UNIVERSITY OF LIVERPOOL

COMP201: Software Engineering I

Object Oriented Design

Coursework Assignment 1

Requirement Engineering

Ziheng Zhang – 201220030 – x6zz

10 November 2016

Task 1

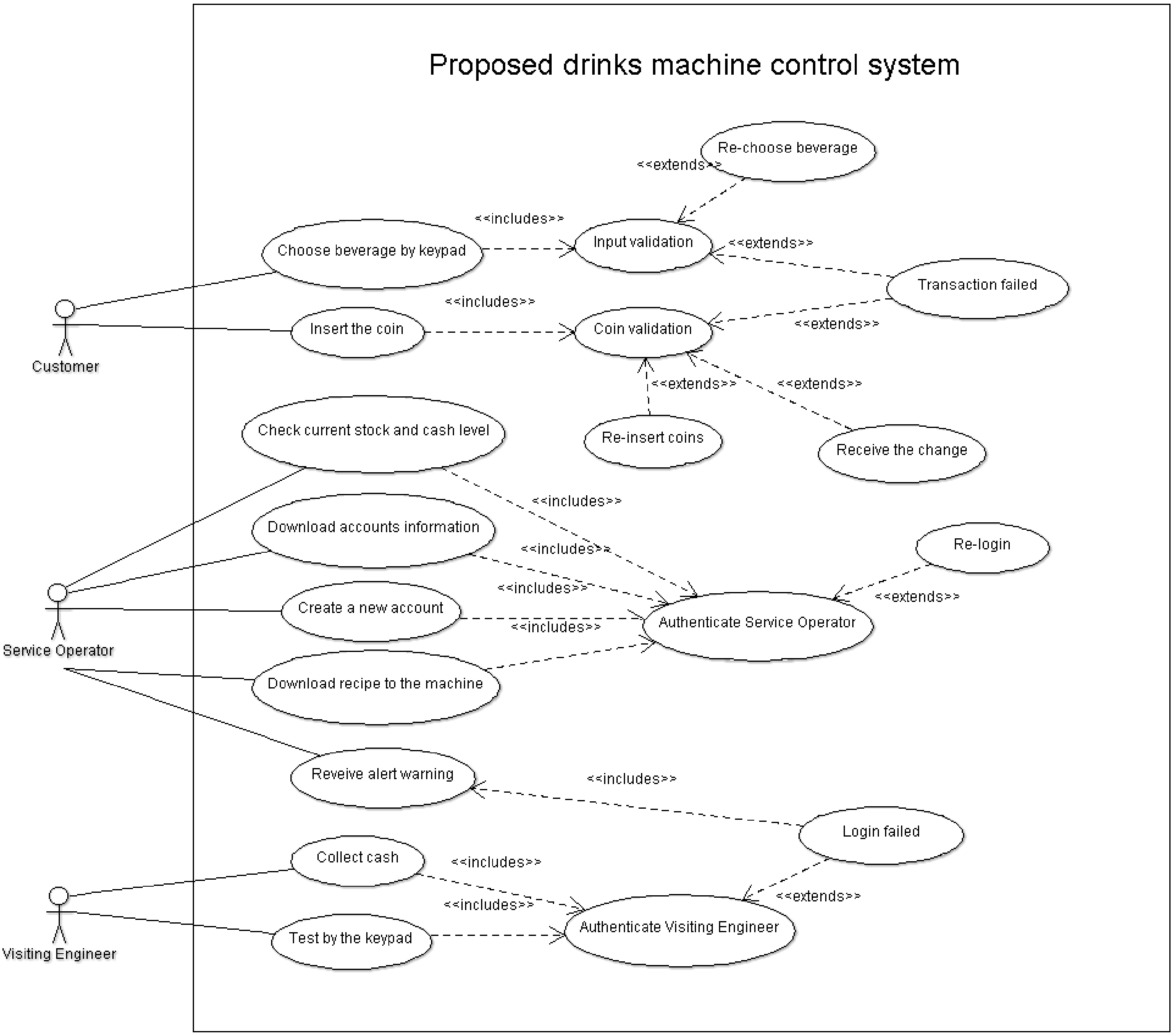
All tasks for this assignment refer to the given scenario “Proposed Drink machine system” (overleaf on page 2).

