

Live It Rich

Algorithm

1. New booking of a room	5
2. Modification of a booking for a room	6
3. Delete a booking for a room	7
4. A guest arrives	8
5. New guest	9
6. Modification of a guest	10
7. Every morning	11
8. End of a journey	12
9. New room service order	13
10.Modification of an order	14
11.Cancellation of an order	15
12.New complaint	16
13.Extension of a stay	17
14.Shortening of a stay	18
15.A guest uses a product from the mini-bar	19
16.A guest wants to look at the planning	20
17.A guest wants to look at his bill	21
18.A guest forgot the code of his safe	22
19.Add a new compensation	23
20.End of day	24
21.Products delivered	25
22.New provider	26

23.New product	27
24.Modification of a product	28
25.Delete a product	29
26.New booking for a service	30
27.Modification of a booking for a service	31
28.Cancel a booking for a service	32
29.New request	33
30.New employee	34
31.Modification of an employee	35
32.Clocking in	36
33.New convention collective	37
34.Modification of a convention collective	38
35.New contract	39
36.Modification of a contract	40
37.Ten years	41
38.A staff member leaves	42
39.A player arrives	43
40.Modification of an information about a player	44
41.A player wants to see his score history	45
42.The game is finished	46
43.A receipt arrives	47
44.A provider delivers an equipment	48
45.A contractor comes to maintain the green	49
46.An equipment is broken	50
47.A guest arrives	51

48.A guest leaves	52
49.New ESpa care, care menu or product	53
50.Modification of an ESpa care, care menu or product	54
51.A receipt arrives	55
52.A contractor analyzes the pool	56
53.A provider delivers a product or an equipment	57
54.A guest wants to buy a product	58
55.New order	59
56.New beverage	60
57.Modification of a beverage	61
58.Delete a beverage	62
59.Receipt arrival	63
60.A guest arrives	64
61.New order	65
62.Modification of an order	66
63.Delete an order	67
64.An order is prepared	68
65.A guest leaves	69
66.New menu	70
67.Modification of a menu	71
68.Delete a menu	72
69.New wine	73
70.Modification of a wine	74
71.Delete a wine	75
72.New dish	76

73.Modification of a dish	77
74.Delete a dish	78
75.Ask an invoice	79
76.A guest no show	80
77.End of a day	81
78.The chef requests products	82
79.New receipt	83
80.A laundry request is ready	84
81.Start work	85
82.A bag has to be deposited	86
83.A maid arrives in a room	87

Legend

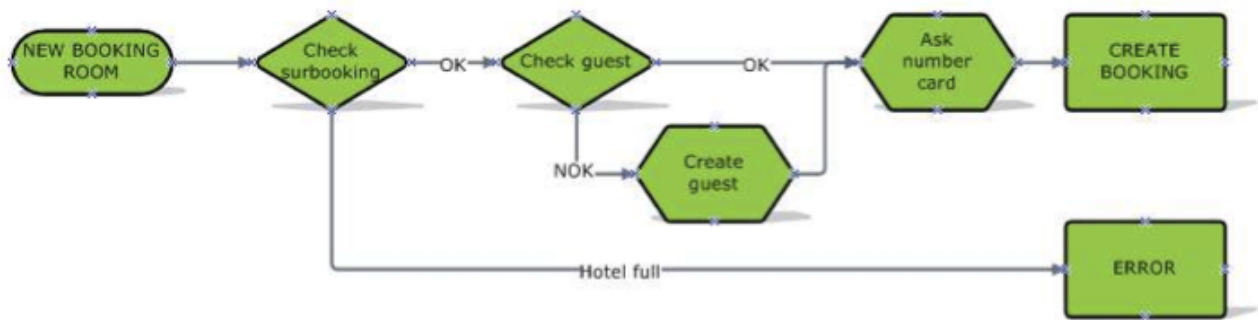
Human interaction



Informatics Treatment



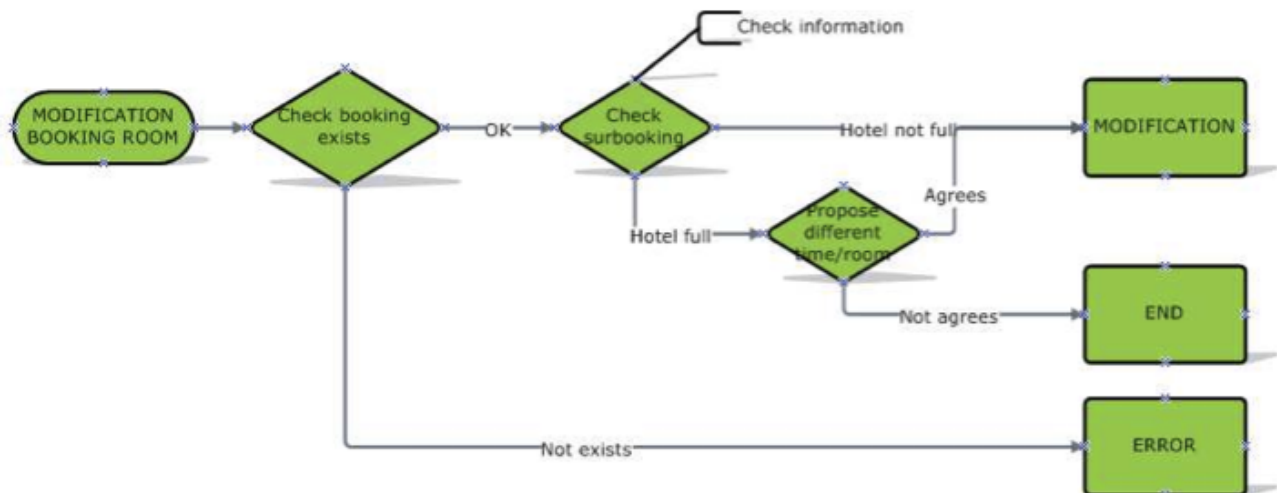
1. New booking of a room



@param RoomId, BookingDateOfBeginning, BookingDateOfEnd

- Ask **RoomTypeLabel**(*facultative*), **BookingDateOfBeginning**, **BookingDateOfEnd**
- Treatments of information and display the corresponds bookings
- Treatments of information and display the room availability
- Select the room line into the list box of room availability
- Click on the new booking button
- Recuperation of **RoomId** selected **BookingDateOfBeginning**, **BookingDateOfEnd**
- Calculate the **BookingNumberOfDays**
- Open New windows and display information selected
- Input disable button 'OK'
- Input **GuestName** and **GuestSurname** into the champ
- On change of Guest champ, *Call M7 (check guest existence)*
- If Guest not exist
 - Show the CreateGuest button
 - Click on the new guest button
 - *Call Mev5 (New Guest)*
 - Update the display guest information
- Select the guest line into the list box of guest
- Click on the credit card button
- Open new windows and display champ for the credit card's guest
- Input data into champ and click ok
- Credit card windows closed and transmit to new booking windows
- Enable the Ok button
- Click on the Ok button
- Input data into database :
 - **RoomId** ;
 - **GuestId** ;
 - **BookingDateOfBeginning** ;
 - **BookingDateOfEnd** ;
 - **BookingNumberOfDays** ;
 - **BookingNumberOfGuests** ;
 - **BookingDate** ;
 - **CreditCard** information.

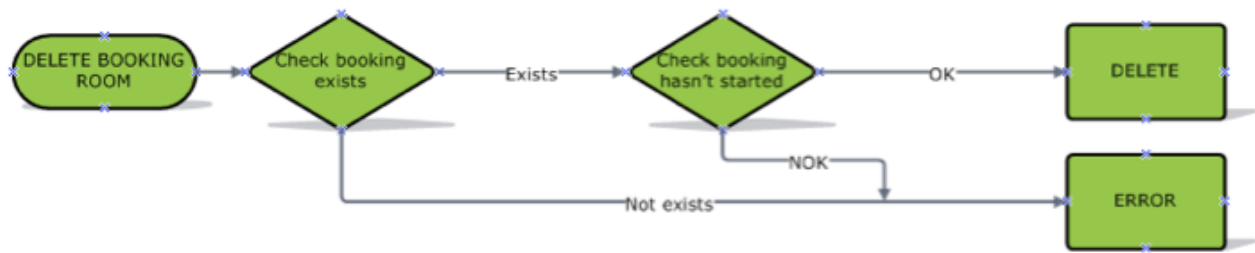
2. Modification of a booking for a room



@param GuestName, GuestSurname, BookingDateOfBeginning

- Ask **BookingDateOfBeginning**, **GuestName** and **GuestSurname**
- Treatments of information and display the corresponds bookings
- Select the booking line into the list box of booking
- Click on the modification booking button
- Recuperation of **BookingId** selected
- Display new screen
- Retrieve data of booking for the guest
- Input data of the database correspond of **BookingId** and **GuestId** into champs
- Champs surname and name are disabled
- Input disable button 'OK'
- Input data into champs
- Verification of room availability for the date asked
- Display the result of verification with pop up
- If verification is ok, enable the button 'OK'
- Click on the Ok button
- Update data of the database

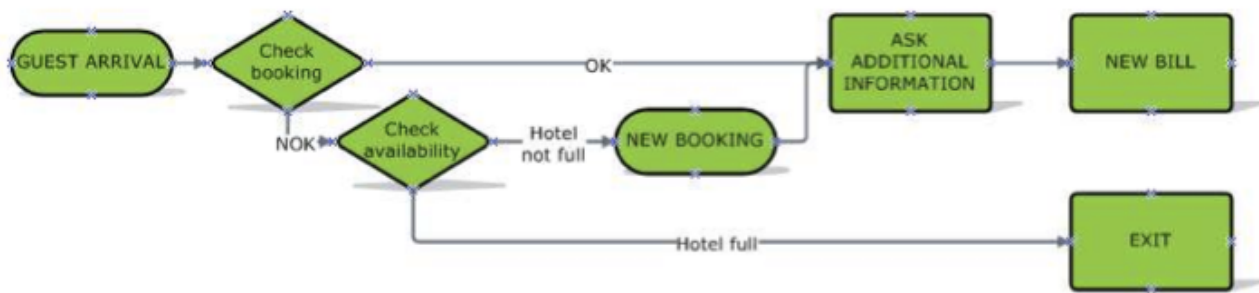
3. Delete a booking for a room



@param BookingDateOfBeginning, GuestName, GuestSurname

- Ask the **GuestName** and **GuestSurname**
- Enter the **GuestName** and **GuestSurname** into the database
- Display the list of guest(s)
- If the guest doesn't exist :
 - Error
- Else
 - Ask the **BookingDateOfBeginning**
 - Enter the **BookingDateOfBeginning** into the database
 - Display the list of corresponding booking
 - If the booking exists
 - If the **BookingDateOfBeginning** < dateToday :
 - If the button « delete booking » is clicked :
 - display a pop-up menu « Error »
 - Else
 - Delete the booking selected
 - Else
 - Error

