquirerId» parameter of the ticketing Web Service response)
  - **Merchant Reference** («MerchantReference» parameter of the response sent to the success page)
  - **Approval Code** («ApprovalCode» parameter of the response sent to the success page)
  - **Parameters** («Parameters» parameter of the response sent to the success page)
  - **Response Code** («ResponseCode» parameter of the response sent to the success page)
  - **Support Reference id** («SupportReferenceID» parameter of the response sent to the success page)
  - **Auth Status** («AuthStatus» parameter of the response sent to the success page)
  - **Package No** («PackageNo» parameter of the response sent to the success page)
  - **Status Flag** («StatusFlag» parameter of the response sent to the success page)

- During the concatenation of data, before the hashing, the special character "**;**" is used as a delimiter character between the fields.

- The generated value is encrypted via the **HMACSHA256** hashing encryption algorithm, using the «Transaction Ticket» value as the secret key.

- The encrypted UPPERCASE value via HEX encoding is the «Hash Key».

Note:
A «Hash Key» calculation example using specific values, is illustrated in **Annex 4**.

Attention!

The «Hash Key» must be calculated in **HEX encoding**.

B) «Hash Key» Comparison:

If the calculated Hash Key is the same as the response Hash Key sent to the success page («HashKey» parameter), then the response authenticity is verified.

If the two Hash Keys are different, **the order must not be completed.**

Automated Emailing to the Merchant

When a transaction has been processed (the response received via either GET, or POST), a notification email (in Greek or English language) may be sent automatically by Piraeus Bank to the merchant.
The address from which emails will be sent is paycenter@piraeusbank.gr and is used only to send, not receive emails.

There are the following alternatives for the merchant to choose:

- Email may be sent only for successful transactions.
- Email may be sent both for successful and unsuccessful transactions.
- In addition to either of the above or not, an email may be sent every time a user exits the payment page (i.e. the user is directed to the payment page, but no transaction is executed).

The email addresses and the types of emails to be received, must be provided by the merchant to Piraeus Bank.

Attention!

It is possible that an email is not sent to the merchant due to technical problems. Therefore, it is suggested that emails should not be the only way to receive successful transaction information.

6. Merchant Application Action Flow

Following the analysis of the individual process modules to be implemented (ticketing mechanism, form parameters, response parameters), the flow of actions to be performed by the merchant application in collaboration with ePOS Paycenter, for a transaction to be executed, is illustrated in the flowchart below.

It is important to use the proposed algorithm so that all cases are taken into account and no problems occur during the site productive operation.

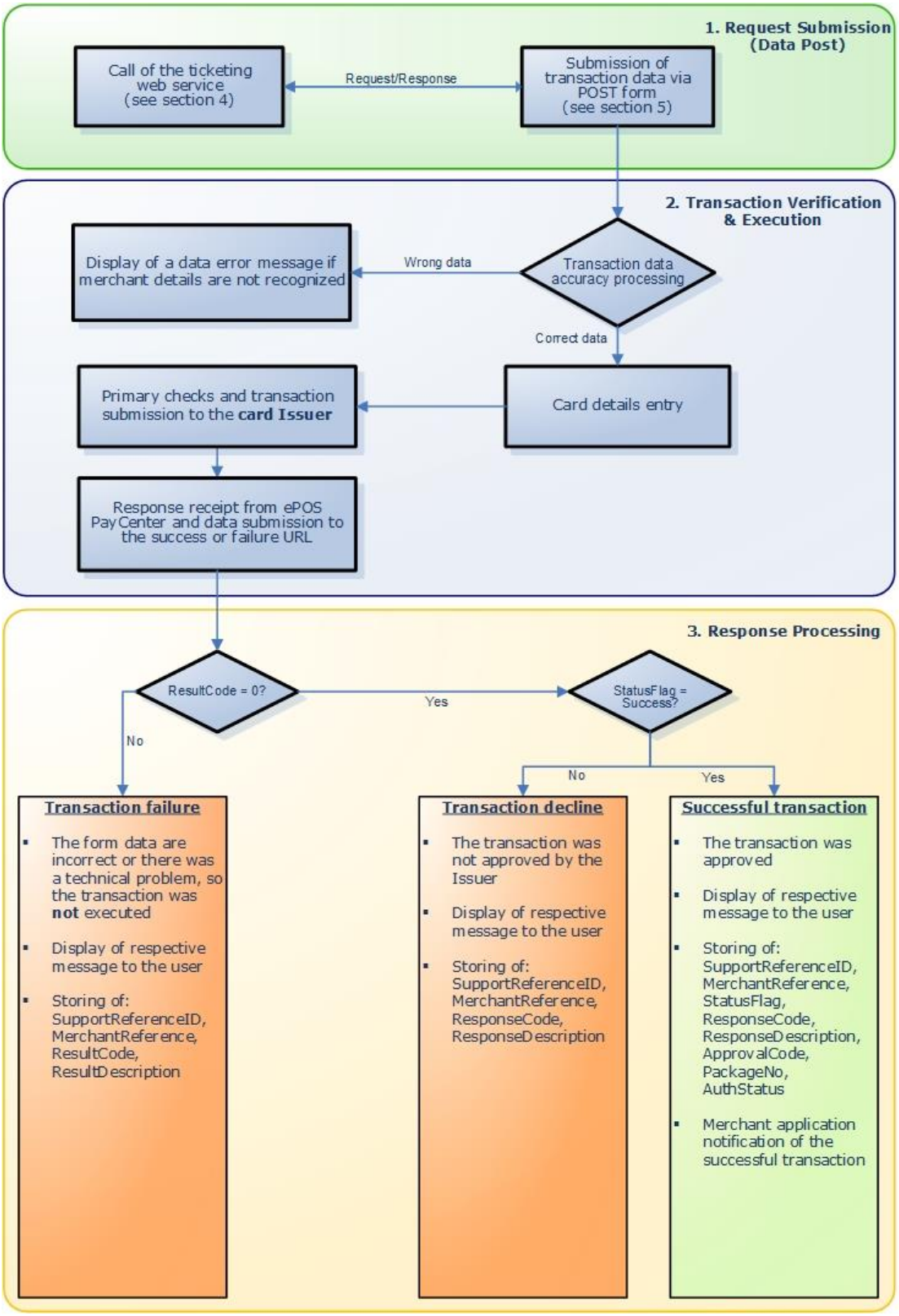


Chart 4: *Merchant application actions*

As shown in the chart, the overall process consists of 3 phases:

1.	**Request Submission (Data Post)**

The ticketing Web Service is used in order to report the details of the transaction to be sent with a specific «Merchant Reference» and acquire a «Transaction Ticket» (see section 4).