Produce a **UML use-case model** (i.e., **BOTH** a use-case diagram and use-case descriptions) and identify as many actors as you can in your model that are within the scope of the system. For the use-case diagram part of the model, you may use any method to draw it, including a hand-drawn diagram or ArgoUML software (available on the departmental computers (Click start and then type Argouml into the search box) or for free download via the internet) for example. The demonstrators will be able to help you with using this program. There is also app.genmymodel.com this is easy, free and online (for public projects), so very good if you are not in the lab. For the model diagram if you find it difficult keeping it looking good on 1 diagram, feel free to split it into multiple diagrams. This is encouraged if it has become difficult to read. Keep all text easy to read and all fonts at least 14pt.

For use cases please include the following: name of use-case, name of actor involved, pre-conditions and possible extension use cases

Solution:

Use-case diagram is shown as below:


Proposed drinks machine control system
Re-choose beverage
<<extends>>
<<includes>>
Choose beverage by keypad
Input validation
<<extends>>
Transaction failed
<<includes>>
Insert the coin
Coin validation
<<extends>>
Customer
<<extends>>
<<extends>>
Check current stock and cash level
Re-insert coins
Receive the change
<<includes>>
Download accounts information
<<includes>>
Re-login
<<includes>>
<<extends>>
Create a new account
Authenticate Service Operator
<<includes>>
Service Operator
Download recipe to the machine
Reveive alert warning
<<includes>>
Login failed
Collect cash
<<includes>>
<<includes>>
<<extends>>
Authenticate Visiting Engineer
Test by the keypad
Visiting Engineer

Use-case descriptions are listed as below:

ID	UC1
Name	Choose beverage by keypad
Description	Customer choose the hot beverage with the keypad
Actor involved	Customer
Pre-conditions	Drinks machine in service
Event flow	1. Include Use Case 10 'Input validation' 2. Browse all the hot beverages 3. Chosen drinks is available for purchasing 4. Finish purchasing
Extension	
Trigger	
Post-condition	

ID	UC2
Name	Insert the coin
Description	Customer insert coins into drinking machine
Actor involved	Customer
Pre-conditions	Drinks machine in service
Event flow	1. Include Use Case 11 ‘Coin validation’ 2. Customer can choose to insert coins of different values
Extension	
Trigger	
Post-condition	

ID	UC3
Name	Check current stock and cash level
Description	Service operator checks current stock and cash level from machine
Actor involved	Service Operator
Pre-conditions	Drinks machine connected online
Event flow	1. Include Use Case 16 ‘Authenticate Service Operator’ 2. Check status of current recipe level 3. Check current cash level of each value
Extension	
Trigger	
Post-condition	

ID	UC4
Name	Download accounts information
Description	Service operator downloads accounts information from machine
Actor involved	Service Operator
Pre-conditions	Drinks machine connected online
Event flow	1. Include Use Case 16 ‘Authenticate Service Operator’ 2. Download username and password of accounts
Extension	
Trigger	
Post-condition	

ID	UC5
Name	Create a new account
Description	Service operator creates a new account
Actor involved	Service Operator
Pre-conditions	Drinks machine connected online
Event flow	1. Include Use Case 16 ‘Authenticate Service Operator’ 2. Download username and password of accounts
Extension	
Trigger	
Post-condition	

ID	UC6
Name	Download recipe to the machine
Description	Service operator download recipe to the machine
Actor involved	Service Operator
Pre-conditions	Drinks machine connected online
Event flow	1. Include Use Case 16 ‘Authenticate Service Operator’ 2. Download specific recipe to the machine
Extension	
Trigger	
Post-condition	

ID	UC7
Name	Receive alert warning
Description	Service operator receives alert warning from alarm mode
Actor involved	Service Operator
Pre-conditions	Use Case 19 'Login failed'
Event flow	1. Visiting engineer open the back of machine but does not login 2. Service operator receives an alert warning
Extension	
Trigger	Visiting engineer fails to login the drinking machine system
Post-condition	

ID	UC8
Name	Collect cash
Description	Visiting engineer collects cash from drinking machine
Actor involved	Visiting Engineer
Pre-conditions	Drinks machine in service
Event flow	1. Include Use Case 18 ‘Authenticate Visiting Engineer’ 2. Collect remaining cash, including coins and notes, from the machine
Extension	
Trigger	
Post-condition	