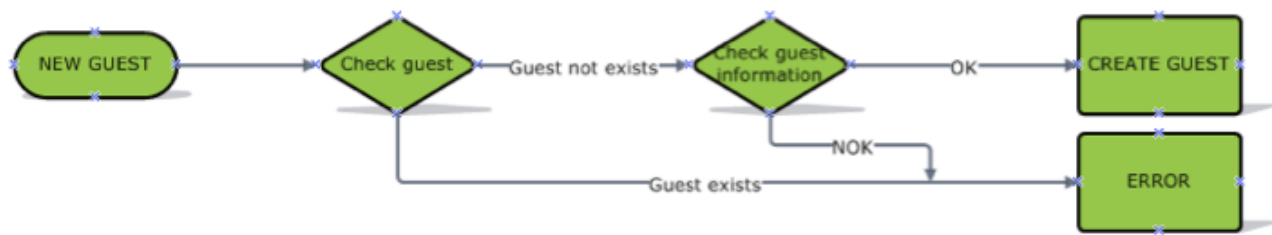
4. A guest arrives



@param GuestName, GuestSurname

- Ask **GuestName** and **GuestSurname**
- Treatments of information and display the corresponds bookings
- If there is correspondences into the list box of bookings
 - If there is informations additional
 - See the *Mev6 (Modification of a guest)*
 - Select the booking line into the listbox of booking
 - Click on the button arrivals
 - Recuperation of **BookingId** selected
 - Retrieve the **GuestId** for this booking
 - Input data into database :
 - **BillPaidOrNot**.
- Else see *Mev2 (New booking of a room)*

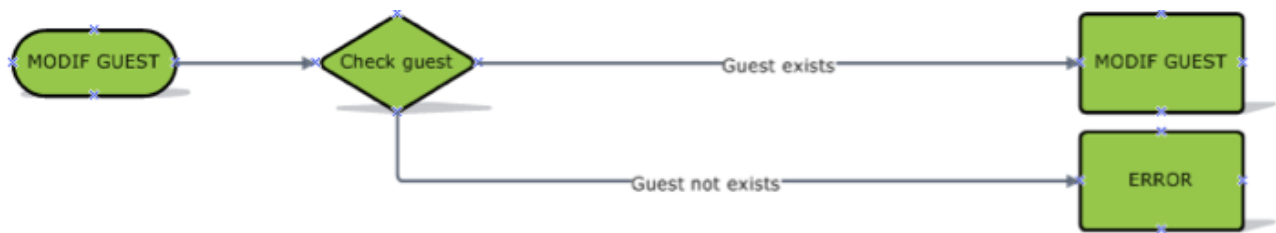
5. New guest



@param GuestName, GuestSurname

- Ask the **GuestName** and **GuestSurname**
- Enter the **GuestName** and **GuestSurname** into the fields
- Display the list of corresponding guest(s)
- If the guest doesn't exist :
 - Click on « new guest »
 - Ask the information about the new guest
 - Enter the information about the guest into the fields
 - Input the information into the database
- Else
 - Error

6. Modification of a guest



@param GuestName, GuestSurname

- Ask the **GuestName** and **GuestSurname** to the guest
- Enter the **GuestName** and the **GuestSurname** into the fields
- Display the list of corresponding guest
- If the guest doesn't exist :
 - Error
- Else
 - Ask the new information of the guest
 - Enter the new information into the fields
 - Replace the old information by the new

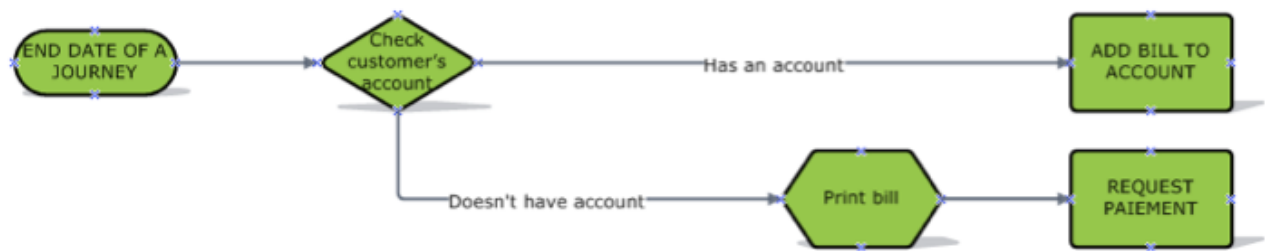
7. Every morning



@param BookingDateOfBeginning, GuestId

- Display all the booking
- For each booking
 - If the **BookingDateOfBeginning** – DateToday =1 :
 - Recover the corresponding **GuestId**
 - See Mev8
 - Else
 - Next booking

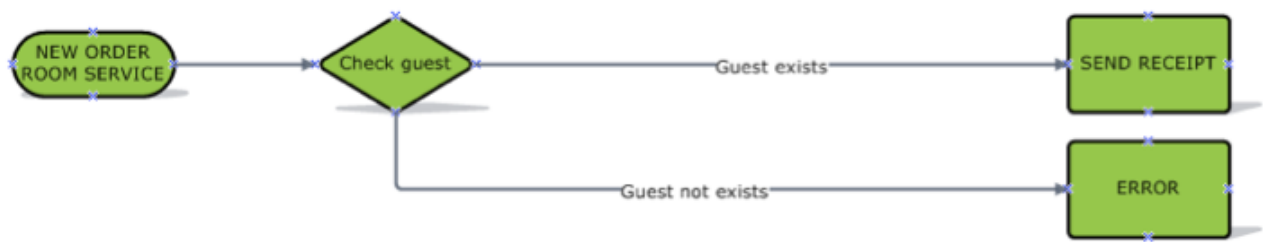
8. End of a journey



@param GuestName, GuestSurname

- Ask if the guest has a customer's account
- If true
 - Ask if the guest wants to use his customer's account
 - If true
 - Recover the data of the firm
 - Send the bill to the firm

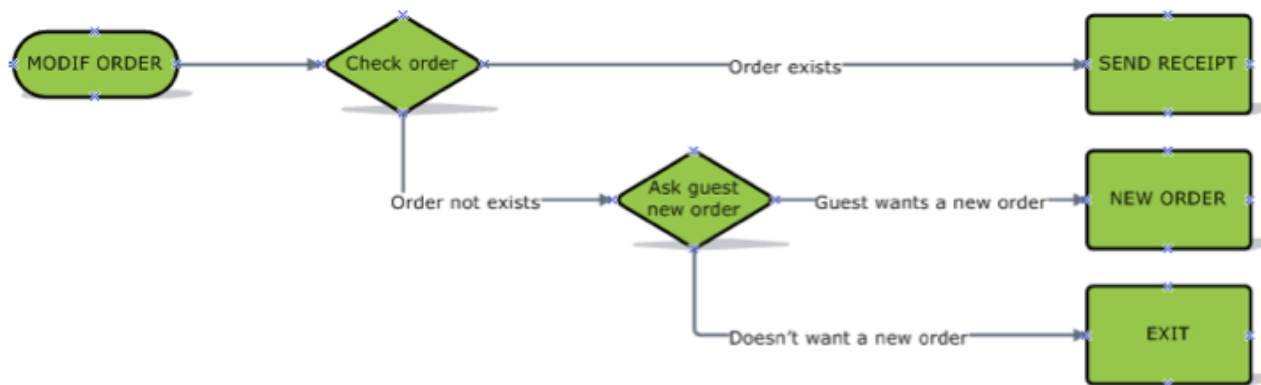
9. New room service order



@param RoomNumber

- Recover the **RoomNumber** on the telephone
- Ask the content of the order
- Create a new order
- Fill the order with the content given

10. Modification of an order



@param GuestName, GuestSurname, OrderId

- Ask **GuestName** and **GuestSurname**
- Treatments of information and display the corresponds orders
- If there is correspondences into the list box of bookings
 - Select the order line into the listbox of order
 - Click on the button modification of order
 - Recuperation of **OrderId** selected
 - Retrieve data of order for the guest
 - Open New windows and display information
 - Input disable button 'OK'
 - Input new data into champs
 - Verification of disposability of is request
 - If there is availability
 - Enable the button 'OK'
 - Click on the Ok button
 - Update data of the database
 - Else
 - Display pop up for not availability of request
- Else
 - If guest wants a new order
 - Return on the last windows
 - Click on the new order button
 - See *Mev9 (New room service order)*
 - Else
 - Return on the last page

11.Cancellation of an order



@param OrderId

- Delete off the system the order with its **OrderId**

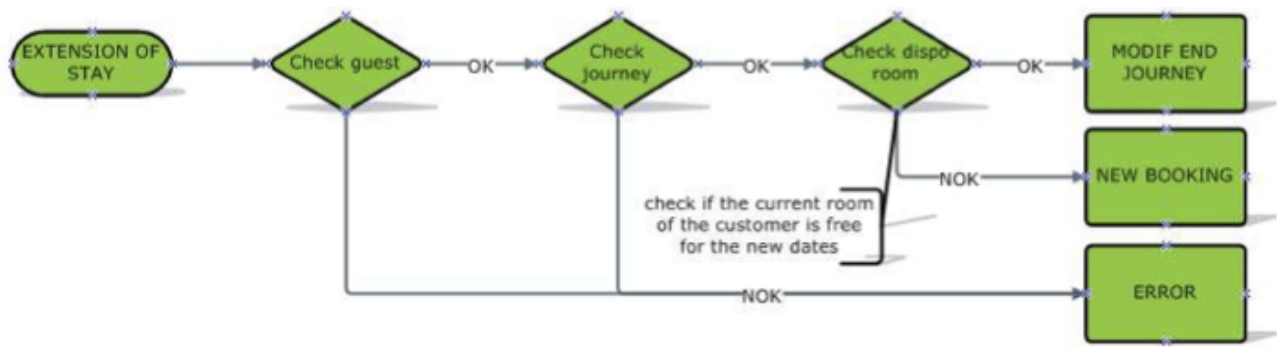
12. New complaint



@param GuestName, GuestSurname, ServiceLabel, ServiceId

- Ask **GuestName** and **GuestSurname**
- Enter the **GuestName** and the **GuestSurname** in the database
- Display the list of guest(s)
- If the guest not exists :
 - Error
- Else
 - Click on « New complaint »
 - Ask the **ServiceLabel** concerned by the complaint
 - Recover the **ServiceId** from the **ServiceLabel**
 - Enter the **ServiceId** into the corresponding field
 - Input the **ServiceId** into the database
 - Fill the content of the complaint
 - Input the content of the complaint into the database
 - Send the receipt to the manager of the service

