Canteen Food Ordering and Management System

Jeffrey Ong, Shan Shan Lee, Khang Phan, Neil Remolar

CMPE 131-01

Group 14

San Jose State University

Backlog

User stories	Detailed requirement	Implement (code)

US1
View food categories
	R1.1
	To view food choices
	R1.2
	To choose food and drinks
		FoodCategories viewFoodCategories(){
		Return foodCategories;
		}

US2
Order food
	R2.1
	Food is available
	R2.2
	Quantity ordered is available
	R2.3
	To add food to cart
		Void orderFood(Food food){
		cart.add(food);
		}

US3
Proceed to checkout
	R3.1
	There is food being ordered
	R3.2
	To make payment
		Void makePayment(){
		User.money - price;
		}

US4
View again the ordered food list
	R4.1
	To check what is in cart
		Cart checkOrder(){
		Return cart;
		}

US5
Choose payment method
	R5.1
	There is food already ordered
	R5.2
	Amount to pay is greater than 0
	R5.3
	To be able to check out
		Void paymentMethod(String method){
		if(credit card){
		creditCardPayment - price;
		}
		
		if(debit card){
		debitCardPayment - price;
		}
		
		if(student card){
		studentCardPayment - price;
		}
		}

US6
Track ordered food status using tracking number
	R6.1
	There is food ordered to be checked
	R6.2
	racking number exists
		String Tracking(){
		if(Tracking ==true){
		Return checked;
		} else
		Return Null;
		}

User stories	Detailed requirement	Implement (code)

US7
View order received
	R7.1
	There is food ordered
	R7.2
	Tracking number exists
	R7.3
	View order status
		``
		String Tracking(){
		if(checked == true){
		Return Tracking;}
		else
		Return Null;
		}
		``

US8
Send out status of food to customer with their tracking number
	R8.1
	Tracking number exists
	R8.2
	Status exists
	R8.3
	Send out order status
		``
		String Tracking(){
		if(Tracking == true){
		Return StatusOder;}
		else
		Return Null;
		}
		``

US9
Inventory management
	R9.1
	Check if inventory is up to date
	R9.2
	Make sure nothing runs out
	R9.3
	Add item in inventory
	R9.4
	Remove item from inventory
		``
		Boolean checkInventoryFull(String Item){
		if(Item.length <
		item.getMaximumLength()){
		Return false ;
		else
		return true;
		}
		
		Void addInventory(Item i){
		Item.add(i);
		}
		
		Void removeInventory(Item i){
		Item.remove(i);
		}
		``

US10
Setting up food categories
	R10.1
	Food categories is empty, if not, choose update instead of set up
		``
		Void setUpFoodCategories(){
		if(FoodCategories.isEmpty()){
		FoodCategories =new ArrayList<Food>();}
		}
		``

US11
Update food categories
	R11.1
	Food categories is available
	R11.2
	Add food into category
		``
		Void updateFoodCategories(Food f){
		FoodCategories.add(f);
		}
		``