National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

Practical assignment. 7 types of UML diagrams

“Postrelation database”

Performed by:

Bohdan Kalika

TM-01mp

Reviewed by:

Iryna Mikhaylova

Kyiv – 2020

**1. Diagram of precedents**

@startuml

left to right direction

actor "customer" as user

actor "admin" as admin

rectangle Product as "Products page" {

usecase "Get" as get\_products

usecase "Add to a cart" as add

usecase "Delete from the cart" as delete\_product

usecase "See details" as details

}

user --> get\_products

user --> add

user --> delete\_product

user --> details

rectangle Orders as "Cart page" {

usecase "Get" as get

usecase "Buy" as buy

usecase "Delete" as delete

}

user --> get

user --> buy

user --> delete

rectangle Admin as "Admin page" {

usecase "Get products" as aget

usecase "Add a product" as aadd

usecase "Delete a product" as adelete

usecase "Change price" as change\_p

}

' admin page:

aget <-- admin

aadd <-- admin

adelete <-- admin

change\_p <-- admin

' products page

get\_products <-- admin

add <-- admin

delete\_product <-- admin

details <-- admin

' cart page

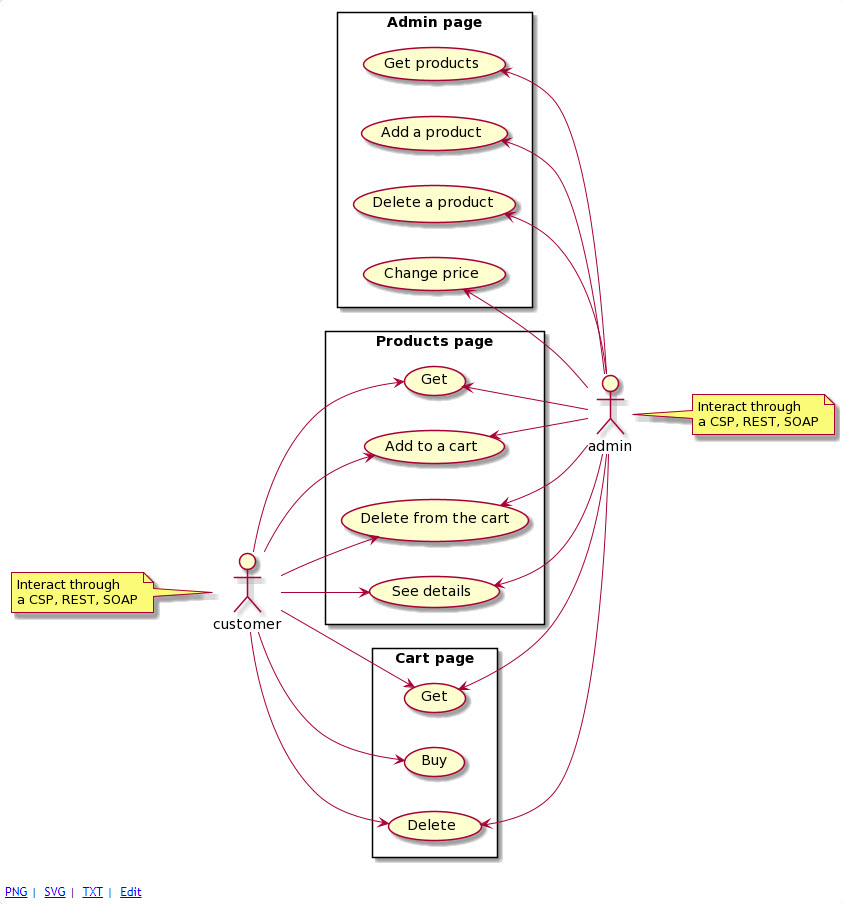
get <-- admin

delete <-- admin

note left of user : Interact through \na CSP, REST, SOAP

note right of admin : Interact through \na CSP, REST, SOAP

@enduml



**2. Class diagram**

@startuml

class "%Persistent" as persistent

class "%Library.Populate" as library

class "%XML.Adaptor" as adaptor

class "%SOAP.WebService" as web\_service

class "Product" as site {

String name

int price

String description

ArrayList : Array[] images

get()

create()

delete()

update()

}

class "SoapHandler" as soap\_handler

class "broker" as broker{

AcceptsContentType()

