Credit card processing System:

1. Introduction
   1.1 Purpose of this Document
      To provide insights on credit card processing system's requirements and attributes and budget.
   1.2 Scope of this document The document provides a comprehensive discription of the CCP and it must be completed in 30 days
   1.3 Overveiw
      The system allows business to accept credit card payment from customers.

2. General Description
   Payment authorization: The customer should enter valid credentials
   -Transaction submission: The site must securely connect with the respective bank status notification. The business site must be able to send SMS to registered mobile.

3. Functional requirement
   The credit card processing system must be implemented as shown and described in the general description

   - Maintain record history.
   - Process multiple transaction for efficiency.
   - authenticate transaction.

4. Interface requirement:
The payment authorization must have text field for every ~~tree~~ detail of the credit card.
Transaction submission must have a simple buffering page.
Status notification must display a static page saying it has forwarded.
The interface must use React framework

5. Performance Requirements
The system should not exceed 4 secs of ~~response time and~~ must be able to run ~~response time~~ on 8 gb and 256 gb ram and rom system.

6. Design constraints
- The system must use only DFS algorithm for detecting the valid users in the database and it must use only Node js 11.5 version.

7. Non-functional Attributes.
The system must work only an apple macos, it must be compatible for all macos sessions and must use a DES encryption standard.

8. Preliminary Schedule and Budget
The project must be completed in 1 month of time and a total of 500 dollars is to be alloted to it.

Credit card system:

CardHolder
- CardHolder ID
- Name
- Address
- applyCard()
- updateCard()

Credit card
- card no
- expiration date
- cvv
- credit limit
- makePayment()
- checkBalance()

Standard

Student

Business

Transaction
- transaction ID
- amount
- date
- status
- processTransaction()
- getDetails()

Merchant
- merchant ID
- name
- category
- registerMerchant()
- processRegistration()

Authorization:
- auth code
- fee
- #Inhibit1
- status
- generate.code()
- validate auth.code()

Payment Gateway
- gatewayID
- name
- fee
- authtransaction()
- processRefund()

Account Statement
- Statement ID
- period
- Total amount
- generateStatement()
- getStatement()

«enumeration»
Payment Gateway
- RazorPay
- PhonePay
- AmazonPay

AccountDetails
- Time Period

owns owns has generates

given passthrough

belongs to

CardHolder

+CardHolderID
+Name
+Address

+applyorCard()
+updateDetails()

CreditCard

+cardno
+expirationdate
+CVV
+creditLimit

+makePayment()
+chckBalance()

standard

STudent

Business

+owns   +Owns   +has   1   0..*

+performs   0..*

+blongsto   1

Transaction

+transactionID
+amount
+date
+status

+processTransaction()
+getDetails()

+DoneBy   0..*

Merchant

+merchantID
+name
+category

+registerMerchant()
+processRegistraion()

+nenerates   0..*

1   1   +has   1   +passthrough   1   +given   1   +belongsto   1

Authorization

+auth.code
+date
+status

+generate.code()
+validate.auth.code()

PayementGateway

+gatewayIP
+name
+fee
+Attribute1

+auth.transaction()
+processRefund()

AccountStatement

+StatementID
+preiod
+totalamount

+generateStatement()
+ViewStatement()

«enumeration»
PayemntGateway

Razorpay
PonePay
AmazonPay

AccountDetails

TimePeriod