Then, use is made of a form with the POST method and required information as hidden parameters, so that the user may be directed to the page where he/she will enter his/her card details (see section 5).

2.	**Transaction Verification & Execution**

Once the Paycenter receives the form parameters, a primary check is run and if the details are correct, the payment page is configured. If the details sent via the form are not correct, then a relevant error message is displayed.
After the card details are filled out by the user, primary checks are run and the transaction is sent to the card Issuer.
Several seconds later, a response is received from the Issuer and a relevant response is sent to the success or failure URL of the merchant site.

3.	**Response Processing**

The merchant system must check the response parameters, to verify whether the transaction is successful. The cases of responses indicated in orange in the flow chart are sent to the failure page, whereas the cases in green are sent to the success page. Specifically:

- **If ResultCode ≠ «0»,** then there was either a problem with transaction details, or some **technical problem**, thus the transaction was not executed. A problem description is contained in the «ResultDescription» parameter (not to be displayed on the user page). If necessary, the details of the technical problem (SupportReferenceID, MerchantReference, ResultCode, ResultDescription) are stored in the merchant system.

- **If ResultCode = «0»:**
  - If StatusFlag ≠ «Success», then the transaction was executed but **not approved by the card Issuer**. If necessary, the details of the unsuccessful transaction (SupportReferenceID, MerchantReference, ResponseCode, ResponseDescription) are stored in the merchant system.

- If StatusFlag = «Success», then **the transaction was successful**, thus the transaction information, such as SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, ResponseDescription, ApprovalCode, PackageNo, AuthStatus should be stored and the merchant system notified of the successful transaction.

Attention!
The «SupportReferenceID» value should always be stored so that it may be used as reference in the communication between the merchant and Piraeus bank.

Note:
- It is suggested that the transaction «ApprovalCode» be indicated and/or sent on a transaction confirmation email from the merchant to the user.
- It is recommended that the transaction decline or technical error message should not appear as such on the user page.

7. Test Cases

Below there is a list and description of the test cases that may be executed in the test environment of the Redirection service. Test transactions must be performed for all the test cases that are marked as «MANDATORY». Optional test cases may be run to the extent that they are applicable to the system under implementation. All in all, the following tests are the most common scenarios occurring in a production system.

Below there is a test cases concise list:

No.	TITLE	MANDATORY
Test case 1	APPROVED TRANSACTION (VISA)	YES
Test case 2	DECLINED TRANSACTION	YES
Test case 3	RECHARGE ATTEMPT	YES
Test case 4	COMMUNICATION ERROR	YES
Test case 5	INVALID CARD NUMBER	YES
Test case 6	UNDER-PROCESS TRANSACTION WAS RE-SENT	YES
Test case 7	BATCH IS CLOSING	YES
Test case 8	GENERAL ERROR	YES
Test case 9	APPROVED TRANSACTION WITH INSTALLMENTS	NO
Test case 10	APPROVED TRANSACTION (MASTERCARD)	NO
Test case 11	APPROVED TRANSACTION (MAESTRO)	NO
Test case 12	APPROVED TRANSACTION (DINERS)	NO
Test case 13	APPROVED TRANSACTION (DISCOVER)	NO
Test case 14	APPROVED TRANSACTION (AMERICAN EXPRESS)	NO
Test case 15	APPROVED TRANSACTION (GBP)	NO
Test case 16	APPROVED TRANSACTION (USD)	NO

The following applies to all test cases:

- The «AcquirerId», «MerchantId», «PosId», «Username» and «Password» parameter values used in the ticketing Web Service and/or on the form via which details are sent with the POST method, are provided by Piraeus Bank.
- The «RequestType» parameter in the ticketing Web Service is entered depending on the transaction type (00 for authorization, 02 for sale).
- The «Amount» parameter in the ticketing Web Service, may take any valid value (see section 4).

- The «Installments», «ExpirePreauth» and «CurrencyCode» parameter values in the ticketing Web Service are entered according to the values provided in the test cases.
- The «Bnpl» parameter in the ticketing Web Service must always be given the value 0 (intended for future use) (see section 4).
- The «LanguageCode» parameter on the form, may take any valid value (see section 5).
- The «Parameters» parameter in the ticketing Web Service and the «ParamBackLink» parameter in the form, are entered according to the application requirements.

Note:

- It is reminded that preauthorization is a transaction by means of which the amount is simply committed. The preauthorization must be completed by the merchant (via the ePOS Paycenter AdminTool or a Web Service call) within the days defined via the «ExpirePreauth» parameter for the transaction to be settled.

- In preauthorization test transactions, the «ExpirePreauth» parameter must always be given the value 30, but in the live environment its value may vary from 2 to 30.

Test Case 1: APPROVED TRANSACTION (VISA)

MANDATORY

Scenario: Approval of transaction (without installments) with Visa card

It is applicable:

- When the response is sent to the success URL

  or

- When ResultCode=0 and StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	VISA
Card number for sale	4111111111111111
Card number for preauthorization	4000000000000002
Expiry month	01
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	03

Merchant application actions:

- Display of transaction approval message on the user page
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, PackageNo, ApprovalCode and AuthStatus parameter values
- Merchant application update for the successful transaction

Test Case 2: DECLINED TRANSACTION

MANDATORY

Scenario: Decline of a transaction

It is applicable:

- When the response is sent to the failure URL & ResultCode = 0

or

- When ResultCode = 0 & StatusFlag=Failure

Form parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Input parameters:

Type	VISA
Card number for sale	4111111111111111
Card number for preauthorization	4000000000000002
Expiry month	02
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	12
AuthStatus	03

Merchant application actions:

- Display of transaction decline message received from Issuer on the user page
- Storing of SupportReferenceID, MerchantReference, ResultCode, ResponseCode, ResponseDescription parameter values
- Merchant application update for the declined transaction

Test Case 3: RECHARGE ATTEMPT

MANDATORY

Scenario: Attempt to recharge a transaction (the request sent had a «MerchantReference» value already used for an approved transaction)

It is applicable:

- When the response is sent to the failure URL & ResultCode = 1048

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	VISA
Card number for sale	4111111111111111
Card number for preauthorization	4000000000000002
Expiry month	03
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	1048
ResponseCode	
AuthStatus	03

Merchant application actions:

- Display of a transaction failure message on the user page
- Storing of SupportReferenceID, MerchantReference, ResultCode, ResultDescription parameter values
- Merchant application update for the recharge attempt (if necessary, to make a thorough check)

Test Case 4: COMMUNICATION ERROR

MANDATORY

Scenario: Failure to execute a transaction due to (technical) communication problem with the transaction processing system

It is applicable:

- When the response is sent to the failure URL & ResultCode = 50x (i.e. 500, 501 etc.)