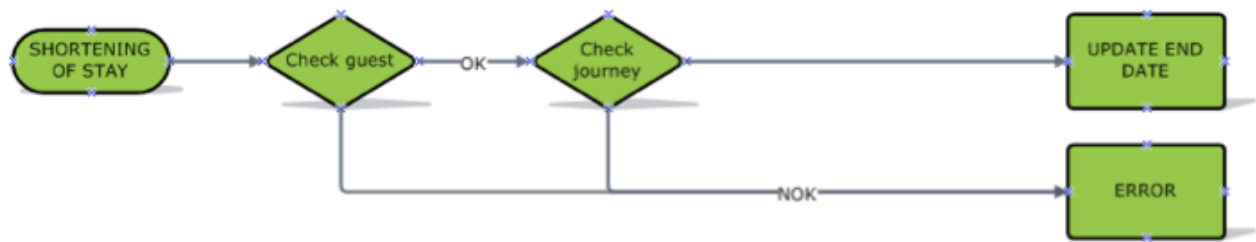
13.Extension of a stay



@param GuestName, GuestSurname, BookingDateOfBeginning

- See *Mev2* (Modification of a booking for a room)

14. Shortening of a stay



@param GuestName, GuestSurname, BookingDateOfBeginning

- See *Mev2 (Modification of a booking for a room)*

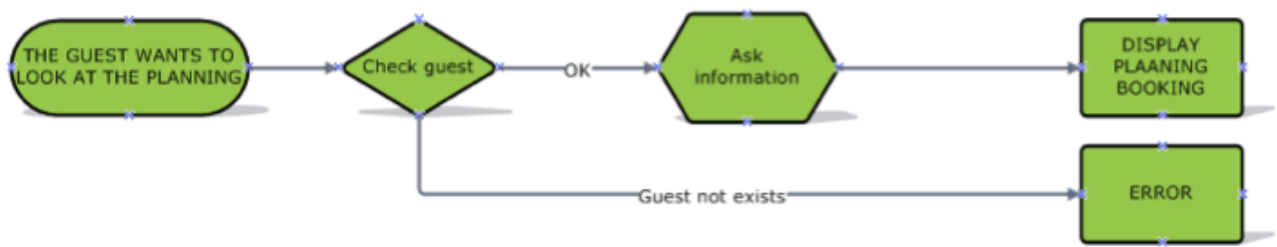
15.A guest uses a product from the mini-bar



@param RoomNumber

- Recover the **RoomNumber** associated to the mini-bar
- Recover the actual **GuestId** associated to the **RoomNumber**
- Recover the actual **BillId** associated to the **GuestId**
- Insert to data base a new bill line associated for the product used

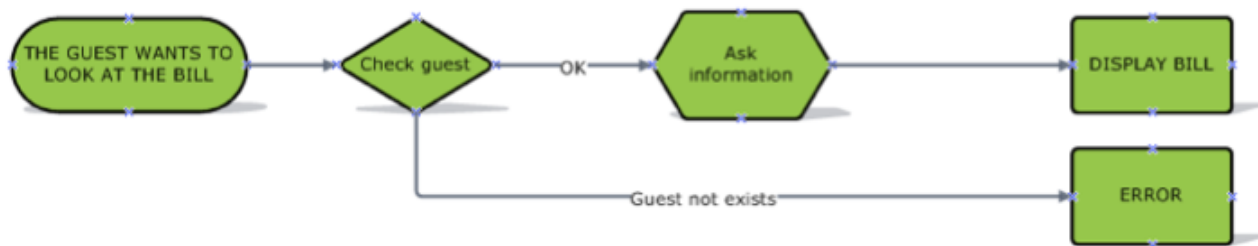
16.A guest wants to look at the planning



@param RoomNumber

- Input the **RoomNumber** into the field
- Treatment of information and display booking Planning

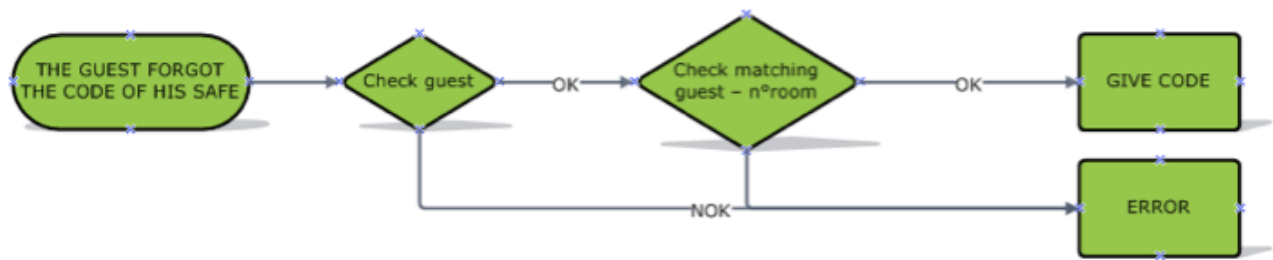
17. A guest wants to look at his bill



@param none

- Input **GuestName** and **GuestSurname** into the field
- On change of Guest field, *Call M7 (check guest existence)*
- If Guest not exist
 - Error
- Else
 - Display his bill

18.A guest forgot the code of his safe



@param GuestName, GuestSurname

- Input **GuestName** and **GuestSurname** into the field
- On change of Guest filed, *Call M7 (check guest existence)*
- If Guest not exist
 - Error
- Else
 - Ask to the guest his **RoomNumber**
 - Match **RoomNumber** with **GuestId** at the actual date
 - If there is no correspondences
 - Error
 - Else
 - Display the vault code and give him

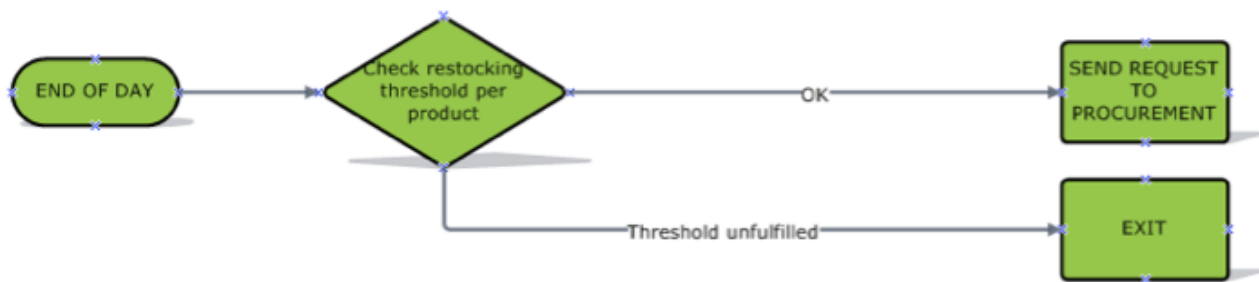
19. Add a new compensation



@param GuestName, GuestSurname

- Input **GuestName** and **GuestSurname** into the field
- On change of Guest champ, *Call M7 (check guest existence)*
- If Guest not exist
 - Error
- Else
 - Display the list of his complaint(s)
 - If the list box is empty
 - Error
 - Else
 - Select the corresponding complaint
 - Click on the button « add compensation » attached to the selected complaint
 - Input the amount of the compensation into the field
 - Recover the **GuestId** from the **GuestName** and the **GuestSurname**
 - Insert a new bill line in the bill of the corresponding guest

20. End of day



@param none

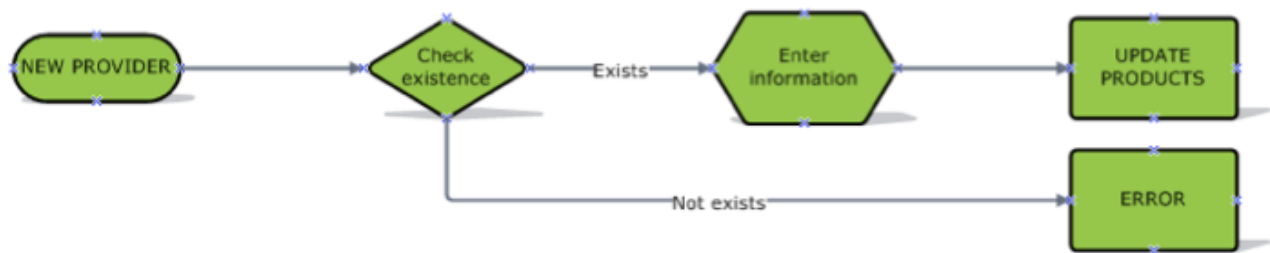
- For each stock
 - For each product make the difference between amount threshold and actual quantity
 - If ≤ 0
 - Send request request to procurement
 - Else
 - Next product row

21. Products delivered



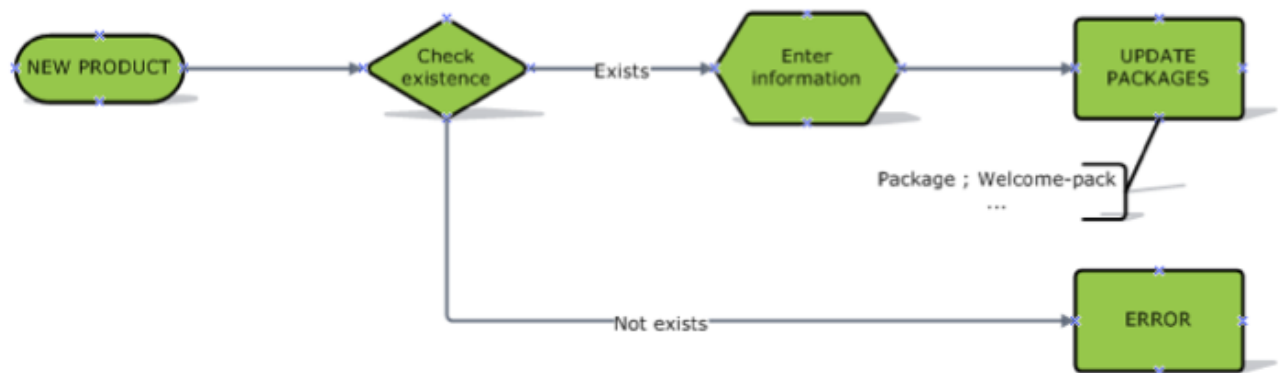
- For each product
 - Check if product is good (Quantity and quality)
 - If OK update stock
 - Click on stock
 - Click on modify product for update stock or new product
 - If NOKs
 - If quantity is NOK
 - Send report to procurement
 - Update stock
 - If quality is NOK
 - Send to procurement

