Payment Gateway Solution

Scalable
Scalability and isolation is the basic requirement of any payment gateway system, therefore implemented four endpoints (microservices) in the separate repositories. Each of the repositories could be deployed with a docker container to any container orchestration system.

Logging
The system also supports logging controlled by environment variables. Each logging for end-points is prefixed by different string for example the capture API
CAPTURE-API#2022/01/11 11:44:44 [][INFO] Connected to DB
Similarly for refund API it is like this
REFUND-API#2022/01/11 11:44:44 [][INFO] Connected to DB
It is very important in log analysis.

Source Code
Here is the code URL
https://github.com/syedomair/ex-paygate-approve for /authorize
https://github.com/syedomair/ex-paygate-void for /void
https://github.com/syedomair/ex-paygate-capture for /capture
https://github.com/syedomair/ex-paygate-refund for /refund

https://github.com/syedomair/ex-paygate-lib common code used by all the above repositories.

Interface
The Database and Payment Service is implemented through Interface. It allows you to change databases and payment services by changing the code in the handler. Also it is easier to test handler code with mock DB and mock Payment Service.
https://github.com/syedomair/ex-paygate-approve/blob/main/routes/approve/handler_test.go

Transaction Support
The code supports transactions, it ensures reliability.
https://github.com/syedomair/ex-paygate-refund/blob/main/routes/refund/repository_postgres.go#L77
https://github.com/syedomair/ex-paygate-capture/blob/main/routes/capture/repository_postgres.go#L76

Record level locking to prevent unwanted record update
https://github.com/syedomair/ex-paygate-capture/blob/main/routes/capture/repository_postgres.go#L79
https://github.com/syedomair/ex-paygate-refund/blob/main/routes/refund/repository_postgres.go#L80

Concurrent Code
Concurrent code improves the overall performance of the system.
https://github.com/syedomair/ex-paygate-refund/blob/main/routes/refund/repository_postgres.go#L91
https://github.com/syedomair/ex-paygate-capture/blob/main/routes/capture/repository_postgres.go#L110

Unit Test
https://github.com/syedomair/ex-paygate-refund/blob/main/routes/refund/handler_test.go
https://github.com/syedomair/ex-paygate-capture/blob/main/routes/capture/handler_test.go
https://github.com/syedomair/ex-paygate-void/blob/main/routes/void/handler_test.go
https://github.com/syedomair/ex-paygate-approve/blob/main/routes/approve/handler_test.go

Integration Test
This test is only possible when all the four api are running on different ports.
In the following Makefile there is a target for integration test
https://github.com/syedomair/ex-paygate-lib/blob/master/Makefile#L7
It runs this standalone application (API Client) for testing
https://github.com/syedomair/ex-paygate-lib/blob/master/helper/test_integration/test_integration.go

Concurrent API calls to test DB Locking and transactions.
https://github.com/syedomair/ex-paygate-lib/blob/master/helper/test_integration/test_integration.go#L101

API Doc
I have created an api doc yaml, you can open it by going to
https://editor.swagger.io/ -> File -> Import file.
You should be able to see API Documentation.
https://github.com/syedomair/ex-paygate-lib/tree/master/APIDoc

Middleware
We can add more middleware as per requirements.
https://github.com/syedomair/ex-paygate-approve/blob/main/routes/router.go#L86

Luhn Check
Implement Luhn Check
https://github.com/syedomair/ex-paygate-approve/blob/main/routes/approve/handler.go#L79

Edge Error Case
Implemented Here
https://github.com/syedomair/ex-paygate-approve/blob/main/routes/approve/payment_service.go#L29
https://github.com/syedomair/ex-paygate-capture/blob/main/routes/capture/payment_service.go#L31
https://github.com/syedomair/ex-paygate-refund/blob/main/routes/refund/payment_service.go#L31

Merchant Authentication
https://github.com/syedomair/ex-paygate-approve/blob/main/routes/approve/handler.go#L71

Project Setup

Database Create
DB and table create statement is in this folder
https://github.com/syedomair/ex-paygate-lib/tree/master/database

Each Makefile requires a .env file in the repo root.

Content of the ex-paygate-approve .env file

SERVICE_NAME=APPROVE-API
LOG_LEVEL=DEBUG
PORT=8321
SIGNINGKEY=wYu8P1HYJGJYJHTY
DATABASE_URL=postgres://<username>:<password>@127.0.0.1:5432/expaygate
DATABASE_URL_DOCKER=postgres://<username>:<password>@172.17.0.1:5432/expaygate

Content of the ex-paygate-void .env file

SERVICE_NAME=VOID-API
LOG_LEVEL=DEBUG
PORT=8322
SIGNINGKEY=wYu8P1HYJGJYJHTY
DATABASE_URL=postgres://<username>:<password>@127.0.0.1:5432/expaygate
DATABASE_URL_DOCKER=postgres://<username>:<password>@172.17.0.1:5432/expaygate
Content of the ex-paygate-capture .env file

SERVICE_NAME=CAPTURE-API
LOG_LEVEL=DEBUG
PORT=8323
SIGNINGKEY=wYu8P1HYJGJYJHTY
DATABASE_URL=postgres://<username>:<password>@127.0.0.1:5432/expaygate
DATABASE_URL_DOCKER=postgres://<username>:<password>@172.17.0.1:5432/expaygate

Content of the ex-paygate-refund .env file

SERVICE_NAME=REFUND-API
LOG_LEVEL=DEBUG
PORT=8324
SIGNINGKEY=wYu8P1HYJGJYJHTY
DATABASE_URL=postgres://<username>:<password>@127.0.0.1:5432/expaygate
DATABASE_URL_DOCKER=postgres://<username>:<password>@172.17.0.1:5432/expaygate