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	VISA
Card number for sale	4111111111111111
Card number for preauthorization	4000000000000002
Expiry month	04
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	500
ResponseCode	
AuthStatus	03

Merchant application actions:

- Display of a transaction failure message on the user page (with a prompt to try again later)
- Storing of SupportReferenceID, MerchantReference, ResultCode, ResultDescription parameter values
- Merchant application update for the failure to execute the transaction

Test Case 5: INVALID CARD NUMBER

MANDATORY

Scenario: Failure to execute a transaction due to incorrect card details or a card not supported by the system

It is applicable:

- When the response is sent to the failure URL & ResultCode = 981

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	VISA
Card number for sale	4111111111111111
Card number for preauthorization	4000000000000002
Expiry month	05
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	981
ResponseCode	
AuthStatus	03

Merchant application actions:

- Display of a transaction failure message on the user page (with a prompt to try again and check the card details or enter a different card).
- Storing of SupportReferenceID, MerchantReference, ResultCode, ResultDescription parameter values.
- Merchant application update for the failure to execute the transaction.

Test Case 6: UNDER-PROCESS TRANSACTION WAS RE-SENT

MANDATORY

Scenario: Attempt to send a transaction with the same «MerchantReference» as that of the transaction currently processed by ePOS Paycenter (it is possible that a response has not been received from the Issuer or a problem has occurred in the transaction processing system; as a result the transaction is «stalled»)

It is applicable:

- When the response is sent to the failure URL & ResultCode = 1045

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	VISA
Card number for sale	4111111111111111
Card number for preauthorization	4000000000000002
Expiry month	06
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	1045
ResponseCode	
AuthStatus	03

Merchant application actions:

- Display of a transaction failure message on the user page (with a prompt to try again later)
- Storing of SupportReferenceID, MerchantReference, ResultCode, ResultDescription parameter values
- Merchant application update for the failure to execute the transaction prompting the merchant to investigate the transaction status via the epos Paycenter AdminTool

Note:

A user prompt to try again later is recommended, because if the transaction is finally executed successfully, then a subsequent attempt will reproduce Test case 3.

Test Case 7: BATCH IS CLOSING

MANDATORY

Scenario: Failure to execute a transaction because the current transaction batch is being settled (batch closing)

It is applicable:

- When the response is sent to the failure URL & ResultCode = 1072

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	VISA
Card number for sale	4111111111111111
Card number for preauthorization	4000000000000002
Expiry month	07
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	1072
ResponseCode	
AuthStatus	03

Merchant application actions:

- Display of a transaction failure message on the user page (with a prompt to try again later)
- Storing of SupportReferenceID, MerchantReference, ResultCode, ResultDescription parameter values
- Merchant application update for the failure to execute the transaction (if necessary)

Test Case 8: GENERAL ERROR

MANDATORY

Scenario: Failure to execute a transaction due to a temporary technical problem

It is applicable:

- When the response is sent to the failure URL & ResultCode = 1

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	VISA
Card number for sale	4111111111111111
Card number for preauthorization	4000000000000002
Expiry month	08
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	1
ResponseCode	
AuthStatus	03

Merchant application actions:

- Display of a transaction failure message on the user page (with a prompt to try again later)
- Storing of SupportReferenceID, MerchantReference, ResultCode, ResultDescription parameter values
- Merchant application update for the failure to execute the transaction (if necessary)

Test Case 9: APPROVED INSTALLMENT TRANSACTION

OPTIONAL

Scenario: Approval of installment transaction

It is applicable:

- When the response is sent to the success URL

  or

- When ResultCode=0 & StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	3
Amount	*Greater than 90*

Card details:

Type	VISA
Card number for sale	4908440000000003
Card number for preauthorization	4908460000000001
Expiry month	08
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	03

Merchant application actions:

- Display of transaction approval message on the user page in «x» installments (where «x», the number of installments entered)
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, TransactionId, PackageNo, ApprovalCode, AuthStatus parameter values
- Merchant application update for the successful transaction

Test Case 10: APPROVED TRANSACTION (MASTERCARD)

OPTIONAL

Scenario: Approval of transaction (without installments) with Mastercard

It is applicable:

- When the response is sent to the success URL

  or

- When ResultCode=0 and StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	MasterCard
Card number for sale	5100150000000001
Card number for preauthorization	5100160000000000
Expiry month	01
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	03

Merchant application actions:

- Display of transaction approval message on the user page
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, PackageNo, ApprovalCode and AuthStatus parameter values
- Merchant application update for the successful transaction

Test Case 11: APPROVED TRANSACTION (MAESTRO)

OPTIONAL

Scenario: Approval of transaction (without installments) with Maestro card

It is applicable:

- When the response is sent to the success URL

  or

- When ResultCode=0 and StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	Maestro
Card number for sale	6773111111111115
Card number for preauthorization	6773110000000009
Expiry month	01
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	03

Merchant application actions:

- Display of transaction approval message on the user page
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, PackageNo, ApprovalCode and AuthStatus parameter values
- Merchant application update for the successful transaction

Test Case 12: APPROVED TRANSACTION (DINERS)

OPTIONAL

Scenario: Approval of transaction (without installments) with Diners card

It is applicable:

- When the response is sent to the success URL

  or

- When ResultCode=0 and StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	Diners
Card number for sale	36131111111119
Card number for preauthorization	36131100000000
Expiry month	01
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	

Merchant application actions:

- Display of transaction approval message on the user page
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, PackageNo, ApprovalCode and AuthStatus parameter values
- Merchant application update for the successful transaction

Test Case 13: APPROVED TRANSACTION (DISCOVER)

OPTIONAL

Scenario: Approval of transaction (without installments) with Discover card

It is applicable:

- When the response is sent to the success URL

  or

- When ResultCode=0 and StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	Diners/Discover
Card number for sale	6011111111111117
Card number for preauthorization	6011000000000004
Expiry month	01
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	

Merchant application actions:

- Display of transaction approval message on the user page
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, PackageNo, ApprovalCode and AuthStatus parameter values
- Merchant application update for the successful transaction