22. New provider



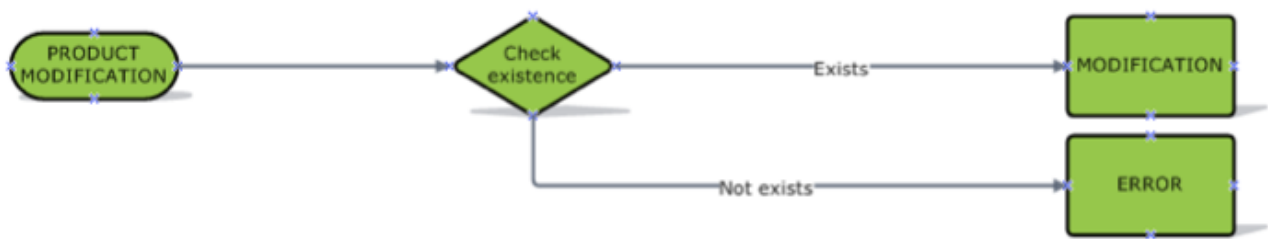
- Check existence provider by researches in pages jaunes
- Check existence provider in the database
- If not exists
 - Error
- If exists
 - Enter information
 - Move to « stock »
 - « modify product » or « new product » to update the stock according to the new provider

23. New product



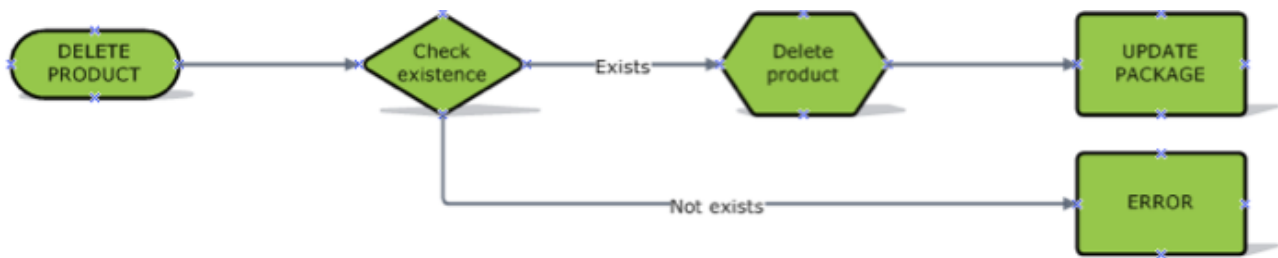
- Check existence product in the database
- If already exists
 - Error
- Else
 - Click on «stock»
 - Click on «new product»
 - Enter new information into the fields
 - Enter the new product into the DataBase with informations

24. Modification of a product



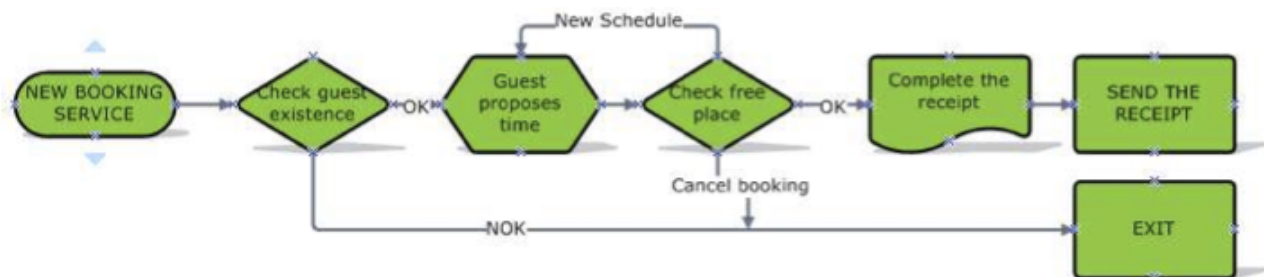
- Check existence product in the database
- If not exists
 - Error
- Else
 - Click on «stock»
 - Click on «modify product»
 - Enter new information into the fields
 - Update the corresponding information in the database

25. Delete a product



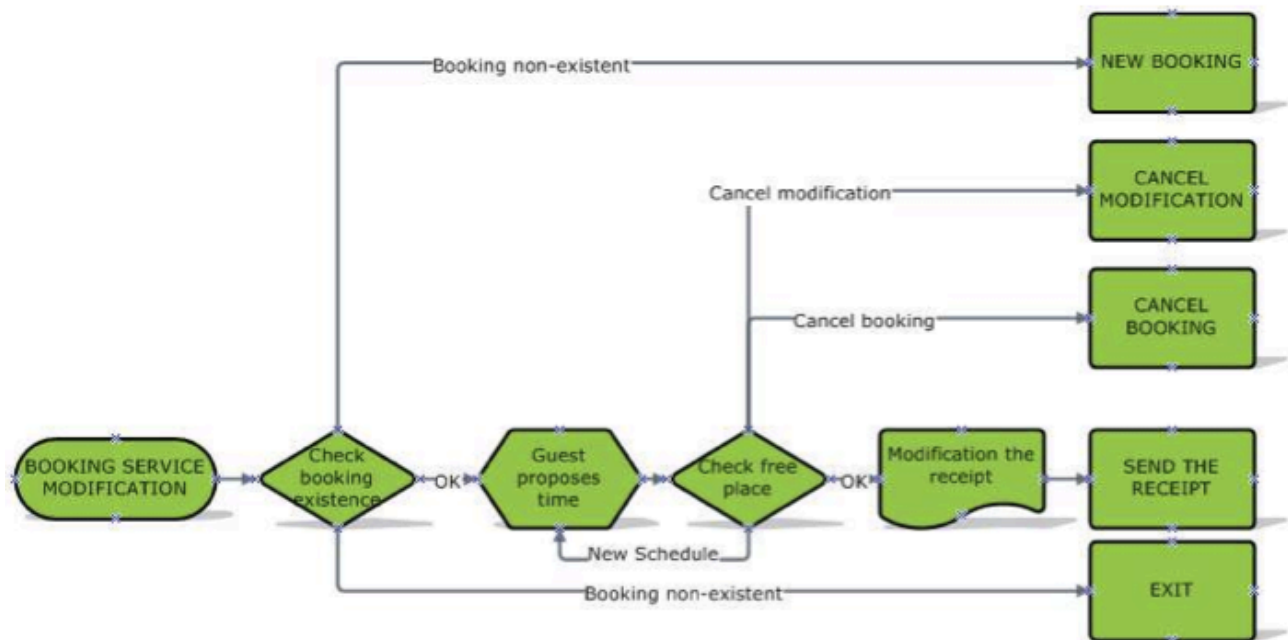
- Check existence product in the database
- if not exists
 - Error
- Else
 - Delete the corresponding product with its productId from the database

26. New booking for a service



- Input **GuestName** and **GuestSurname** into the field
- On change of Guest field, *Call M7 (check guest existence)*
- If Guest not exist
 - Error
- Else
 - Ask to the guest the service concerned
 - Click on « reservation »
 - Click on « new booking »
 - Display the planning of all the bookings for the service selected
 - See with the guest when he can take his reservation

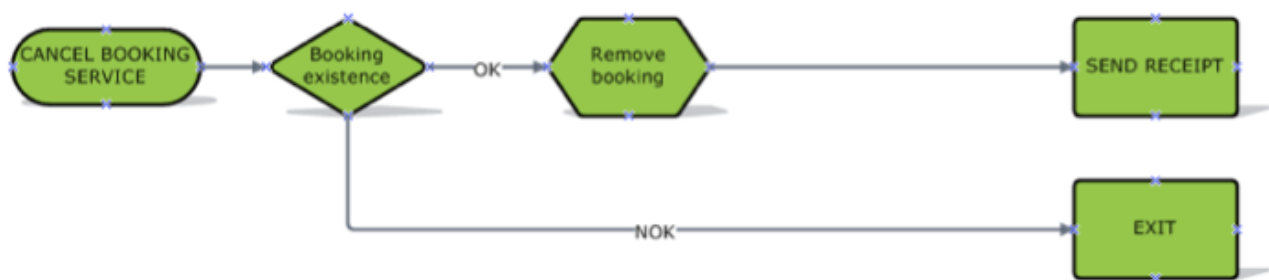
27. Modification of a booking for a service



@param GuestName, GuestSurname, BookingDateOfBeginning

- Ask **ServiceLabel** and go on the tab corresponding
- Ask **GuestName**, **GuestSurname** and **BookingDateOfBeginning**
- Treatments of information and display the corresponds bookings
- If there is correspondences into the list box of bookings
 - Select the booking line into the list box of booking
 - Click on the modification booking button
 - Recuperation of **BookingId** selected
 - Display new screen
 - Retrieve data of booking for the guest
 - Input data of the database correspond of **BookingId** and **GuestId** into champs
 - Champs surname and name are disabled
 - Input disable button 'OK'
 - Input data into champs
 - Verification of free place for the date time asked
 - Treatments of information and display the corresponds bookings
 - If there is correspondences into the list box of bookings
 - Select the booking line into the list box of booking
 - Enable the button 'OK'
 - Click on the Ok button
 - Update data of the database
 - Free space for the date of the old booking into the service requested
 - Else
 - Display pop up for not free space of request
 - If Click on the Cancel Booking
 - Delete the booking data into the database
 - Free space for the date of the old booking into the service requested
- Else
 - See *Mev26 (New booking for a service)*

