# ServiceLayer

**MessageController**
- my_context : ApplicationContext
- smt : SimpMessageTemplate
+ emails_to_sockSeassion_map : HashMap<String,String>

notification_handler

**NotificationHandler**
- context : ApplicationContext
- users_notifications : Map<String, List<String>>

0..1

«interface»
**Iservice**

**Service**
- dis : DataInputStream
- dos : DataInputStream
- s : Socket

Use    Use

**MarketSystem**
- external_system_url : String

MarketFace

# DomainLayer

**AppointmentInformation**
- member_email : String
- appointer_email : String
- type : String

**BidInformation**
- price : double
- quantity : int
- status : String

**StoreManagerInfo**
- store_name : String

product_info

**CartInformation**
- founder_email : String

**ProductInformation**
- product_id : int
- name : String
- category : String
- key_words : List<String>
- price : double
- quantity : int
- store_id : int

**StoreInformation**
- store_id : int
- founder_email : String
- name : String
- foundation_date : String

Use

## FacadeModule

<<Interface>>
**IFacade**

Implements

**MarketFacade**
- loggedUser : int
- isGuest : boolean

payment_adapter

supply_adapter

## ExternSystems

<<Interface>>
**PaymentAdapter**
+ payment() : boolean
+ connect_to_payment_system() : boolean

**PaymentAdapterImpl**

<<Interface>>
**SupplyAdapter**
+ supply() : boolean
+ connect_to_supply_system(): boolean

**SupplyAdapterImpl**

store_controller

user_controller

StoreReview

ProductReview

## StoreModule

frame — Policy Reference

frame
Purchase Policy Reference

frame
Discount Policy Reference

frame
Bidding Reference

**StoreController**
- store_ids_counter : AtomicInteger
- purchase_ids_counter : AtomicInteger
- storesLock : Object

«interface»
**IStore**

Implements

store_review

**StoreReview**
- avg_rating : int
- rating : Map<String, Integer>

**Store**
- store_id : int
- founder_email : String
- name : String
- foundation_date : String
- active : boolean
- product_ids_counter : AtomicInteger

founder
owner
manager

inventory

**Product**
- store_id : int
- product_id : int
- name : String
- category : String
- key_words : List<String>
- price : double

productReview

**ProductReview**
- reviews : Map<String, String>
- rating : Map<String, Integer>

**Basket**
- basket_id : Pair<String,Integer>
- total_price : double

## Appointment

<< enum >>
**StorePermission**

<< enum >>
**StoreManagerType**

**Appointment**
- member_email : String
- appointer_email : String
- store_id : int

baskets

stuff_and_appointment

**Cart**

## Purchase

**StorePurchaseHistory**
- store_name : String

**UserPurchaseHistory**

HistoryList

**StorePurchase**
- buyer_email : String
- purchase_id : int

**UserPurchase**
- purchase_id : int
- total_price : double

Purchase

store_id_purchase

**Purchase**
- transaction_date : LocalDateTime
- product_and_quantity : Map<int,int>
- product_and_totalPrice : Map<int,double>
- product_and_name : Map<int,String>

store_purchase_history

userPurchaseHistory

## UserModule

**UserController**
- isGuest : AtomicBoolean
- isLogged: AtomicBoolean
- usersLock : Object

onlineUsers    Users

**User**
- isGuest : AtomicBoolean
- isLogged : AtomicBoolean

state

<< Abstract >>
**AssignState**

Extends

**AssignUser**
- email : String
- name : String
- lastName : String

security

**Security**
- password: String

Extends

**PremiumSecurity**
- question: String
- answer: String

**Admin**

Extends

**Guest**

StatisticsManager

## Statistics

«interface»
**IstatisticsManager**

implement

**StatisticManager**
- init_time : LocalDateTime
- login_count : AtomicInteger
- logout_count : AtomicInteger
- connect_system_count : AtomicInteger
- register_count : AtomicInteger
- buy_cart_count : AtomicInteger

create

**Statistic**
- init_system_time: LocalDateTime
- login_per_minutes: long
- logout_per_minutes : long
- connect_per_minutes : long
- register_per_minutes : long
- buy_cart_per_minutes : long

Use

## Utils

<U,V>
**Pair**
+ first : U
+ second : V

**Utils**

<< Singleton >>
**ErrorLogger**
+ logger : Logger
+ FileHandler : handler

<< Singleton >>
**SystemLogger**
+ logger : Logger
+ FileHandler : handler

<T>
**Response**
+ value : T
+ wasException: boolean
+ message : String

## Communication

<>
**Question**
# question_id : int
# message_date : LocalDateTime
# answer_date : LocalDateTime
# message : String
# answer : String
# boolean : has_answer

Extends

**SystemQuestion**
- user_email: String

**BuyerQuestion**
- buyer_email : String
- store_id : String

**UserQuestion**
- user_email : String

system_to_user    buyer_to_store    user_to_admin

**QuestionHandler**
- questions_ids_counter : AtomicInteger

sender

Use

# Data Acess Layer

**HibernateUtils**
- em : EntityManager
- persistence_unit : String

Use

**DataBase**

1

# References

## Bidding Reference

<<Reference>>
**Store**

1

bids

*

«interface»
**iBid**

Implement

<<Reference>>
User

1

**Bid**
- quantity : int
- offer_price : double
- buyer_email : String
- negotiation_price : double

1

**BidManagerAnswer**
- has_answer : boolean
- answer : String

*

*

<<Reference>>
Product

1

*

1

<<*Enum*>>
**BidStatus**

## Policy Reference

<<Referenced>>
**Store**

1

discountPolicy

purchasePolicy

### Discount Policy Reference

1

**DiscountPolicy**
- id: Long

1

policy

left

1

**<<Abstract>>**
**DiscountComponent**
- id : Long

1

right

1

rule

Extends

**ComplexDiscountComponent**

predict

**<<Interface>>**
**Ipredict**

Extends

**CompositePredict**
left:Ipredict
right:Ipredict

Extends

**andCompsoitePredict**

**OrCompositePredict**

Extends

0..1

0..1

**maxDiscountComponent**
left:DiscountComponent
right:DiscountComponent

**xorDiscountComponent**
left:DiscountComponent
right:DiscountComponent

0..1

**plusDiscountComponent**
left:DiscountComponent
right:DiscountComponent

0..1

**simpleDiscountComponentByStore**

0..1

0..1

0..1

Extends

**simpleDiscountComponent**
double:percentOfDiscount

**simpleDiscountComponentByCategory**
CategoryOfDiscount:String

**simpleDiscountComponentByProduct**

*

ProductToDiscount

1

<<Reference>>
**Product**

### Purchase Policy Reference

1

**PurchasePolicy**
id:Long
policyId:Long

1

policy

*

**<<Abstract>>**
**PurchaseRule**

1

right

1

left

*

**SimplePurchaseRule**

**purchaseLogicComponent**

predict

**OrPurchaseRule**

**AndPurchaseRule**

1

**Predict**

category:String
product:Product
above:boolean
equel:boolean
num:int
price_constraint:boolean
quantity_constraint:boolean
age_constraint:boolean
time_constraint:boolean
category_constraint:boolean
product_constraint:boolean
year:int
month:int
day:int