

nteraface>> Database paseRequest): IResponse

## <<Interface>> IRequestHandler

- + setNextHandler(handler: IRequestHandler) + handleRequest(request: IServerRequest): IResponse

## Firewall - serverManager : ServerManager + Firewall() - checkConnection() : boolean + execute(request : IServerRequest) : IResponse

<<Enumer Database

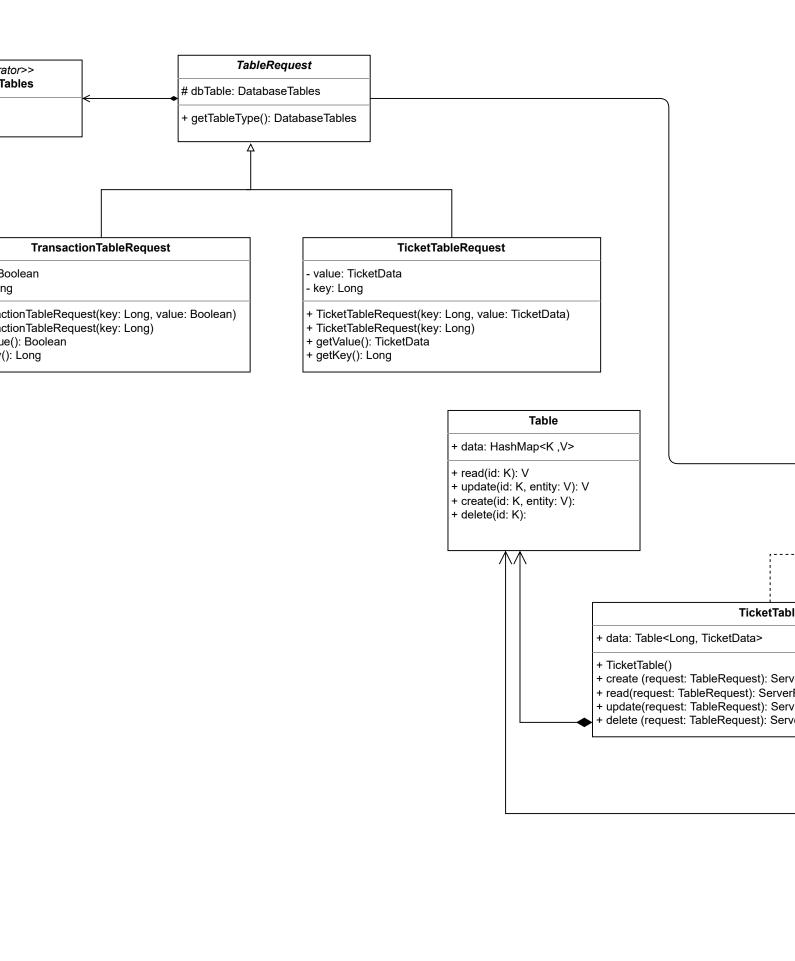
TicketTable Transaction

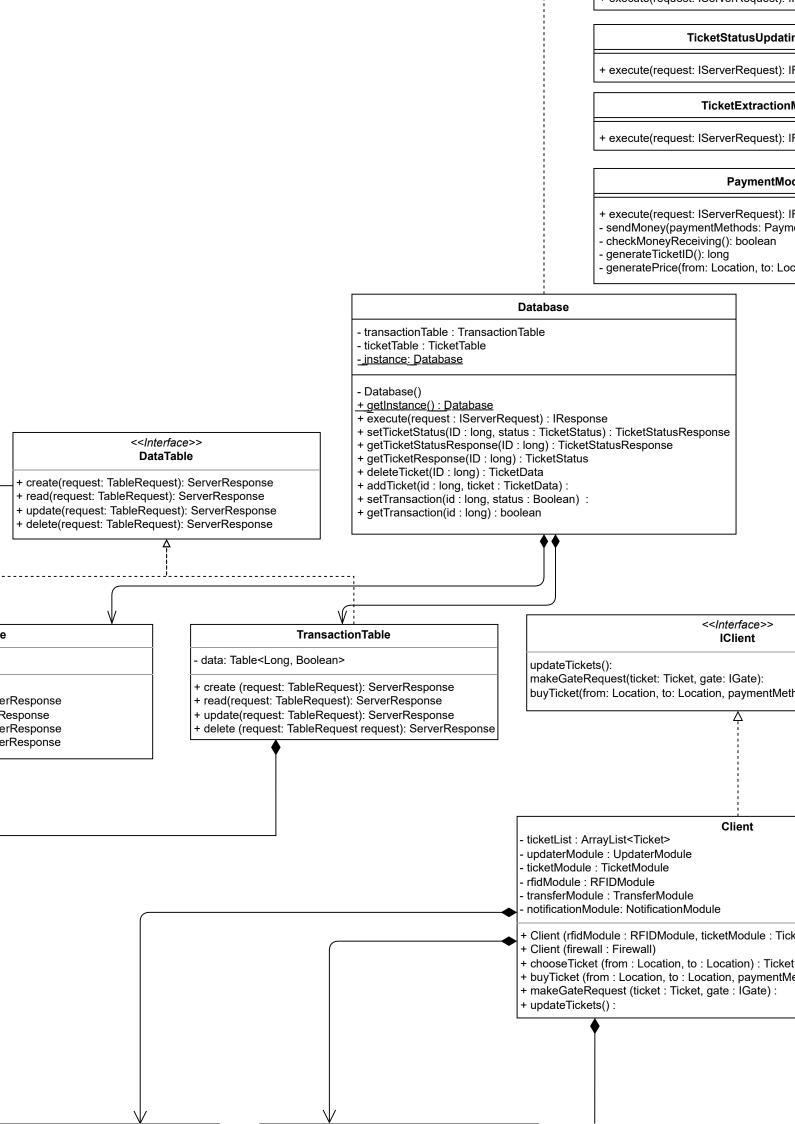
- value: I

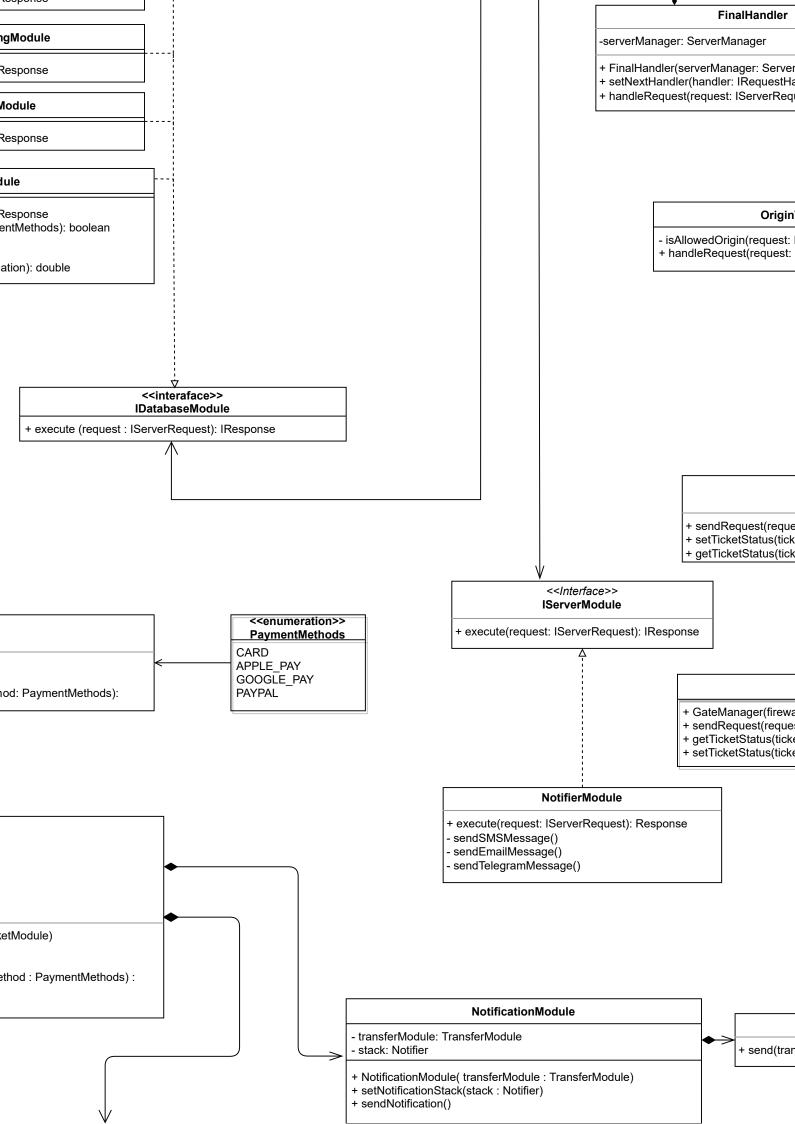
- key: Lo

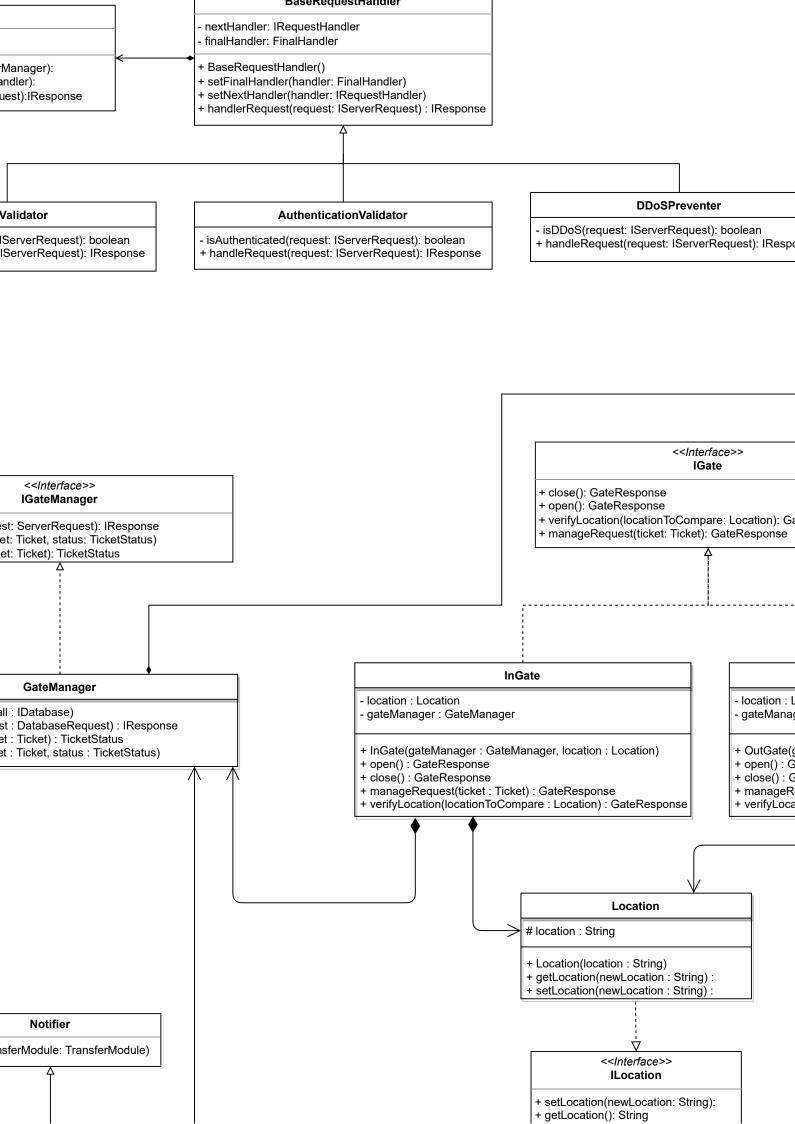
+ Transa

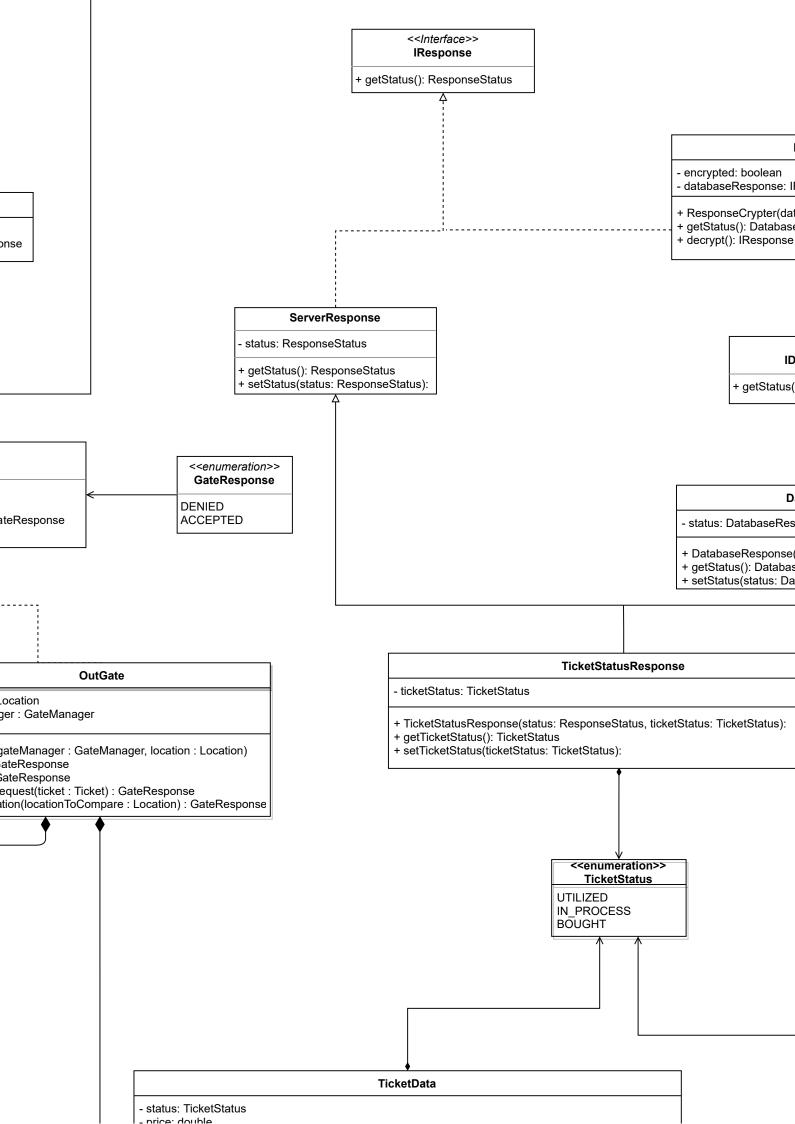
+ Transa + getVal + getKey

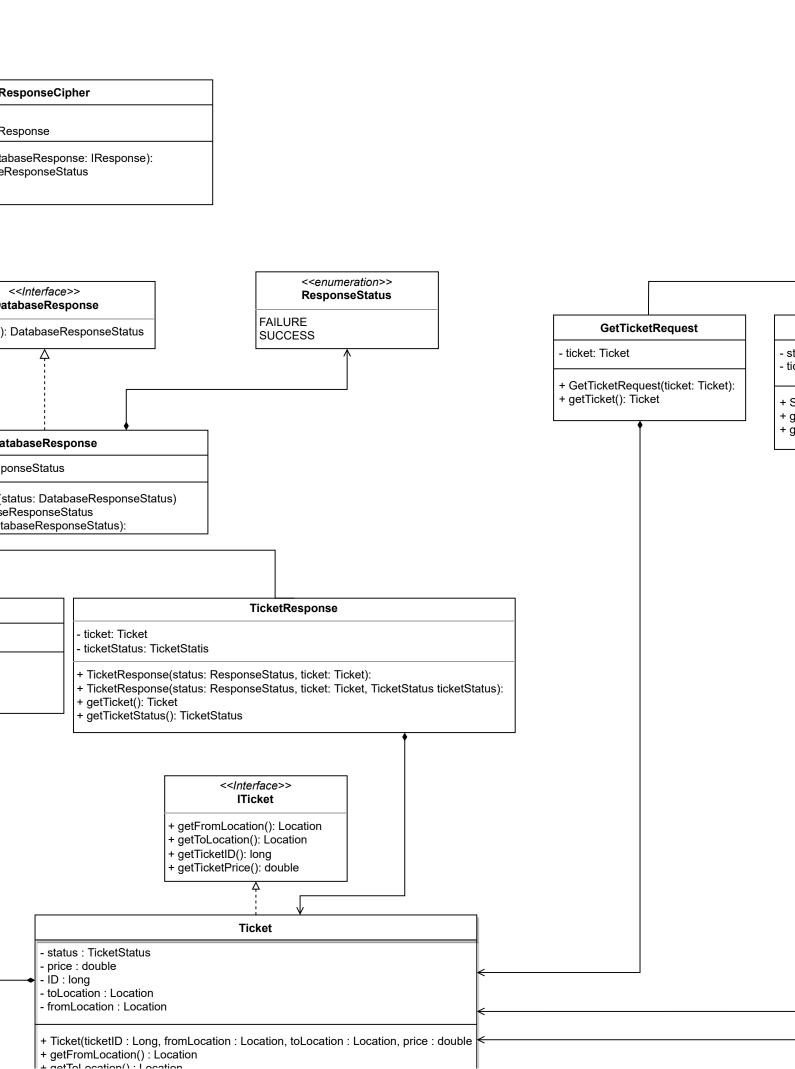