ID	UC9
Name	Test by the keypad
Description	Visiting engineer test drinking machine by the keypad
Actor involved	Visiting Engineer
Pre-conditions	Drinks machine in service
Event flow	1. Include Use Case 18 ‘Authenticate Visiting Engineer’ 2. Test drinking machine by input from the keypad
Extension	
Trigger	
Post-condition	

ID	UC10
Name	Input Validation
Description	Drinking machine system validate customer input
Actor involved	Customer
Pre-conditions	Use Case 1 'Choose beverage by keypad'
Event flow	1. If input already validated exit from use case 2. Customer inputs by the keypad 3. Drinking machine validates input and respond
Extension	Use Case 12 'Re-choose beverage' Use Case 13 'Transaction failed'
Trigger	Validation service requested and input from the keypad not validated
Post-condition	Input is validated and process continues

ID	UC11
Name	Coin validation
Description	Drinking machine system validate inserted coins
Actor involved	Customer
Pre-conditions	Use Case 2 'Insert the coin'
Event flow	1. If coins already validated exit from use case 2. Customer inserts coins 3. Drinking machine validates coins and respond
Extension	Use Case 14 'Re-insert coins' Use Case 13 'Transaction failed' Use Case 15 'Receive the change'
Trigger	Validation service requested and inserted coins not validated
Post-condition	Coins are validated and process continues

ID	UC12
Name	Re-choose beverage
Description	Customer re-chooses hot beverage
Actor involved	Customer
Pre-conditions	Use Case 10 'Input validation'
Event flow	1. Customer inputs by the keypad and is rejected 2. Customer re-inputs by the keypad
Extension	
Trigger	Last input from customer has been validated wrong and rejected
Post-condition	

ID	UC13
Name	Transaction failed
Description	Transaction fails due to either wrong input or rejected coins
Actor involved	Customer
Pre-conditions	Use Case 10 'Input validation' Use Case 11 'Coin validation'
Event flow	1. Customer input is validated wrong 2. Inserted coins are validated counterfeit 3. Drinking machine terminates transaction
Extension	
Trigger	Customer input validated wrong and inserted coins validated counterfeit
Post-condition	Transaction fails and drinking machine restart to next transaction

ID	UC14
Name	Re-insert coins
Description	Customer re-inserts coins since last was rejected
Actor involved	Customer
Pre-conditions	Use Case 11 'Coin validation'
Event flow	1. Inserting coins last time was rejected 2. Re-insert coins to purchase drink
Extension	
Trigger	Coins that was inserted last time was rejected
Post-condition	

ID	UC15
Name	Receive the change
Description	Customer receives the change once transaction done
Actor involved	Customer
Pre-conditions	Use Case 11 'Coin validation'
Event flow	1. Customer inputs correctly and insert coins 2. Transaction was done and drink was offered 3. Customer receives the change he deserved
Extension	
Trigger	Transaction done and inserted coin’ value over drinks’ value
Post-condition	Transaction finished

ID	UC16
Name	Authenticate Service Operator
Description	Drinking machine authenticates service operator
Actor involved	Service Operator
Pre-conditions	Online system in service
Event flow	1. If service operator authenticated exit from use case 2. Service operator inputs username and password 3. Service operator re-inputs username and password if login fails
Extension	Use Case 17 'Re-login'
Trigger	Authentication service requested and service operator not authenticated
Post-condition	Service operator is authenticated

ID	UC17
Name	Re-login
Description	Service operator re-login the system
Actor involved	Service Operator
Pre-conditions	Use Case 16 'Authenticate Service Operator'
Event flow	1. Service operator login the system but fails 2. Service operator re-login the system with username and password
Extension	
Trigger	Authentication service requested and service operator not authenticated
Post-condition	

ID	UC18
Name	Authenticate Visiting Engineer
Description	Drinking machine authenticates visiting engineer
Actor involved	Visiting Engineer
Pre-conditions	Drinking machine in service
Event flow	1. If visiting engineer already authenticated exit from use case 2. Visiting engineer enters username and password 3. Service operator receives alert warning if login fails
Extension	Use Case 19 'Login failed'
Trigger	Authentication service requested and visiting engineer not authenticated
Post-condition	Visiting engineer is authenticated