Test Case 14: APPROVED TRANSACTION (AMERICAN EXPRESS)

OPTIONAL

Scenario: Approval of transaction (without installments) with American Express card

It is applicable:

- When the response is sent to the success URL

  or

- When ResultCode=0 and StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	978
Installments	0

Card details:

Type	American Express
Card number for sale	375537111111116
Card number for preauthorization	375537000000008
Expiry month	01
Expiry year	*Any future year*
CVV2	1234

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	

Merchant application actions:

- Display of transaction approval message on the user page
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, PackageNo, ApprovalCode and AuthStatus parameter values
- Merchant application update for the successful transaction

Test Case 15: APPROVED TRANSACTION (GBP)

OPTIONAL

Attention! For every different currency, a different test and live account is required

Scenario: Approval of transaction (without installments) in GBP currency

It is applicable:

- When the response is sent to the success URL

or

- When ResultCode=0 and StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	826
Installments	0

Card details:

Type	Visa
Card number for sale	4001151111111110
Card number for preauthorization	4001150000000004
Expiry month	01
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	03

Merchant application actions:

- Display of transaction approval message on the user page
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, PackageNo, ApprovalCode and AuthStatus parameter values
- Merchant application update for the successful transaction

Test Case 16: APPROVED TRANSACTION (USD)

OPTIONAL

Attention! For every different currency, a different test and live account is required

Scenario: Approval of transaction (without installments) in USD currency

It is applicable:

- When the response is sent to the success URL

or

- When ResultCode=0 and StatusFlag=Success

Input parameters:

Parameter	Value
RequestType for sale	02
RequestType for preauthorization	00
ExpirePreauth for sale	0
ExpirePreauth for preauthorization	30
CurrencyCode	840
Installments	0

Card details:

Type	Visa
Card number for sale	4408661111111117
Card number for preauthorization	4408660000000001
Expiry month	01
Expiry year	*Any future year*
CVV2	123

Response parameters:

Parameter	Value
ResultCode	0
ResponseCode	00
AuthStatus	03

Merchant application actions:

- Display of transaction approval message on the user page
- Storing of SupportReferenceID, MerchantReference, StatusFlag, ResponseCode, PackageNo, ApprovalCode and AuthStatus parameter values
- Merchant application update for the successful transaction

8. Use of Icons

Through the Redirection service, the merchant site supports and is protected by the cardholder authentication services «Verified by Visa» (for Visa card transactions) and «Mastercard SecureCode» (for Mastercard and Maestro transactions).

When transactions are performed via the Redirection service, no implementation is required to support the strong customer authentication services. The only prerequisite is to post on the site certain icons complying with the Visa and Mastercard specifications.

All relevant material can be downloaded from the following link :
https://paycenter.piraeusbank.gr/services/Manuals/Icons/Icons.zip

Specifically:

Supported cards icons

The icons of the supported cards are included in the folder (Icons/CardsIcons) and they are as follows:

Visa (*Visa.jpg*)
Mastercard (*Mastercard.jpg*)
Maestro (*Maestro.jpg*)

If the merchant supports Diners/Discover and/or American Express cards, then the relevant cards icons must be also included:

Diners (*Diners.jpg*)
Discover (*Discover.jpg*)
American Express (*Amex.jpg*)

These icons must be displayed on the site homepage.

Note:
If MasterPass payments are supported, the relevant icon should be included according to the instructions in the corresponding service manual.

3D-Secure process icons

On the site homepage and security information page (if applicable), the following icons must be displayed:

- **Verified by Visa service:**

The following icon must be displayed (Icons/VbV/vbv.jpg) which is a link to a new window in the URL:

- **Greek version:**
  https://paycenter.piraeusbank.gr/redirection/Content/HTML/3DSecure_el.html
- **English version:**
  https://paycenter.piraeusbank.gr/redirection/Content/HTML/3DSecure_en.html

- **MasterCard SecureCode service:**

One of the icons included in the (Icons/SecureCode) folder must be displayed and will be a link to a new window in the URL:

- **Greek version:**
  https://paycenter.piraeusbank.gr/redirection/Content/HTML/3DSecure_el.html
- **English version:**
  https://paycenter.piraeusbank.gr/redirection/Content/HTML/3DSecure_en.html

For example, the icons in the footer of your homepage could be displayed as below:

Piraeus Bank logo

Piraeus Bank logo can optionally be displayed on the merchant site. The relevant icons are included in the (Icons/PiraeusBank) folder.

9. Tips

Below, there are some remarks-tips which must be taken into account:

- The «Password» parameter in the ticketing Web Service (see section 4) must be sent encrypted with the MD5 hashing algorithm.
- The transaction amount and number of installments entered via the ticketing Web Service for a transaction with specific «Merchant Reference» (see section 4), should be stored in the merchant system so that their values are known.
- For the proper operation of the page where the user enters the card details, javascript must be activated in the browser.
- The values of all the parameters on the form via which the user is redirected to the card details entry page, as well as the parameters returned to the merchant site are in **UTF-8 encoding**.
- The «MerchantReference» parameter value, used in the ticketing Web Service and on the form via which the user is redirected to the payment page, must be unique for each successful transaction. Of course, if a transaction fails, another transaction may be sent with the same «MerchantReference» **provided that the ticketing process is repeated**.

> **Note:**
>
> If the ticketing Web Service is called more than once using the same «MerchantReference» (e.g. due to transaction failure), a different «transaction ticket» will be returned each time. The details of the last call are always the ones in effect.

- It is important that the «**MerchantReference**» parameter takes a value that has a special meaning and is known to the merchant (e.g. order number, contract number, etc.). This value, uniquely designating every successful transaction, appears in the «AdminTool» provided by Piraeus Bank to merchants to monitor their transactions. Using the «AdminTool», merchants can find transactions using the «**MerchantReference**» value as search criterion.
- «MerchantReference» can be up to 50 characters long, containing only Greek or Latin lowercase & uppercase alphanumeric characters, space, or the following special characters /:_().,+-
- For better merchant support by Piraeus Bank, the «**SupportReferenceID**» parameter should be stored with every attempt and be available to the merchant managers, so that it may be used in the communication with Piraeus Bank to solve potential problems. The same parameter must be sent by technical managers to Piraeus Bank in the event of problems during the test transactions.

- It is necessary to authenticate any response received by the site for a transaction by verifying the «HashKey» (see section 5 – *«Hash Key» Verification*).

