

Model

Vegetable details of Order

OrderItem
Long orderId
String vegName
Double price
Integer quantity
Double eachVegPrice

Before confirmation of order

BillDetail
String vegName
int price
Double eachVegBill
<u>int quantity</u>

Order Details of a Customer

OrderDetails
Long orderId
String username
Double totalBill
String status
Boolean active
Timestamp createdAt
Date deliveryDate
String feedback
List<OrderItem> orderItems
String paymentMethod

User Module

UserDetail
String name
int age
String gender
String email
String username
String password
String role

Vegetables Available in shop

VegDetail
String vegName
int price
<u>int quantity</u>

OrderDetails contains OrderItem



## Service

### Service related to sales

#### *SalesService*

```
getBill( String[] selectedVegs, String[] quantities) -> List<BillDetail>
getTotalBill(List<BillDetail> billDetails) - > Double
getDateTime() -> String
getNewStockQuantity(OrderItem vegetable) - > Integer
storeOrderDetails(String username, List<BillDetail> billDetails, String paymentMethod, String deliveryDateStr)
getAllOrderDetails() - > List<OrderDetail>
storeOrderItems(List<BillDetail> billDetails, Long orderId)
getCurrentDeliveryOrders() - > List<OrderDetail>
setStatusAsDelivered(Long orderId) -> Boolean
setStatusAsCanceled(Long orderId -> Boolean
```

### Service related to user details

#### *UserService*

```
addUser(UserDetail user)
loginValidation(String username, String password -> String
removeAccount(String username)
changeEmailToLowerCase(UserDetail user) -> UserDetail
updateName(String newName, String username)
updateMobileNumber(long newMobileNumber, String username)
updateEmail(String newEmail, String username)
getAllUsers() -> List<UserDetail>
```

### Service related to shop vegetables

#### *VegetableService*

```
addVeg(VegDetail vegetable)
removeVeg(String vegName)
setVegNameLowercase(VegDetail vegetable) -> VegDetail
getVegDetails() -> List<VegDetail>
```

## Validator

### Used to validate Bill

#### *BillValidator*

isBillValid(List<BillDetail> billDetails)

### Used to validate login

#### *UserLoginValidator*

userValidator(String username, String password) -> Boolean

### Used to validate user details

#### *UserValidator*

isUserValid(UserDetail user) -> Boolean  
isUserNotRepeated(UserDetail user) -> Boolean  
isUsernamePresent(String username) -> Boolean  
isMobileNotRepeated(long mobileNumber) -> Boolean  
isEmailNotRepeated(String email) -> Boolean  
isUpdateNameValid(String newName, String username) -> Boolean  
isUpdateMobileValid(long newNumber, String username) -> Boolean  
isUpdateEmailValid(String newEmail, String username) -> Boolean

### Used to validate common

#### *UtilValidator*

boolean isStringValid(String name) -> Boolean  
isNumberValid(int number) -> Boolean  
isPasswordValid(String password) -> Boolean  
isEmailValid(String email) -> Boolean  
isUsernameValid(String username) -> Boolean  
isMobileValid(long mobileNumber) -> Boolean

### Used to validate vegetable

Used to validate vegetable

***VegetableValidator***

isVegMatches(String vegName) -> Boolean

isVegNotRepeated(String vegName) -> Boolean

## ***Exceptions***

### ***User Defined Exceptions***

DBException

EmptyBillException

InvalidLoginException

InvalidQuantityException

InvalidSelectionException

UserInvalidException

UserRepeatedException

VegInvalidException


## DAO

### ***SalesDetailsDAO***

```
saveOrderItems(long orderId, OrderItem vegetable)
saveOrderDetails(OrderDetail order)
findOrderId(OrderDetail order) -> Long
findAllOrderDetails() -> List<OrderDetail>
findOrderItemsById(Long orderId) -> List<OrderItem>
findOrdersForDelivery(Date date, String status) -> List<OrderDetail>
updateStatus(String status, Long orderId) -> Boolean
```

### ***UserDetailDAO***

```
addUser(UserDetail user)
getUserDetails() -> List<UserDetail>
removeUser(String username)
updateName(String newName, String username)
updateMobile(long newNumber, String username)
updateEmail(String newEmail, String username)
getRole(String username, String password) -> String
```

### ***VegDetailDAO***

```
saveVeg(VegDetail vegetable)
findAll() -> List<VegDetail>
deleteVegByName(String name)
findPriceByName(String name)
findStockByName(String name)
updateStock(String name, int newStock)
```

## ***DTO***

### ***BillDetailsDTO***

getBillForVeg(String[] selectedVeg, String[] quantities, List<Double> eachVegBill) -> List<BillDetail>

### ***SalesDetailDTO***

setOrderDetailsForOutput(OrderDetail orderDetail, List<OrderItem> orderItems) -> OrderDetail

setOrderItems(List<BillDetail> vegDetails) -> List<OrderItem>

setOrderDetailsForDb(String username, double totalBill, Timestamp createdDateTime, Date deliveryDate, String paymentMethod) -> OrderDetail



etail

## ***Servlet***

### ***List Of Servlets***

AddVegetableServlet  
CanceledServlet  
DeliveredServlet  
DeliveryManServlet  
GetVegDetailsServlet  
LoginValidationServlet  
LogoutServlet  
OrderConfirmServlet  
RemoveAccountServlet  
RemoveVegetableServlet  
SalesDetailsServlet  
SelectVegetablesServlet  
SelfRemoveAccountServlet  
UpdateEmailServlet  
UpdateMobileServlet  
UpdateNameServlet  
UserRegistrationServlet  
ViewUsersServlet

## ***JSP***

### ***List Of JSP***

addVegetable  
BillPage  
DeliveryDetails  
DeliveryManIndex  
editProfile  
header  
index  
loginPage  
newUserRegistration  
OrderConfirmedPage  
removeAccount  
removeVegetable  
SelectVegetables  
ViewSalesDetails  
viewUsers