AccessCheck()

DispatchMap()

DispatchRequest()

Error()

Login()

}

persistent <|-- site

library <|-- site

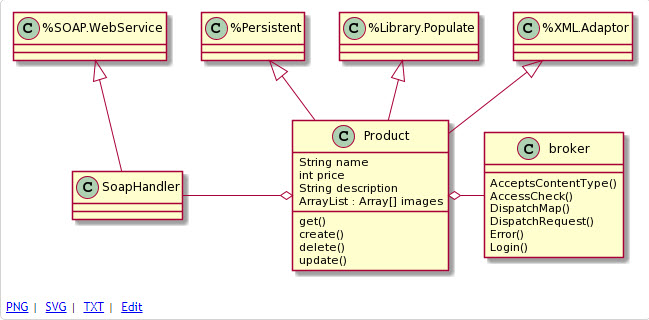
adaptor <|-- site

web\_service <|-- soap\_handler

soap\_handler -o site

site o- broker

@enduml



**3. Diagram of objects**

@startuml

object "Broker" as broker{

XData = </test>

}

object "UrlMap: XData" as map{

<Route Url="/csp/user" Method="GET" Call="GetAll"/>

<Route Url="/csp/user" Method="POST" Call="Creat"/>

<Route Url="/csp/user/:id" Method="DELETE" Call="Delete"/>

<Route Url="/csp/user/:id" Method="PUT" Call="Update"/>

}

broker - map

object "Site" as site{

ID=50

Name="Samsung"

Price=19999

Desctiption="Some desc"

}

object "SoapHandler" as soap{

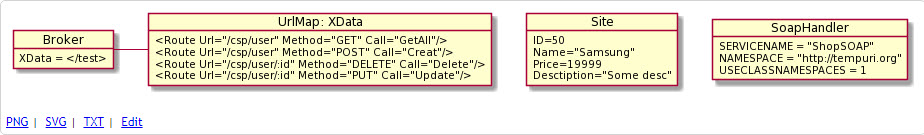
SERVICENAME = "ShopSOAP"

NAMESPACE = "http://tempuri.org"

USECLASSNAMESPACES = 1

}

@enduml



**4. Diagram of packages**

@startuml

package "/csp/user" as site {

package "shop.csp" {

package "Class Product.cls"

package "Class Client.cls"

package "Class Order.cls"

}

package ProductSOAP as soap {

package "Class BrokerSOAP.cls"

package "Class SoapHandler.cls"

}

package ProductREST as rest {

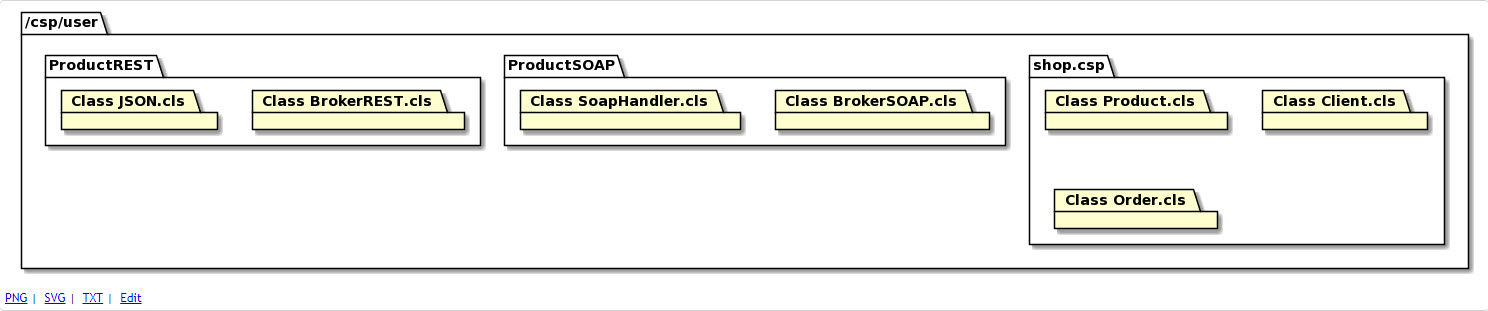
package "Class BrokerREST.cls"

package "Class JSON.cls"

