# COMP3011 Coursework 2 Report

This report outlines the implementation of the payment service provider (PSP) API endpoints. The API endpoints slightly changed from what was planned during development. These changes are shown in the updated API diagram shown below:

A picture containing text, post-it note, handwriting, screenshot

Description automatically generated

The following sections describe the functionality and testing of my implementation. The API endpoints are hosted at <http://sc20asb.pythonanywhere.com>. The libraires which must be installed for this application are listed within requirements.txt.

## Error handling

All errors are handled by the errorHandling function, in which error codes correspond to simple comments which describe the issue that has been faced. The error codes for this API are as follows:

codes = {

100: 'Request body empty.',

101: 'Request body in incorrect format',

102: 'Could not find field "{}".',

103: 'Field "{}" is a {}, when {} was expected.',

104: 'Invalid Field "{}"',

105: 'Request type is not POST',

106: 'Payer card details could not be found',

107: 'Payee bank account details could not be found',

108: 'Payer personal account could not be found',

109: 'Payee business account details could not be found',

201: 'An error occurred with currency conversion.',

301: 'An error occurred with contacting the Payment Network Service.',

401: 'Could not make changes to database: {}',

402: 'Transaction with ID {} could not be located.',

403: 'Refund could not complete.',

404: 'Original transaction already refunded or cancelled.'

}

In the case that the PNS or currency exchange returns an error code and comment, this instead is returned.

## Initiate payment

This API endpoint handles the payment passed by the flight aggregator and is callable at <http://sc20asb.pythonanywhere.com/initiatePayment/>. This is done by:

* Checking the request type is POST.
* Validating that the body contains a JSON object.
* Ensuring that each field in the JSON request is in the correct format. This includes ensuring that the card number is valid using the Luhn algorithm, which is enforced for all major credit card numbers.
* The database is then queried to ensure that the account details match those of a registered user, as well as ensuring that the card details of the payer are also registered.
* The database is also queried to ensure that the payee has a registered business account.
* The currency conversion API is the called, which then converts the ‘Amount’ to the equivalent sum in Great British Pounds, as the payer or payee may be dealing in foreign currency.
* The RequestTransactionPNS is then called, which returns the transaction ID if successful. In this case, the transaction is then stored to the database. If there is an error, this is returned instead.

An example of an API request made to this endpoint in the correct format is shown below.

{

"CardNumber": "356938035643809",

"CVV": "696",

"Expiry": "2023-05-11",

"CardHolderName": "Adam",

"CardHolderAddress": "Leeds",

"Email": "adam@brown.com",

"PayeeBankAccNum": "78465912",

"PayeeBankSortCode": "124578",

"RecipientName": "Jet",

"Amount": 6.79,

"PayerCurrencyCode": "111",

"PayeeCurrencyCode": "111"

}

## Refund payment

This API endpoint handles refunding specific transactions and is available from: <http://sc20asb.pythonanywhere.com/initiaterefund/>. This is done by:

* Checking the request type is POST.
* Validating that the body contains a JSON object.
* Ensuring that each field in the JSON request is in the correct format and is valid (e.g. Amount being over 0.
* The database is then queried to ensure that the requested transaction is being stored within the database.
* The currency converter is then called to convert the amount being refunded into Great British Pounds.
* The PNS is then called to initiate the refund. If successful, the refund is stored as a transaction and a confirmation message is returned. If not, then an error message and corresponding comment is returned instead.

An example of an API request made to this endpoint in the correct format is shown below.

{

"TransactionUUID": "123",

"Amount": 19.00,

"CurrencyCode": "111"

}

## Cancel transaction

This API endpoint handles cancelling transactions. This is done by:

* Checking the request type is POST.
* Validating that the body contains a JSON object.
* Ensuring that each field in the JSON request is in the correct format.
* The database is then queried to ensure that the requested transaction is being stored within the database.
* The transaction is then labelled as cancelled in the database.

An example of an API request made to this endpoint in the correct format is shown below.

{

"TransactionUUID": "123"

}

## Testing

Local testing was done with the help of Postman. This included sending a mixture of valid, erroneous and boundary requests to each API endpoint. Once hosted on pythonanywhere, the application was then tested by the hosted PNS, currency converter, and flight aggregator services. This proved that this service could be successfully implemented within the wider structure of the group’s project.

This helped to expose inconsistencies with the formatting of JSON objects within the request bodies, as well as highlighting the importance of creating test data within each person’s corresponding database.