**Features:**

System Subscription

Might use the core database or new database installation but accounting system need to be set properly if system handles share

**Story of Interna accounting system**

Subscribed Service to provide in house accounting system to invoice products and and handle payment and cancellation

**Story of Customer Portal and mobile service**

Side portal for selling products and payments and orders [late payments]

Able to login and

**subModules:**

ordering [by external portal or internal system]

product handling [internal]

Payment and

Entity Management

Invoice management [Internal System]

---unread

**accounting:**

1-in entity \*->card relationship entityType Must be person/organization

2-bank defines fee for each card

3-account balance needs to be updated by database trigger each time new record inserted

4-invoice payee/payer entity need to be of entityType{person/org}

5-in entity/Card relation entity must be of entityType {person/org}

**Invoice and Services >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

1-all the services attached to invboice need to be in the same currency as invoice

2-all services added to the invoice need to the follow invoice\_payeeID =service\_giverID

***create Services***

entities including in service need to be of entityType{}

***create invoice***

invoiceAction

invoiceActionTransaction

***create service[s] and assign to invoice***

***payInvoice***

invoicePayment

invoiceAction

invoiceActionTransaction

**CARDS and BANKING >>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

***create bank***

***assign card to the bank***

***create fee and assign to card***

**Test Scenario >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**(1)Invoice and payments**

**1-reset database and seeds , 2-setup basic entities**

2-create initial lookups

3-create currency

4-create 2 persons

5-attach accounts to persons

6-Create 2 services

7-Creeat Invoice

8-assign Services to invoice

9-Finalize Invoice

10-Create Cards and assign to users

11-assign cards to persons

12-to partial payments for invoices

**(2)Payment Cancellation**

cancel payments w/o applying fees > everything will be reversed

Note: if fees need to be applied then transactions related to fees will be added.

**Scenario 1 plus cancel payments**

**Scenario[3]**

create card

create bank

create card Type

bank assign[Replace/New] fee to card Type

***Note:***

1-payments can be done for any type of actions like deposit or invoice.

2-track of original payment transactions will be kept in paymentTransaction Table.

3-any status change happened to the payment of any kind, e\will be recorded into separate category like invoice/invoiceAction/invoiceActionTransaction or deposit/depositAction/depositActionTRansaction

4-payment cancellation happens thru invoice and invoice calls payment Cancellation and saves another log for recorded transactions and action happened.

5-in polymorphism if a method is defined virtual-override, even if method is called in base class, will bubble up to inherited class. which is fine but tricky

1-

service issuer = invoice issuer = payee

service receiver = invoice receiver = payer

2-

after any operation , total balance of account table remains ZERO

3-

payment-action & payment-Action-Transaction tables can also be effected by invoice or deposit operations.

*Restrictions*:

1-payer can pay more than what exists in wallet money

2-there is no fee for internal payment

3-only fee for [bank/db-card] and [bank/cc-Card-Type] and

4-if [bank/cc-Card-Type] is defined then overrides [bank/card-Type]

5-fee and cc-Fee need to be unique per bank-card-Type for now