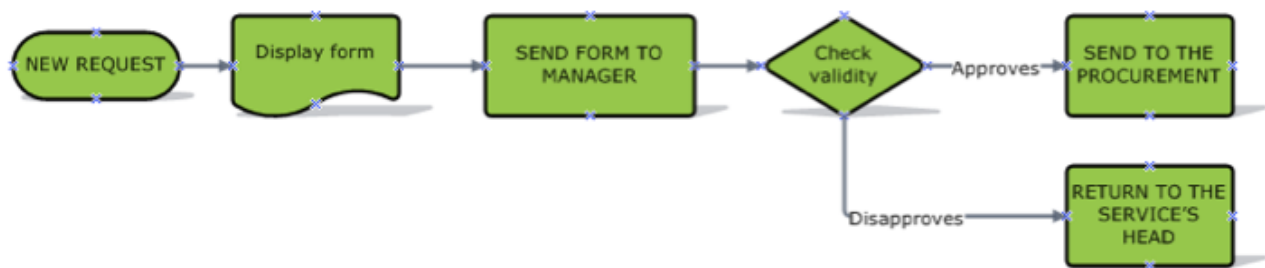
28. Cancel a booking for a service



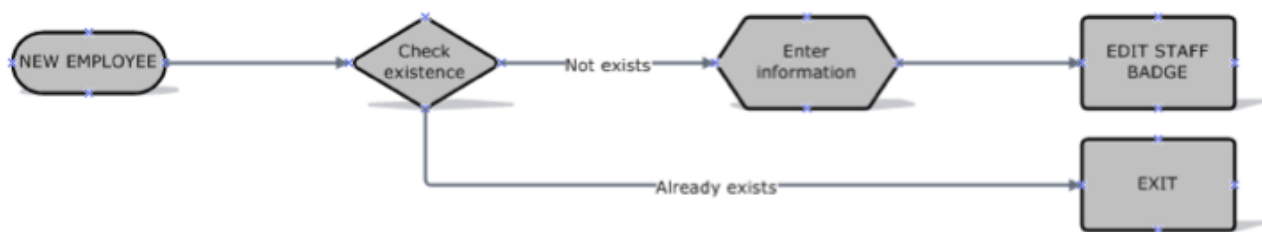
@param ProductLabel, ProductId

- Click on « Product »
- Enter the **ProductLabel** into the field
- If product exists
 - Recover the **ProductId** corresponding
 - Click on « Modify product »
 - Enter the **ProductId** into the field
 - Enter the new information about the product
- Else
 - Error

29. New request



30. New employee



@param StaffName, StaffSurname

- Ask **StaffName** and **StaffSurname**
- Enter **StaffName** and **StaffSurname** into the database
- Display the list of employees
- If the employee already exists
 - Error
- Else
 - Ask the information of the new employee
 - Click on « New employee »
 - Enter the information
 - Input the information given into the database

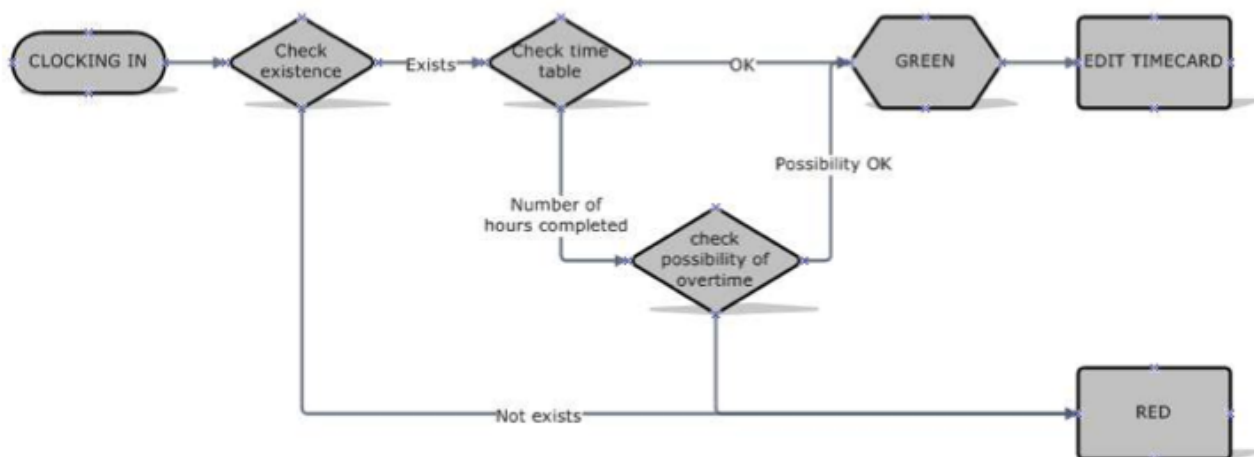
31. Modification of an employee



@param StaffName, StaffSurname

- Ask the **StaffName** and **StaffSurname** to the employee
- Enter the **StaffName** and **StaffSurname** into the database
- Display the list of employee(s)
- If the employee doesn't exist :
 - Error
- Else
 - Ask the new information
 - Enter the new information into the fields
 - Replace the old information by the new

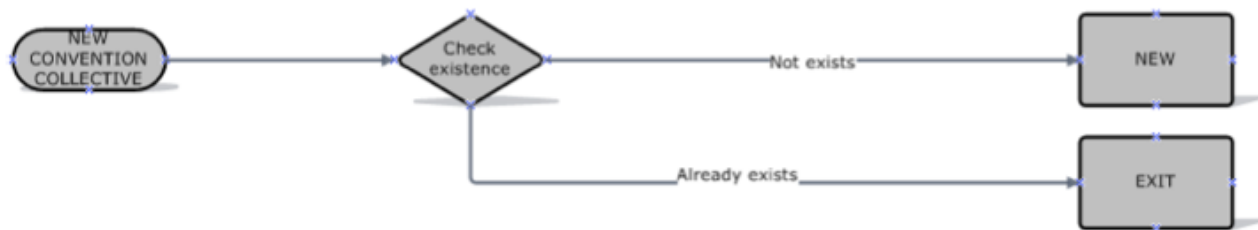
32.Clocking in



@param none

- Recuperation of StaffId
- Call M10 (check Staff Existence)
- If staff exist
 - Call M11 (Check Time Table)
 - If time = zero
 - Display green light
 - Edit TimeCard
 - Else
 - Call M12 (Check Possibility of overtime)
 - If possibility = true
 - Display green light
 - Edit TimeCard
 - Else
 - Display red light
- Else
 - Display red light

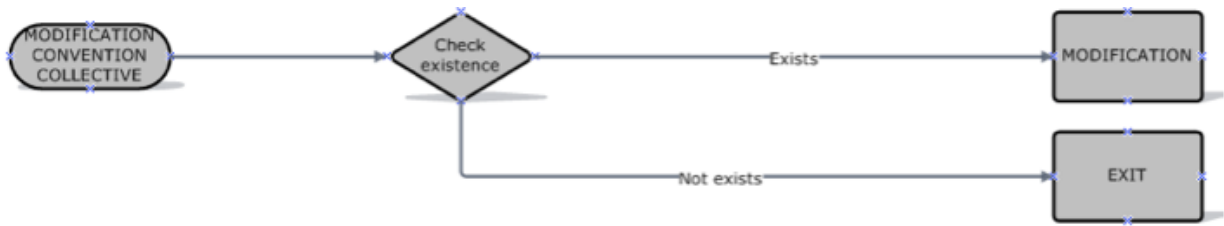
33.New convention collective



@param CCId, CCLabel, CCNbOfHoursMaxPerDay, CCNbOfHoursMaxPerWeek, CCNbOfHoursMaxPerMonth.

- Enter information about the new convention collective into the database
- Check if this convention collective already exists in the database
- If exists
 - Error
- Else
 - Click on "Staff"
 - Click on "New convention collective"
 - Enter the information
 - Input a new line with the information given

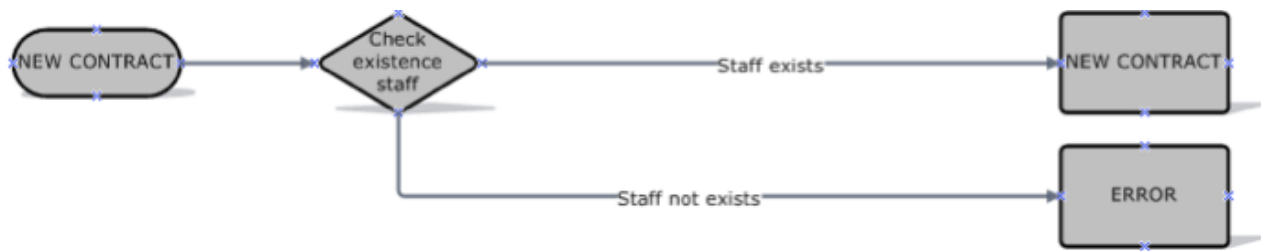
34. Modification of a convention collective



@param CCId, new CCId, new CCLabel, new CCNbOfHoursMaxPerDay, new CCNbOfHoursMaxPerWeek, new CCNbOfHoursMaxPerMonth.

- Enter the **CCId** into the database
- Check if this convention collective already exists in the database
- If not exists
 - Error
- Else
 - Click on the “Staff” button
 - Click on the “Modify convention collective” button
 - Enter the new information
 - Replace old **CCId** with new **CCId**,
 - Replace old **CCLabel** with new **CCLabel**,
 - Replace old **CCNbOfHoursMaxPerDay** with new **CCNbOfHoursMaxPerDay**,
 - Replace old **CCNbOfHoursMaxPerWeek** with new **CCNbOfHoursMaxPerWeek**,
 - Replace old **CCNbOfHoursMaxPerMonth** with new **CCNbOfHoursMaxPerMont**

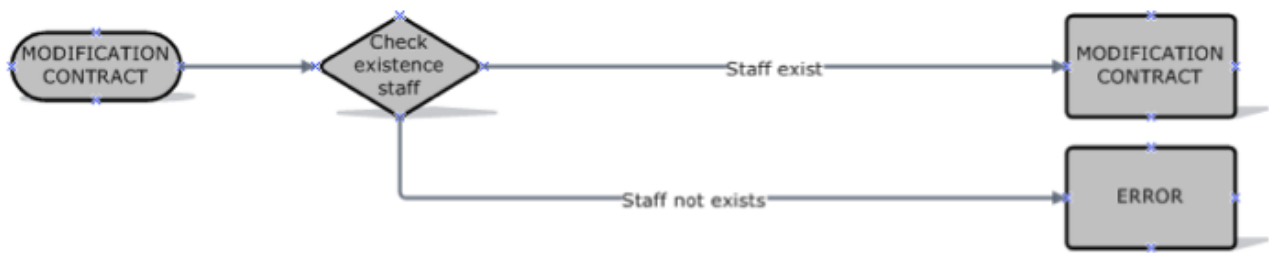
35. New contract



@param StaffName, StaffSurname, ContractId, ContractDateBeginning, ContractDateEnd, ContractNbOfHoursMinPerDay, ContractNbOfHoursMinPerWeek, ContractNbOfHoursMinPerMonth, ContractTypeld

- Enter the **StaffName** and **StaffSurname** into the database
- Select the right one
- Recover his **StaffId**
- Call M10 (*Check staff existence*)
- If true
 - Click on "Staff"
 - Click on "New contract"
 - Enter the information about the contract
 - Input the information into the database :
 - **ContractId** ;
 - **ContractDateBeginning** ;
 - **ContractDateEnd** ;
 - **ContractNbOfHoursMinPerDay** ;
 - **ContractNbOfHoursMinPerWeek** ;
 - **ContractNbOfHoursMinPerMonth** ;
 - **ContractTypeld**.