- The value of the «Transaction Ticket» returned by the ticketing Web Service **may in no case be visible to the user** (e.g. not to be transferred via hidden parameters to an html form).

- It is not allowed to use frames of any kind (frames/iframes) on the page via which data are sent (via POST).

- It is possible to create a dynamic URL to which the user is directed when the «Cancel» button is pressed. In particular, the «ParamBackLink» parameter contains an alpharithmetic string to be used as a parameter ('query string') in the backlink URL designated when the account was created.

  Example:
  If http://www.site.gr/cancel has been designated as backlink URL and the «ParamBackLink» parameter value is «*p1=v1&p2=v2*», then the «Cancel» button will direct you to the URL:

  http://www.site.gr/cancel?p1=v1&p2=v2

  Note:

  The «ParamBackLink» parameter value must not have the character «?» as a prefix.

- The description of the «ResultDescription» and «ResponseDescription» parameters may not be displayed to the user. These values may be stored in the merchant system and be available to the merchant; it is recommended that only general information messages are displayed to users.

- The URLs of a (live or test) account are designated per pos id. This means that, if transactions need to be sent from 2 different URLs, Piraeus Bank should be requested to generate 2 pos ids, and the merchant site should use a different pos id depending on the page used to send the form.

- It is possible to send automated emails to the merchant for transactions executed via its site (see section 5 – *Automated Emailing to the Merchant*).

10. Implementation Checklist

No.	TASK
1.	**CONTRACT SIGNING**
	Signing of acquiring contract for Redirection solution between the company and Piraeus Bank.
2.	**TECHNICAL IMPLEMENTATION**
	Implementation of: ■ Ticketing mechanism (see section 4) ■ Page through which data are sent via POST (see section 5) ■ Success page (see section 5) ■ Failure page (see section 5)
3.	**TEST ACCOUNT INFORMATION SUBMISSION**
	Submission of the required information to Piraeus Bank to create a test account (see section 3)
4.	**PERFORMANCE OF TEST TRANSACTIONS**
	■ Piraeus Bank forwards test account details: ▪ AcquirerId ▪ MerchantId ▪ PosId ▪ Username ▪ Password (solely used in the ticketing mechanism) ■ Use the test accounts to run the mandatory test cases and as many optional ones as required (see section 7)
5.	**USE OF ICONS**
	Display the necessary icons on the merchant site (see section 8)
6.	**TEST COMPLETION – SUBMISSION OF LIVE DATA**
	■ Inform Piraeus Bank about the successful completion of the test transactions and use of icons. ■ Forward the live technical data to Piraeus Bank: **Website URL:** The URL of the website (**Attention!** It must be the same as the URL designated by the merchant in the contract). **Referrer URL:** The page URL from which data will be sent. **Success URL:** The page URL to which the transaction success response will be sent. **Failure URL:** The page URL to which the transaction failure response will be sent.

	Backlink URL: The URL to which the user is returned when the «Cancel» button is pressed. **IP address:** The IP address of the server sending requests to the ticketing mechanism. ■ Send to Piraeus Bank an email address belonging to the merchant; this email address will be used for informational purposes about ePOS Paycenter.
7.	**LIVE ACCOUNT RECEIPT**
	■ Piraeus Bank forwards live account details: ▪ AcquirerId ▪ MerchantId ▪ PosId ▪ Username ▪ Password (solely used in the ticketing mechanism) ■ Replace the test account details with the live account details.

Annex 1

Sample html form via which the transaction data are sent to the Paycenter (POST method) so that the user may be redirected to the payment page:

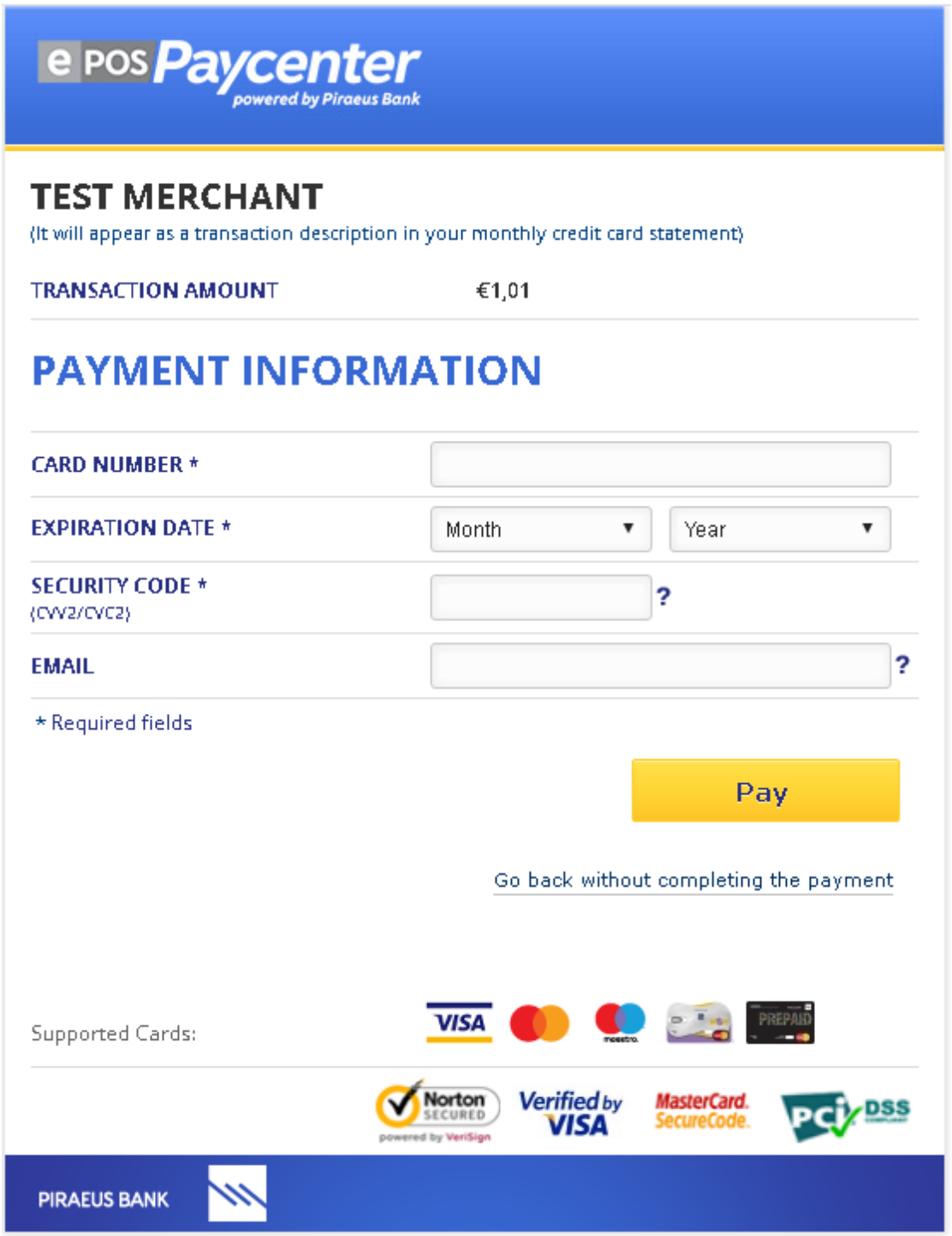
```
<form action=" https://paycenter.piraeusbank.gr/redirection/pay.aspx"
method="POST">
    <input name="AcquirerId" type="hidden" value="14" />
    <input name="MerchantId" type="hidden" value="140000001" />
    <input name="PosId" type="hidden" value="99999999" />
    <input name="User" type="hidden" value="certuser" />
    <input name="LanguageCode" type="hidden" value="el-GR" />
    <input name="MerchantReference" type="hidden" value="Test1" />
    <input name="ParamBackLink" type="hidden" value="p1=v1&p2=v2" />
    <input type="submit" value="Check out" />
</form>
```