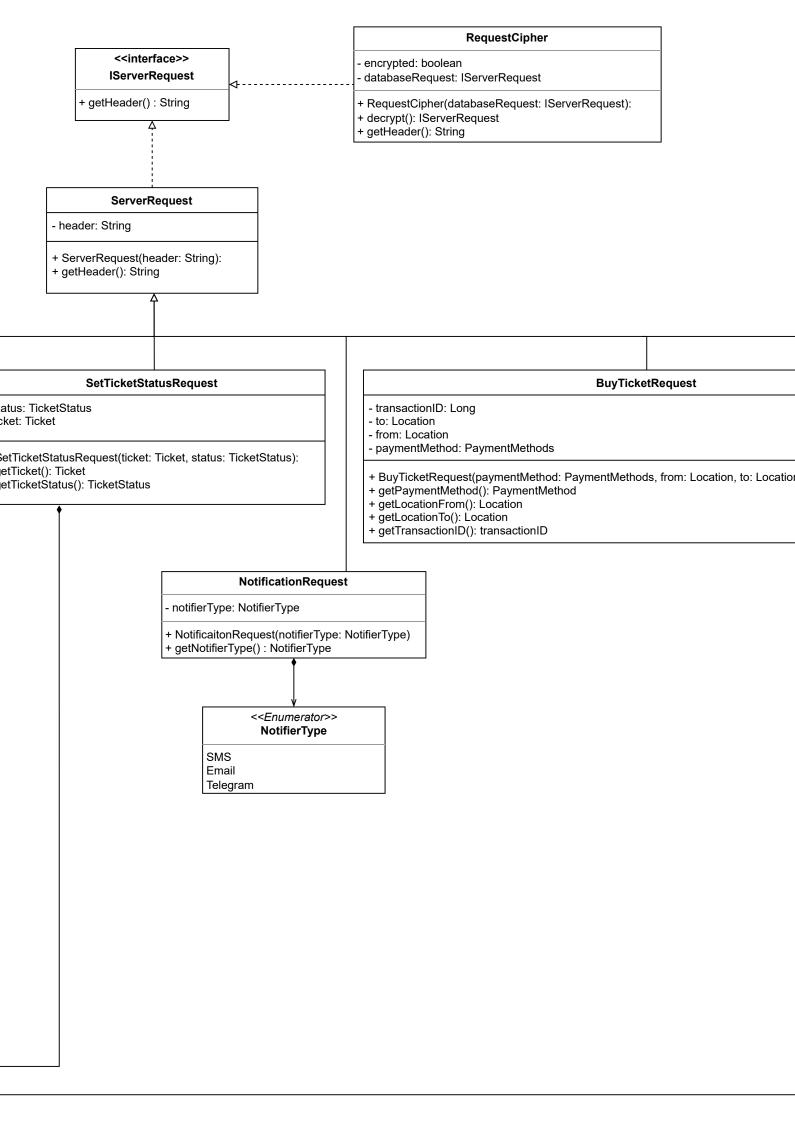


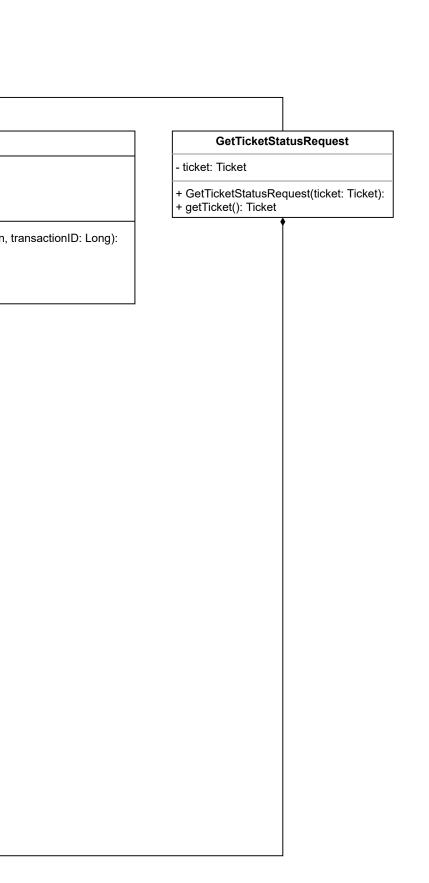












- id : UUID {readO

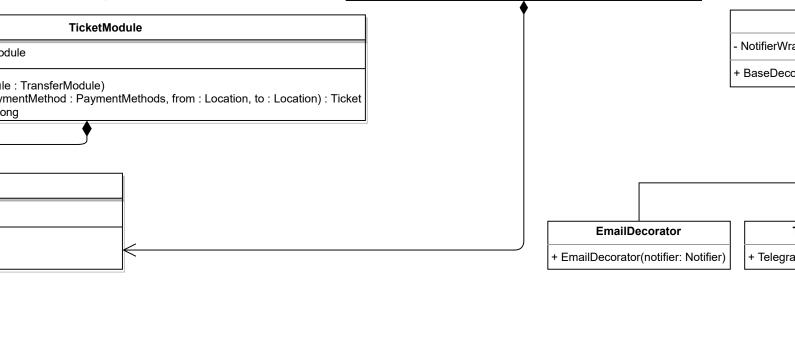
+ RFIDModule() + sendRequest(tic

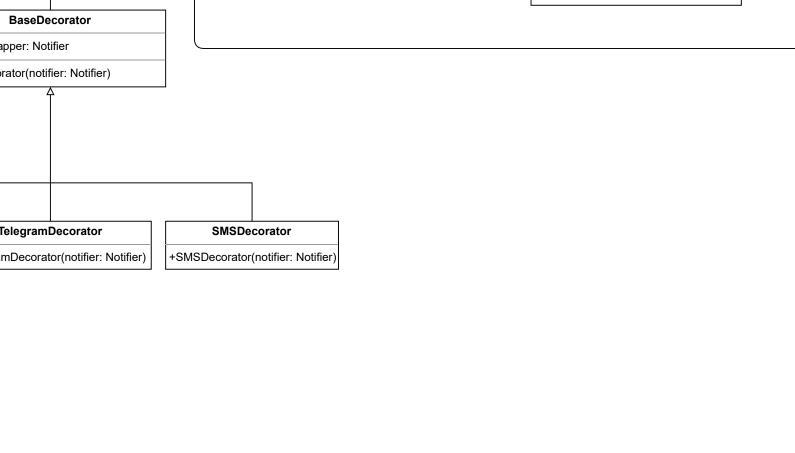
	UpdaterModule			
	- transferModule : TransferModule			
-	+ UpdaterModule (transferModule : TransferModule)			- transferModule : TransferMo
	+ getTicket (ticket : Ticket) : Ticker			+ TicketModule(transferModu
	•			+ sendBuyTicketRequest(pay - generateTransactionID() : L
		_		
		<u>\</u> \	/	$\bigvee$
			Trans	sferModule
	- fireWall : IDatabase			
	+ TransferModule(firewall : + getTicket() : Ticket	: IDatabase	;)	

**RFIDModule** 

ket : Ticket, gate : IGate) : GateResponse

nly}





- price, double - ID: long
- toLocation: Location - fromLocation: Location
- + TickerData(fromLocation: Location, toLocation: Location, ID: long, price: double, status: TicketStatus) + TicketData(ticket: Ticket): + setLocationTo(to: Location): + setLocationFrom(from: Location):

- + setPrice(price: Double): + setStatus(status: TicketStatus):
- + setID(id: Long): + getLocationTo(): Location + getLocationFrom(): Location
- + getPrice(): Double + getStatus():TicketStatus + getID(): Long

+ getTicketID(): long + getTicketPrice(): double + getTicketStatus(): TicketStatus