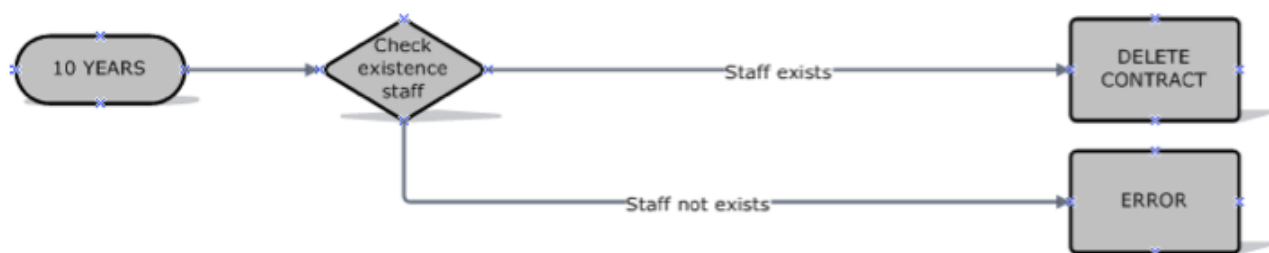
36. Modification of a contract



@param StaffName, StaffSurname, ContractId, new ContractDateBeginning, new ContractDateEnd, new ContractNbOfHoursMinPerDay, new ContractNbOfHoursMinPerWeek, new ContractNbOfHoursMinPerMonth, new ContractTypeId

- Enter the **StaffName** and **StaffSurname** into the database
- Select the right one
- Recover his **StaffId**
- Call *M10 (Check staff existence)*
- If true
 - Enter the **ContractId** in the system
 - Check if the contract exists
 - If exists
 - Click on "Staff"
 - Click on "Modify contract"
 - Enter the new information in the database
 - Update data of contract into database :
 - **ContractDateBeginning** ;
 - **ContractDateEnd** ;
 - **ContractNbOfHoursMinPerDay** ;
 - **ContractNbOfHoursMinPerWeek** ;
 - **ContractNbOfHoursMinPerMonth** ;
 - **ContractTypeId**.

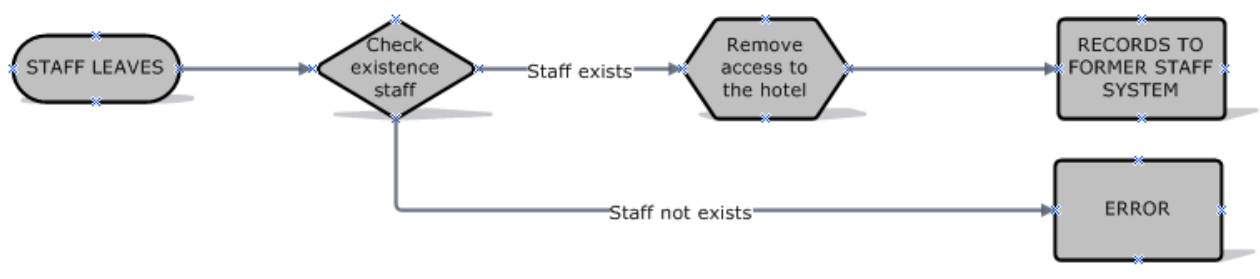
37. Ten years



@param StaffName, StaffSurname, StaffActive

- Enter the **StaffName** and **StaffSurname** into the database
- Select the right one
- Recover his **StaffId**
- Recover the **ContractId** of the **StaffId**
- Call *M10* (Check staff existence)
- If true
 - Check if his flag **StaffActive**
 - if false
 - Delete the contract with the **ContractId**
 - Else
 - Exit
- Else
 - Error

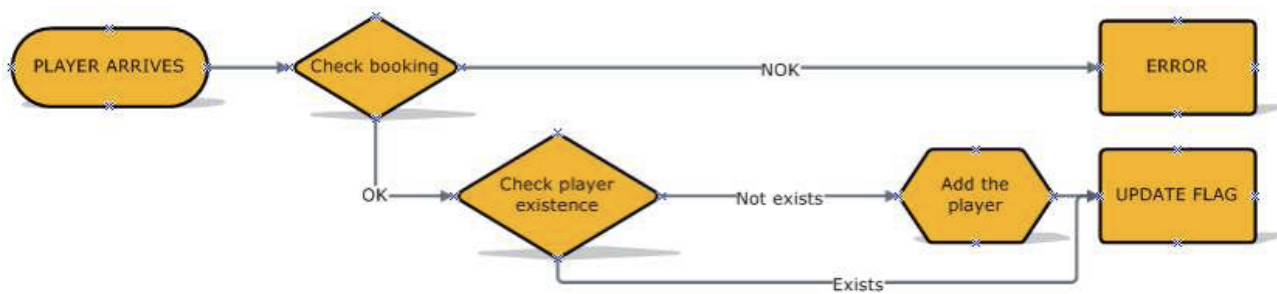
38.A staff member leaves



@param StaffFirstName, StaffSurname

- Ask **StaffFirstName** and **StaffSurname**
- Treatments of information and display the corresponds employee
- Select the employee line into the list box
- Click on the "Delete employee" button
- Recuperation of **StaffId** selected
- Update the status of **StaffActive**
- Ask time card's employee

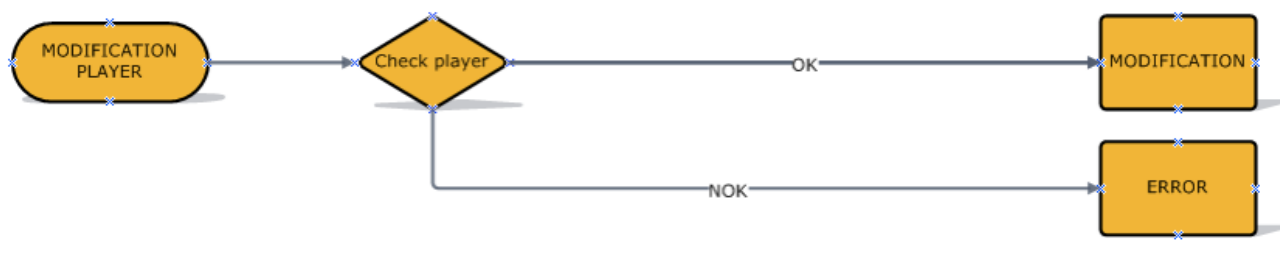
39.A player arrives



@param GuestName, GuestSurname

- Ask **GuestName** and **GuestSurname**
- Treatments of information and display the corresponds bookings
- If there is correspondences into the list box of bookings
 - Select the booking line into the listbox of booking
 - Click on the button arrivals
 - Recuperation of **BookingId** selected
 - Display informations about this booking and list of players
 - Ask all **PlayerName** and **PlayerSurname**
 - Check correspondences with the list of players
- If a player not exists
 - Click on "New Player" button
 - Display new window with :
 - A fields "Name"
 - Input data into fields
 - Click on button "Create Player"
 - If the registration went well
 - Display message box "The player already exists"
 - Else
 - Display message box "A error occurred when saving"
- Retrieve the **GuestId** for this booking
- Input data into database :
 - Delete or classify this **BookingId**.

40. Modification of an information about a player



@param PlayerName, PlayerSurname

- Ask **PlayerName**, **PlayerSurname** and new score
- Treatments of information and display the corresponds players
- If there is correspondences into the list box of player
 - Select the player line into the listbox of player
 - Click on the button modification
 - Recuperation of **PlayerId** selected
 - Display information about this player
 - Input data into fields
 - Click on the "Modify" button
 - Update data into database

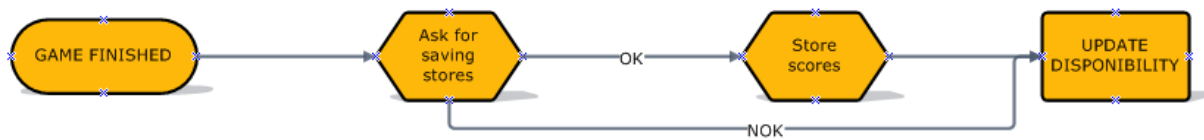
41.A player wants to see his score history



@param PlayerName, PlayerSurname

- Ask **PlayerName**, **PlayerSurname** and new score
- Treatments of information and display the corresponds players
- If there is correspondences into the list box of player
 - Select the player line into the listbox of player
 - Click on the button "Consult score"
 - Recuperation of **PlayerId** selected
 - Display information about this player :
 - Date and score

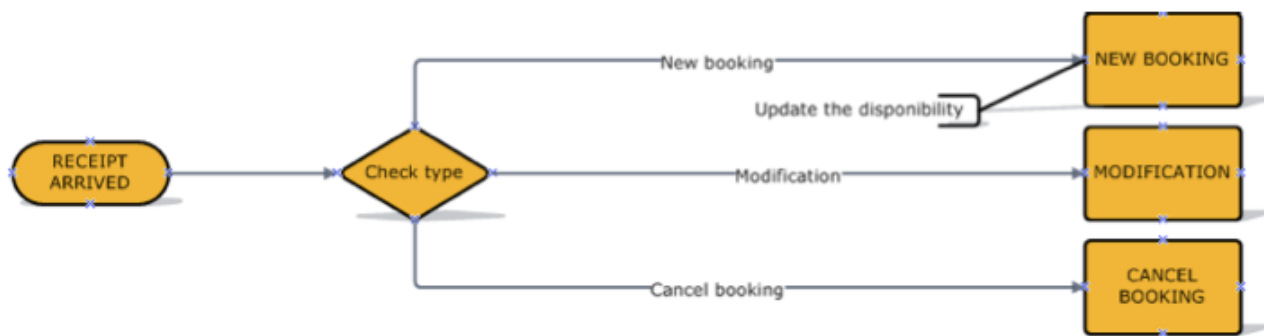
42. The game is finished



@param `PlayerName`, `PlayerSurname`

- Ask **PlayerName**, **PlayerSurname** and new score
- Treatments of information and display the corresponds players
- If there is correspondences into the list box of player
 - Select the player line into the list box of player
 - Click on the button "Add score"
 - Recuperation of **PlayerId** selected
- Display new window with
 - A field "Date"
 - A field "Score"
- Input data into fields
- Click on button "Add score"
- Input data into database