ID	UC19
Name	Login failed
Description	Login of visiting engineer fails
Actor involved	Visiting Engineer
Pre-conditions	Use Case 18 'Authenticate Visiting Engineer'
Event flow	1. Include Use Case ‘Receive alert warning’ 2. Visiting engineer’s login fails 3. Drinking machine enters alarm mode 4. Service operator receives alert warning
Extension	
Trigger	Authentication service requested and visiting engineer not authenticated
Post-condition	Drinking machine enters alarm mode and service operator receives alert warning

Task 2

Identify five non-functional and verifiable requirements of the "Proposed drinks machine control system" below, using the description of the scenario (you can make some assumptions about the system not detailed in the requirement description). For the requirements propose a mechanism and appropriate criteria for making them verifiable. So, all the requirements you list need to have a technique to objectively test them.

Solution:

Verifiable non-functional requirements

1. The system shall always keep the internal environment clean and germ-free, by measuring internal *temperature (T)* and *air pressure (P)* in order to avoid any risk to cause customer ill. Measurement of internal temperature by thermometer and internal air pressure by barometer would test whether the internal environment is germ-free or not.

2. The drinking machine should be designed small and movable for convenience in terms of its *scale (area & height)* and *number of wheels (N)*. Measurement of are and height of drinking machine and number of wheel would provide a strong evidence to test whether it is movable enough.

3. Un-experienced customer shall choose their beverage without any difficulties from the keypad that contains *16 characters* (0123456789ABCDEF) corresponding to 16 different types of beverages. Test on that each beverage corresponds to one specific character can be conducted to prove the friendliness to customers.

4. The disposable cup shall protect customer from burn when it is full of hot beverage where *athermancy* (an inability to transmit radiant heat) of cups matters. Measurement of *temperature* on the outside of cups would prove its athermancy when cups are full of hot water.

5. The drinking machine system shall have a reasonable reliability which guarantees the machine works properly for a long time. Calculation of mean *frequency* to failure could be done through repeated machine test and the frequency below once a month can prove its reliability.

Task 3

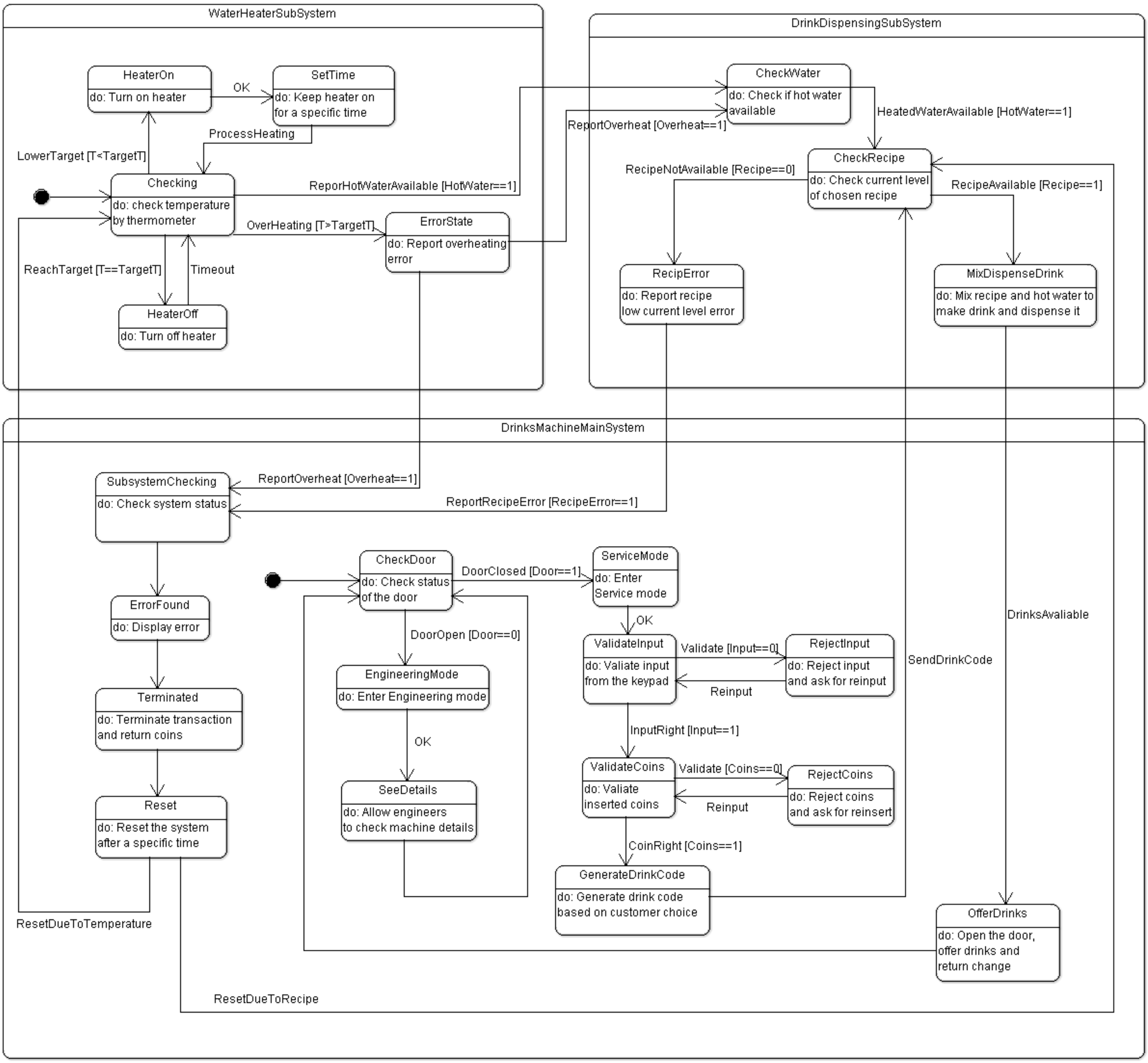
Design **three UML state machine diagrams** (also known as statechart diagrams) to show the various possible states of the following systems. These are totally different state diagrams and not sub-diagrams of on another. You can signal from one diagram to another by generator events it would understand. So for example to signal the water is too low to make a drink add the following Event("WaterTooCold").

