Payment Gateway:

-> The PaymentApplication class contains the incoming Request data, then the Validation methods are called, and appropriate error messages are displayed if the input is not valid. Then the data is encrypted and stored in a hashMap and the processPayment() from paymentProcessing class gets the processRequest queue and the necessary data required for submitting the payment request. On successful execution “Credit/debit card transactions request submitted” / “Mobile Wallet transaction request submitted” message is displayed.

PaymentApplication class:

* This class can receive 2 kind of payment Requests: Credit/Debit card request or mobile Wallet transaction, this class accepts the data and sends it to Validation class for checking if the incoming requests details are valid or not.
* **Security**: This class encrypts all the incomingRequest details using SHA encryption algorithm
* **Logging**: Used log4j to make a note of exception and errors that occur during the program execution
* Data Structures Used: I am using **HashMap**, **Array List** and **Queue** data Structures. HashMap to store the encrypted, actual value (key, value) pairs. And second HashMap stores the encrypted UserId as key, username, amount, card, mobile number as values.

Queue to store all the incoming payment Requests.

ValidatingData class:

* **Validations**: isValid() Checks if the given credit/debit card Number is valid/ or not
* isValidExpiryDate() This function checks if the expiry date is later than the present date
* isValidMobileNum(): This function checks if the mobile Number entered is a valid United States Number

PaymentProcessing Class:

* **Concurrency**: I am using **locks** to allow one process to processed at a time.
* The userId is matched with the existing list of users to make sure we are getting a valid request and then the details are sent to the submitRequest method, depending on the payment Mode overloaded method submitPayment() is called.