43. A receipt arrives



44.A provider delivers an equipment



- See Mev23 (New product)

45.A contractor comes to maintain the green

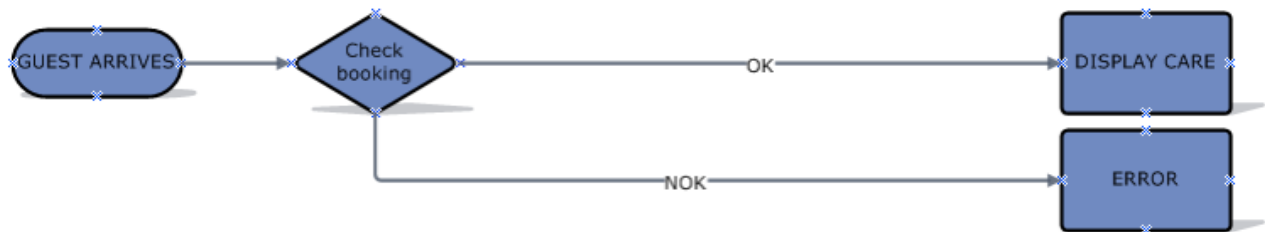


46. An equipment is broken



- See Mev25 (Delete product)

47. A guest arrives



@param GuestName, GuestSurname

- Click on “Guest booking”
- Display list of booking for spa
- Ask **GuestName** and **GuestSurname**
- If there is correspondences into the list box of bookings
 - Show the list of care and care menu to guest

48. A guest leaves



@param GuestName, GuestSurname

- Click on “Guest booking” tag
- Display list of booking for this day concerning the spa
- Ask **GuestName** and **GuestSurname**
- If there is correspondences into the list box of guest
 - Select the guest line into the list box of guest
 - Click on “Care”, “Care Menu” or “Product” button
 - If user click on “Care” button
 - Display list of cares and display a button “Add”
 - Select a care and click on “Add”
 - Recuperation of **CareId** selected
 - Display this care on a frame
 - If user click on “Care Menu” button
 - Display list of care menu and display a button “Add”
 - Select a care and click on “Add”
 - Recuperation of **CareMenuId** selected
 - Display this care menu on a frame
 - If user click on “Product” button
 - Display list of products and display a button “Add”
 - Select a care and click on “Add”
 - Recuperation of **ProductId** selected
 - Display this product on a frame
- Click on the “Add to bill” button
- Insert a bill line into database

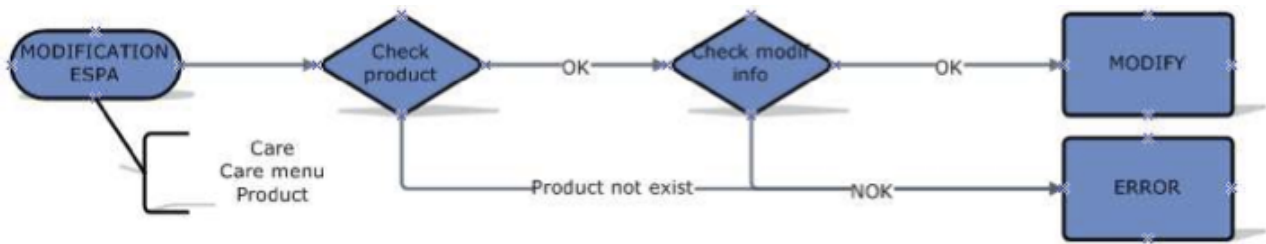
49.New ESpa care, care menu or product



@param Information about a care, a care menu or a product

- Click on “Care”, “Care Menu” or “Product” tag
- Click on “New” or “Modification”
- If user click on “New”
 - Display a new form
 - Input value in fields
 - Click on “Add” button
 - Input data in database

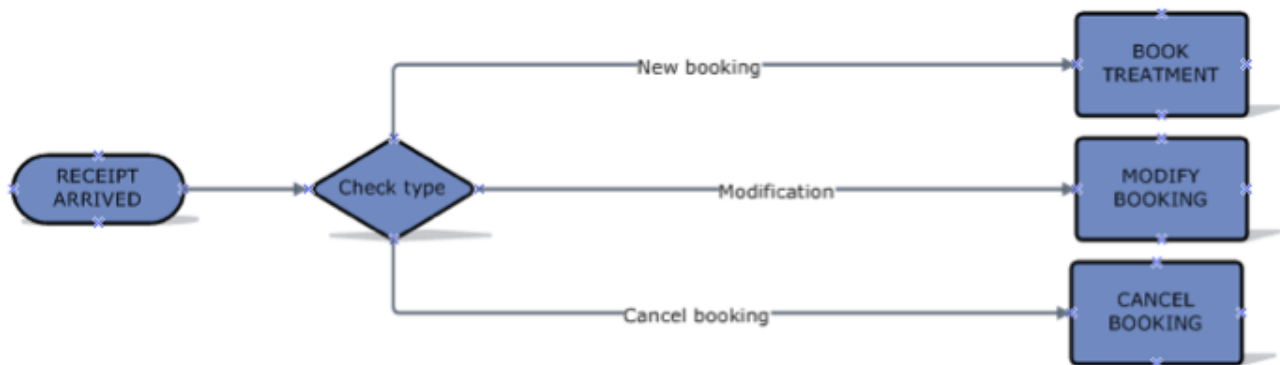
50.Modification of an ESpa care, care menu or product



@param Information about a care, a care menu or a product

- Click on "Care", "Care Menu" or "Product" tag
- Click on "New" or "Modification"
- If user click on "Modification"
 - Display list of care, care menu or product
 - Select a line
 - Retrieve id of line selected
 - Display a new form with fields already filled
 - Input value in fields
 - Click on "Modify" button
 - Update data in database

51. A receipt arrives



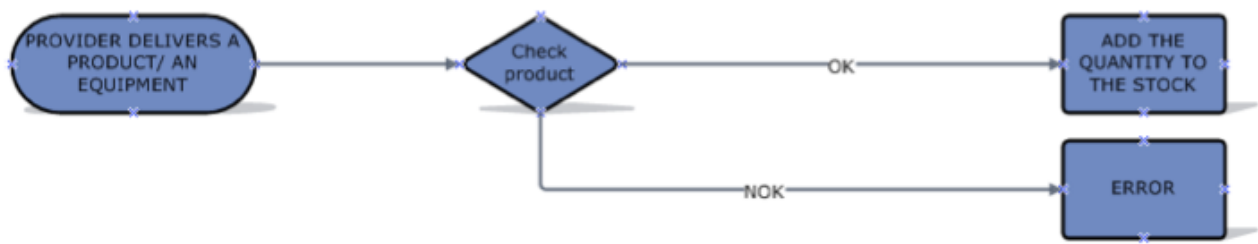
52.A contractor analyzes the pool



@param PoolId, PoolPhLevel, PoolChlorineLevel, PoolTAFactor, PoolTHFactor, PoolSaltLevel, PoolBacterial/ViralSampleResults

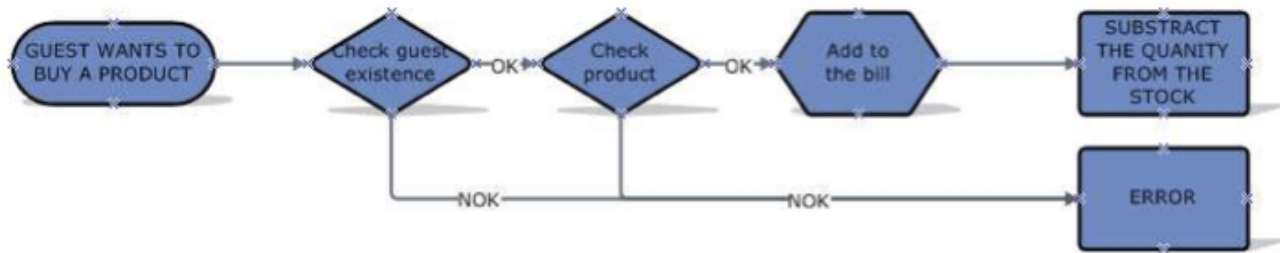
- Click on “New Report”
- Display new window with :
 - A selection field “Pool ID”
 - A field “Ph Level”
 - A field “Chlorine Level”
 - A field “TA Factor”
 - A field “TH Factor”
 - A field “Salt Level”
 - A field “Bacterial/Viral Sample Results”
- Input data into fields
- Click on “Save”
- If user click on “Buy”
- Input data in database with the current date

53.A provider delivers a product or an equipment



- See Mev23 (New product)

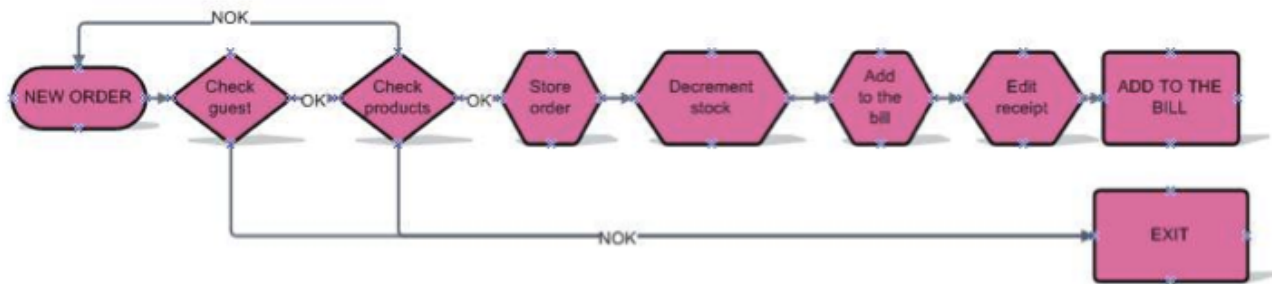
54. A guest wants to buy a product



@param GuestName, GuestSurname

- Click on "Product" tag
- Click on "Buy"
- If user click on "Buy"
 - Ask **GuestName, GuestSurname**
 - Display list of the corresponding guest
 - Select the corresponding line
 - Display list of product
 - Select a product and click on "Add" button
 - Display this product on a new frame
 - Click on "Add to bill" button
 - Input bill line in database