Note you will have to make use of guard conditions.

Also note, you will need to have error states within your diagrams. Each system can send events to other systems, for example the hot water system can send a hot water ready event to the drinks dispensing subsystem to start of always list as many states as possible for each system

Solution:

UML state machine diagrams are displayed as below:


WaterHeaterSubSystem
HeaterOn
do: Turn on heater
OK
SetTime
do: Keep heater on for a specific time
ProcessHeating
LowerTarget [T<TargetT]
Checking
do: check temperature by thermometer
ReporHotWaterAvailable [HotWater==1]
OverHeating [T>TargetT]
ErrorState
do: Report overheating error
ReachTarget [T==TargetT]
Timeout
HeaterOff
do: Turn off heater
DrinkDispensingSubSystem
CheckWater
do: Check if hot water available
HeatedWaterAvailable [HotWater==1]
ReportOverheat [Overheat==1]
RecipeNotAvailable [Recipe==0]
CheckRecipe
do: Check current level of chosen recipe
RecipeAvailable [Recipe==1]
RecipError
do: Report recipe low current level error
MixDispenseDrink
do: Mix recipe and hot water to make drink and dispense it
DrinksMachineMainSystem
SubsystemChecking
do: Check system status
ReportOverheat [Overheat==1]
ReportRecipeError [RecipeError==1]
ErrorFound
do: Display error
Terminated
do: Terminate transaction and return coins
Reset
do: Reset the system after a specific time
ResetDueToTemperature
ResetDueToRecipe
CheckDoor
do: Check status of the door
DoorClosed [Door==1]
DoorOpen [Door==0]
EngineeringMode
do: Enter Engineering mode
OK
SeeDetails
do: Allow engineers to check machine details
ServiceMode
do: Enter Service mode
OK
ValidateInput
do: Valiate input from the keypad
Validate [Input==0]
RejectInput
do: Reject input and ask for reinput
Reinput
InputRight [Input==1]
ValidateCoins
do: Valiate inserted coins
Validate [Coins==0]
RejectCoins
do: Reject coins and ask for reinsert
Reinput
CoinRight [Coins==1]
GenerateDrinkCode
do: Generate drink code based on customer choice
SendDrinkCode
DrinksAvaliable
OfferDrinks
do: Open the door, offer drinks and return change