}

}

@enduml



**5. Deployment diagram**

@startuml

node "IRIS Cache instance" as API{

}

node "Server" as server {

[IRIS.User = Entity.classes]

[IRIS.User = HTTPHandler.classes]

[IRIS.User = CSP.classes]

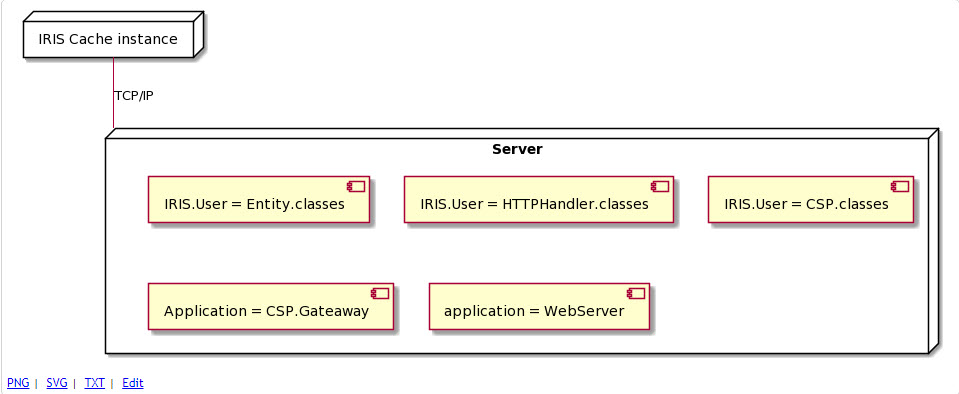
[Application = CSP.Gateaway]

[application = WebServer]

}

API -- server : TCP/IP

@enduml



**6. Activity chart**

@startuml

(\*) --> "HTTP request"

if "Is the CSP request?" then

--> [true] "Sending a static content"

--> (\*)

else

--> "Dispatching to a session"

if "Is a class or static?" then

--> "OnPage Method"

-> "Forming a response"

else

--> [static] "Finding a static\nfile and sendign back to the client"

endif

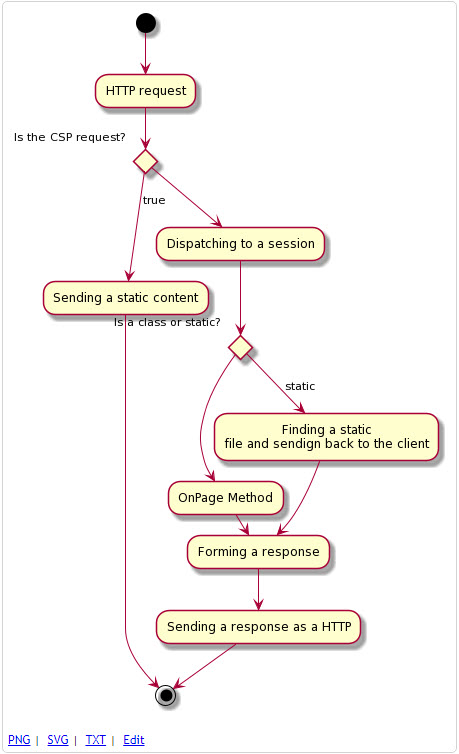
--> "Forming a response"

--> "Sending a response as a HTTP"

--> (\*)

endif

@enduml



**7. Sequence diagram**

@startuml

participant "User" as user

participant "Filter" as filter

participant "User client" as user\_client

participant "CSP Gateway" as gateway

participant "CSP page" as csp

participant "REST Broker" as rest

participant "SoapHandler" as soap

activate user

user -> filter : HTTP request

activate filter

filter -> gateway : Sending as Gateway

deactivate filter

activate gateway

alt csp

gateway -> csp : Dispatching a shop.csp

activate csp

csp -> csp : Request processing

csp -> csp : Forming a response

csp --> gateway : response

deactivate csp

else rest

gateway -> rest : Dispatching a RESTBroker.cls

activate rest

rest -> rest : Request processing

rest -> rest : Forming a response

rest --> gateway : response

deactivate rest

else soap

gateway -> soap : Dispatching SoapHandler.cls

activate soap

soap -> soap : Request processing

soap -> soap : Forming a response

soap --> gateway : response

deactivate soap

end

gateway --> user : Sending a HTTP response

deactivate gateway

deactivate user

@enduml