Note:

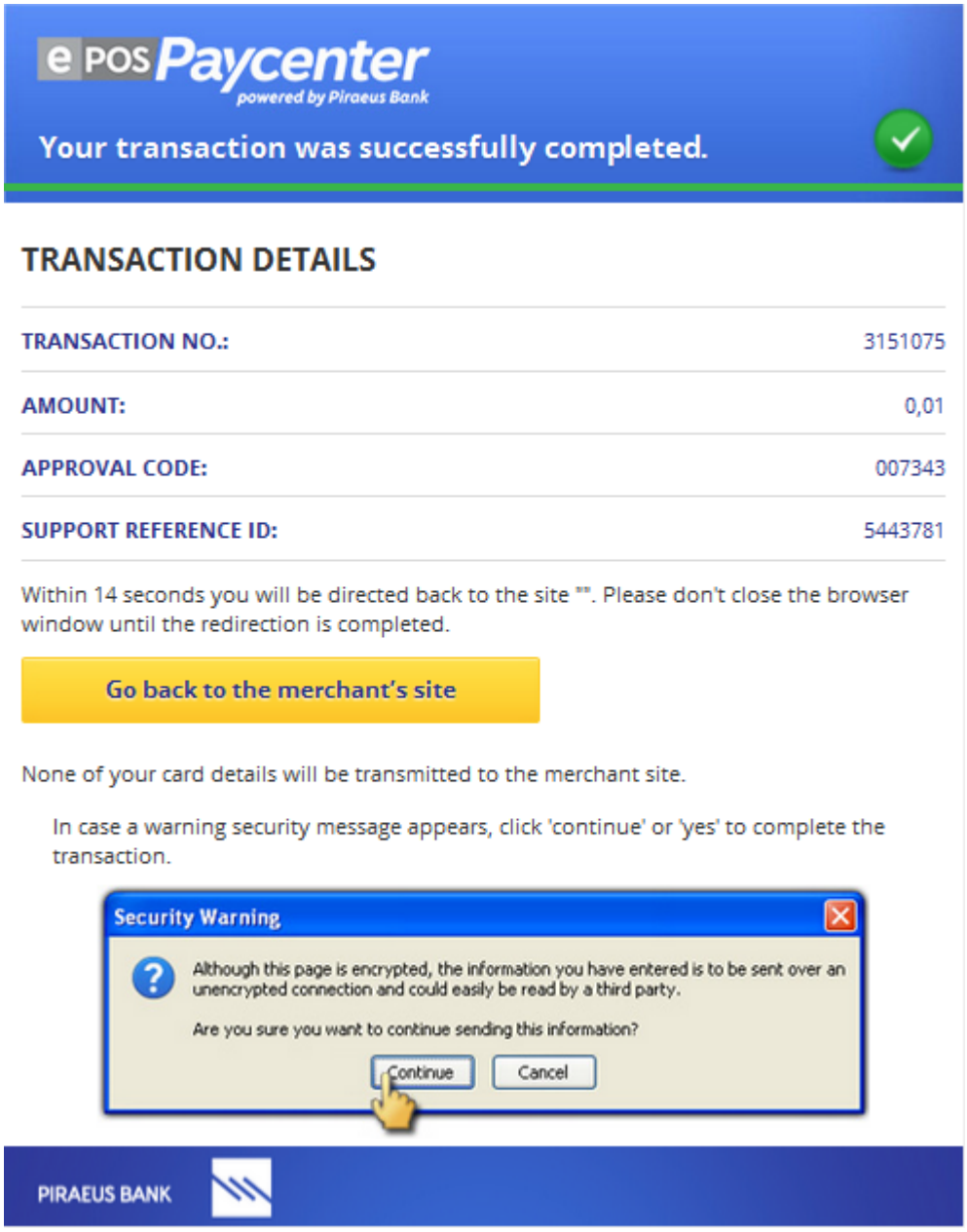
The parameter names are not case sensitive.

Annex 2

Screen shots from the payment and result page:



Screen shot 1: *Page where card data are entered*



Screen shot 2: *Result page*

The result page, where the transaction details are displayed (Screen shot 2), appears upon Merchant's request.

Annex 3

The payment page where the user enters the card details may be modified in the areas shown in the figure below:

Figure 3: *Payment page areas*

The areas shown in the figure above are defined by different «div» and formatted via CSS classes of a Style Sheet file. The default Style Sheet file («default.css») is included in the «StyleSheet» folder in the technical specifications and some of its classes may be modified:

Style Sheet file classes	
MainClass	Area 1 of figure
MainLeftClass	Area 2 of the figure
MainRightClass	Area 3 of the figure
LogoClass	Area 4 of the figure; may be used to post the merchant logo.
HeaderClass	Area 5 of the figure
FooterClass	Area 6 of the figure

If the merchant wishes to make some changes, the technical manager will have to send the modified Style Sheet file to Piraeus Bank.

Annex 4

Below there is an example of «Hash Key» calculation with specific values, which can be used to test whether the merchant system performs correct calculation.

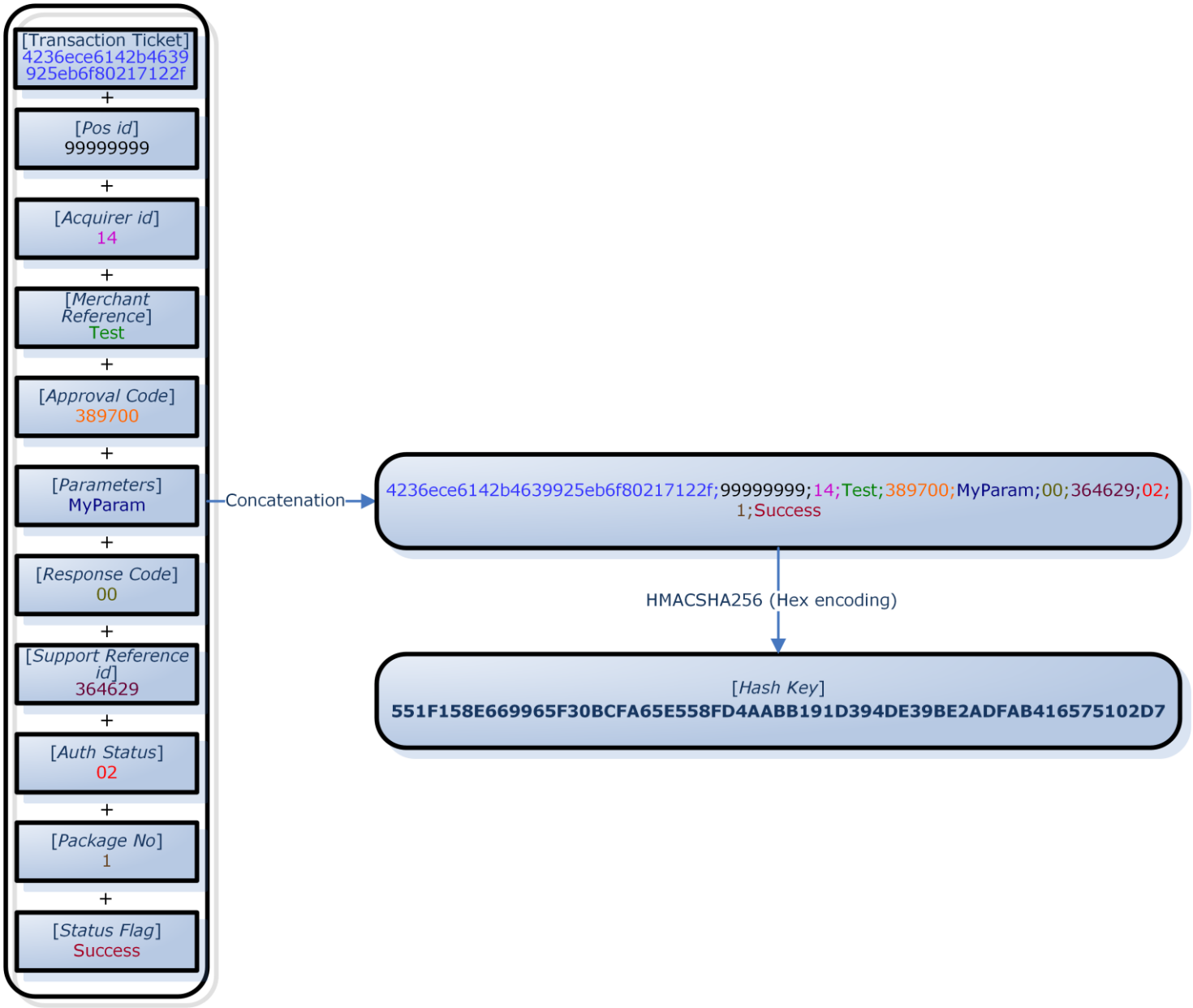


Chart 5: *Example of «Hash Key» calculation*

Namely, for the following specific values:

Parameter	Value
Transaction Ticket	4236ece6142b4639925eb6f80217122f
Pos id	99999999
Acquirer id	14
Merchant Reference	Test
Approval Code	389700
Parameters	MyParam
Response Code	00
Support Reference id	364629
Auth Status	02
Package No	1
Status Flag	Success

the correct «Hash Key» is:

Hash Key
551F158E669965F30BCFA65E558FD4AABB191D394DE39BE2ADFAB416575102D7

Annex 5

Below there is a list of all supported currency codes:

Currency Code	Currency
008	ALBANIAN LEK (ALL)
032	ARGENTINA PESO (ARS)
036	AUSTRALIAN DOLLAR (AUD)
124	CANADIAN DOLLAR (CAD)
152	CHILEAN PESO (CLP)
156	CHINESE YUAN (CNY)
170	COLOMBIAN PESO (COP)
191	CROATIAN KUNA (HRK)
203	CZECH KORUNA (CZK)
208	DANISH KRONE (DKK)
344	HONG KONG DOLLAR (HKD)
348	FIORINT (HUF)
356	INDIAN RUPEE (INR)
360	RUPIAH (IDR)
376	ISRAELI NEW SHEQEL (ILS)
392	YEN (JPY)
398	TENGE (KZT)
410	WON (KRW)
414	KUWAITI DINAR (KWD)
440	LITHUANIAN LITAS (LTL)
446	PATACA (MOP)
458	MALAYSIAN RINGGIT (MYR)
484	MEXICAN PESO (MXN)
504	MORROCAN DIRHAM (MAD)
554	NEW ZEALAND DOLLAR (NZD)
578	NORWEGIAN KRONE (NOK)
604	NUEVO SOL (PEN)
608	PHILIPPINE PESO (PHP)
643	RUSSIAN ROUBLE (RUB)
682	SAUDI RIYAL (SAR)

702	SINGAPORE DOLLAR (SGD)
710	RAND (ZAR)
752	SWEDISH KRONA (SEK)
756	SWISS FRANC (CHF)
764	BAHT (THB)
784	UNITED ARAB EMIRATES DIRHAM (AED)
818	EGYPTIAN POUND (EGP)
826	POUND STERLING (GBP)
840	US DOLLAR (USD)
933	BELARUSIAN RUBLE (BYN)
937	BOLIVAR FUERTE (VEF)
941	SERBIAN DINAR (RSD)
946	ROMANIAN LEU (RON)
949	TURKISH LIRA (TRY)
975	BULGARIAN LEV (BGN)
978	EURO (EUR)
980	UKRAINIAN HRYVNIA (UAH)
985	POLISH ZLOTY (PLN)
986	BRAZILIAN REAL (BRL)

Annex 6

The table below shows the most FREQUENT VALUES of the «ResultCode» (i.e. eventual technical problems) and «ResponseCode» (i.e. most common responses sent by Issuers) parameters.

ResultCode FREQUENT VALUES			
ResultCode	**ResultDescription**	**Explanation**	**Action**
1	An error occured. Please check your data or else contact Winbank PayCenter administrator	General error code which is returned when there is a technical problem	Try again later when the problem has been rectified
100	Authentication Error	Wrong value is used in «Username» / «User» parameter and/or «Password» parameter	Use correct values in «Username» / «User» and «Password» parameter
130	Field «x» contains invalid characters	The «x» parameter contains invalid characters.	Use valid value in «x» parameter
151	Check that field «x» contains data	No value is sent in «x» parameter	Send (valid) value in «x» parameter
215	AMEX cards require 4 digit cvv2	An American Express card was used and the cvv2 did not consist of 4 digits as it should	Re-send the transaction using the correct (4-digit) cvv2
216	Wrong cvv2	An invalid value was used in «Cvv2» parameter (e.g. characters)	Re-send the transaction using a valid cvv2
50x (e.g. 500, 501 etc.)	Communication Error	Communication problem with the transaction processing system	Try again later when the problem has been rectified

981	Invalid Card number/Exp Month/Exp Year	No valid values were used in card details (e.g. wrong card number, past expiration date etc) or unsupported card was used	Re-send the transaction using correct card details
1006	Unknown BIN	The user card is not eligible for Piraeus Bank's interest-free installments program	Use another card or re-send the transaction without installments
1007	Merchant does not support given bin	It concerns installment transaction. The card bin (i.e. the first 6 digits) may not be used in installment transaction in this merchant	Use another card or re-send the transaction without installments
1019	Too many installments asked	The number of installments requested is higher than the maximum allowed for this merchant	Use a lower number of installments
1026	Merchant does not support instalments	Installments were used in the transaction but the merchant does not support installments.	Contact Piraeus Bank in order to activate the use of installments
1034	Terminal does not support given card type	Transaction with unsupported card type	Contact Piraeus Bank
1040, 1041	«Error validating IP address. Contact sysadmin.» (1040), «Invalid IP address.» (1041)	The IP address validation failed as the request was sent through a server with different IP address than the one that was provided by the technical Manager to Piraeus Bank	Check the server's IP address and if it's necessary, contact Piraeus Bank in order to change the IP address that corresponds to the specific merchant id.
1045	Duplicate transaction references are not allowed	The request was sent with the same «MerchantReference» as	Try again later in order for the initial transaction to be completed. If the initial

		that of a transaction currently processed by ePOS Paycenter	transaction is finally approved, then the ResultCode 1048 will be returned in the new attempt (see test case 3 in section 7), otherwise a new transaction will be carried out. Alternatively, check if the initial transaction is approved using Paycenter AdminTool.
1048	Transaction already processed and completed	The request sent had a «MerchantReference» value already used for an approved transaction	Re-send the transaction using a different «MerchantReference» value.
1072	Pack is still closing	The batch settlement process is in progress (batch closing)	Try again later after the batch has been closed
1802	Wrong amount value	Invalid value used in «Amount» parameter (e.g. zero amount)	Use a valid value in «Amount» parameter
7001	*<Code of anti-fraud rule that was fired-up>*	The request was rejected due to anti-fraud checks. The «ResultDescription» parameter contains the code of the rule that was fired-up. The zero value (0) means that the card number is included in a black list. If special anti-fraud rules have been agreed with the merchant, Piraeus Bank will provide the relevant rule codes that may be returned.	Prompt the user for another form of payment or ask for a different card. ⚠ **Attention!** The end user should not be informed that the transaction was rejected due to anti-fraud checks.

ResponseCode FREQUENT VALUES				
ResponseCode	**ResponseDescription**	**Explanation**	**Action**	**Transaction approval**
00, 08, 10, 16	Approved or completed successfully	Transaction approval	Sale approval	Yes
05	Declined	Transaction declined by the Issuer	Cardholder should contact his/her Bank or use another card	No
12	Declined	Transaction declined by the Issuer	Cardholder should contact his/her Bank or use another card	No
51	Declined	Transaction declined by the Issuer	Cardholder should contact his/her Bank or use another card	No
34, 43	Lost card Stolen card,pick-up	Transaction declined by the Issuer	Cardholder should contact his/her Bank or use another card	No
54	Expired card	The card has expired and has not been renewed	Use another card	No
62	Restricted Card	Transaction declined by the Issuer	Cardholder should contact his/her Bank or use another card	No
92	Declined	Communication problem with the payment Organization (Visa, Mastercard etc)	Try again later when the problem has been rectified	No
I2	Installment amount bellow allowed	It concerns installment	Use o lower number	No

	minimum	transactions; the individual installment value is lower than the allowed minimum	of installments	

Note:

More values may be returned in addition to the ones listed in the tables above.

Glossary

Acquirer	An organization enabling merchants to execute card transactions. In this case, Piraeus Bank.
BIN	The first 6 digits of a card designating the Issuer Bank.
Hash Key	An alpharithmetic value generated by the merchant system using the «Transaction Ticket», to authenticate the successful transaction response received by the site from «ePOS Paycenter» comparing this value with the response Hash Key.
Live account	The merchant account through which live transactions are executed. It is made up of the following components: ▪ AcquirerId ▪ MerchantId ▪ PosId ▪ Username ▪ Password
Merchant id	The «*merchant identification*».
Pos id	The «*pos identification*» (**P**oint **O**f **S**ale) of the merchant.
Test account	A test account provided by Piraeus Bank to enable test transactions. It is made up of the same components as a «live account» but has different values.
Ticketing mechanism	A mechanism through which «ePOS Paycenter» is prepared for a new request.
Ticketing Web Service	A Piraeus Bank SOAP Web Service used for the mechanism.
Transaction ticket	An alpharithmetic value returned by the «Ticketing Web Service» necessary for the merchant system to authenticate the «Hash Key» received from «ePOS Paycenter» following the execution of a successful transaction.
ePOS Paycenter	The name of Piraeus Bank's e-payment system.
Failure page	The merchant site page receiving a response following the processing of an unsuccessful transaction.
Success page	The merchant site page receiving a response following the processing of a successful transaction.