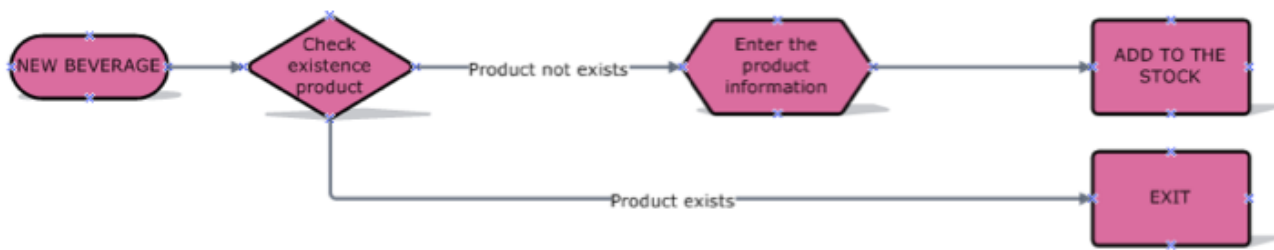
55. New order



@param GuestName, GuestSurname

- Ask GuestName, GuestSurname for the bill
- The waiter input the GuestName and GuestSurname into the program
- Call M7 (check guest existence)
- If guest exist
 - Ask order and select it on the programm
 - If no product fails to prepare the order
 - Prepare order
 - Edit receipt for the guest
 - Add to the bill and update the stock of products
 - Give the receipt to the guest
- Else
 - Call Mev55 (New order)

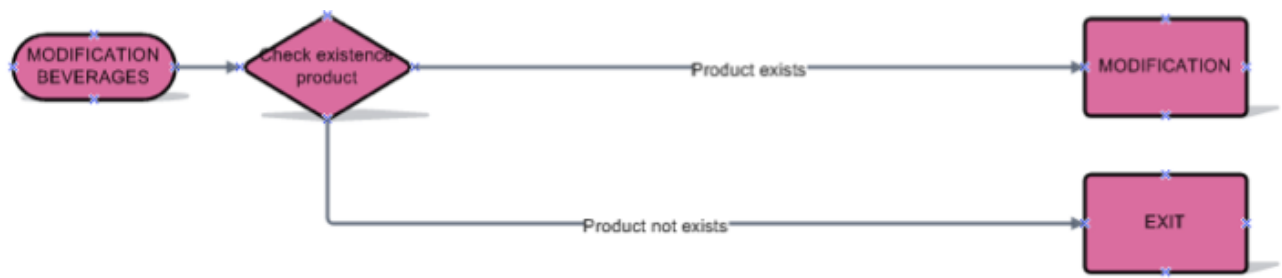
56. New beverage



@param none

- See *Mev23 (New Product)*

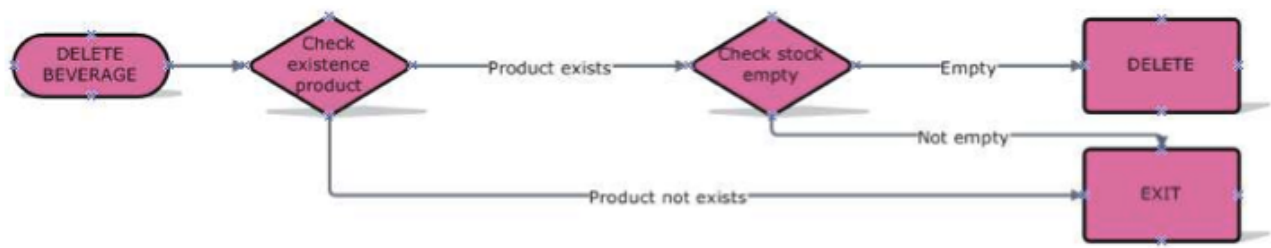
57.Modification of a beverage



@param ProductId

- See *Mev24 (Modification of a product)*

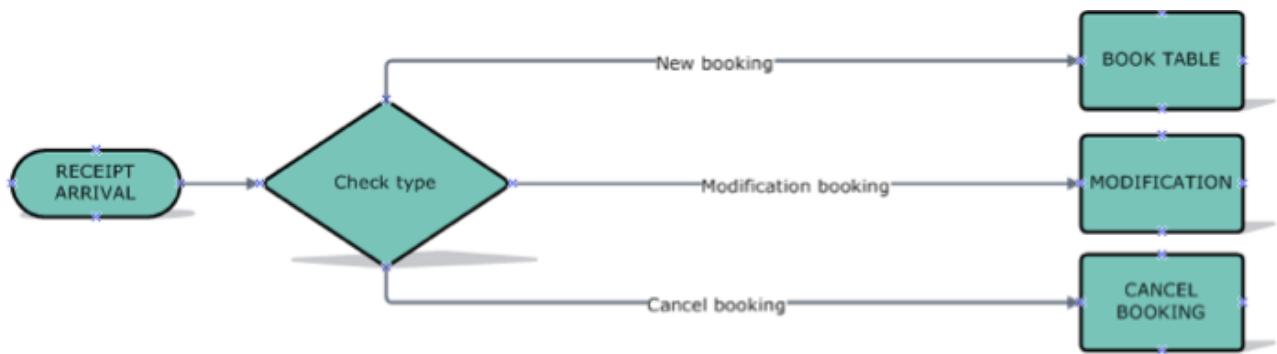
58. Delete a beverage



@param ProductId

- See Mev25 (Delete a product)

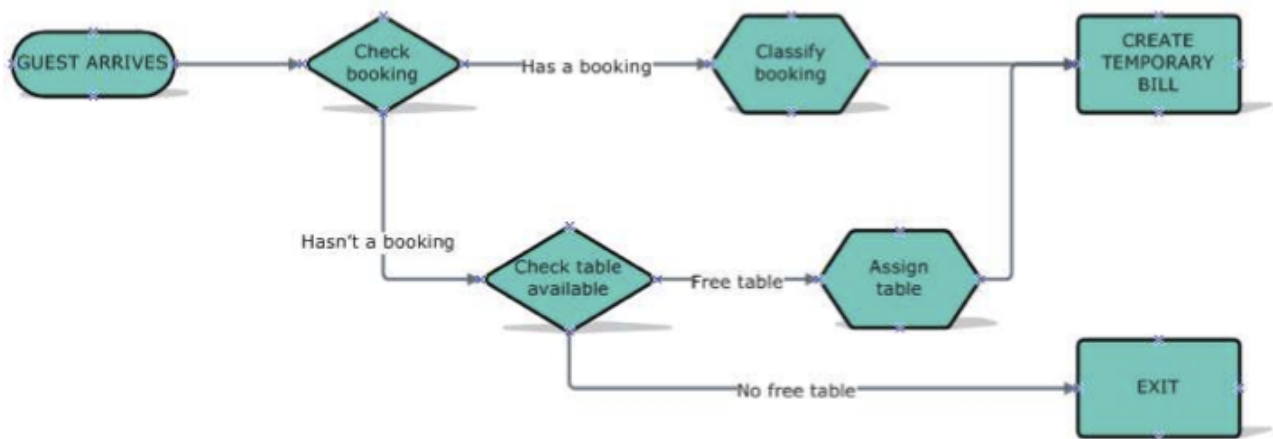
59. Receipt arrival



@param none

- Check the nature of the receipt
- If the receipt is « New booking »
 - *Call Mev26 (New booking for a service)*
- If the receipt is « Modification of a booking »
 - *Call Mev27 (Modification of a booking for a service)*
- If the receipt is « Cancel booking »
 - *Call Mev28 (Cancel a booking for a service)*

60.A guest arrives



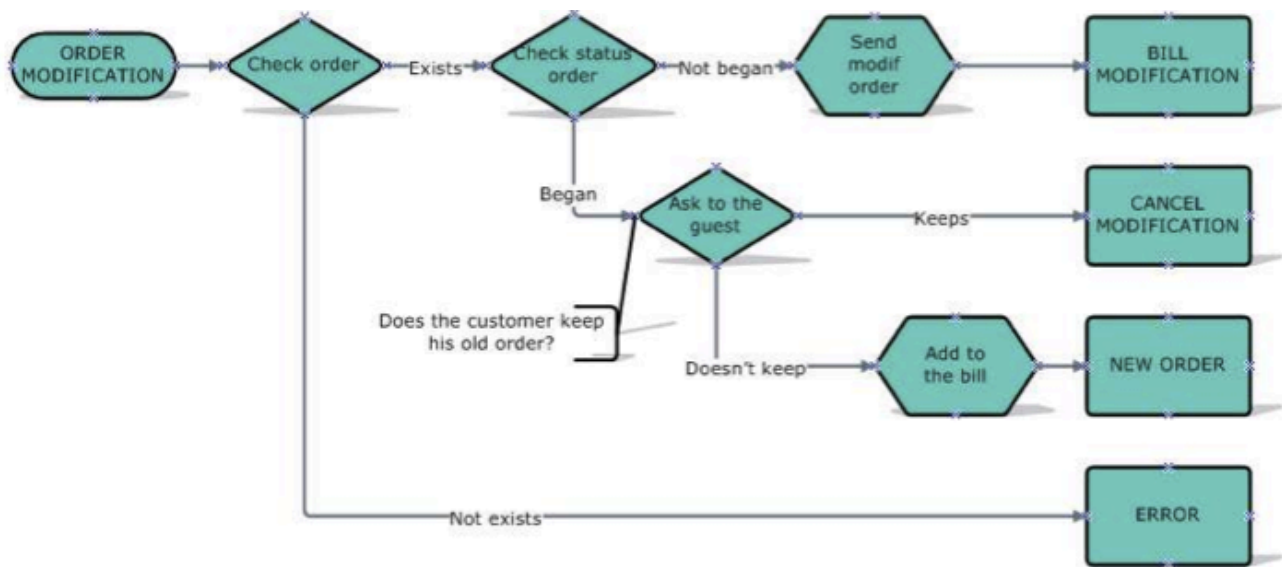
@param GuestName, GuestSurname

- Ask GuestName and GuestSurname
- Treatments of information and display the corresponds bookings
- If there is correspondences into the list box of bookings
 - Select the booking line into the listbox of booking
 - Click on the button arrivals
 - Recuperation of BookingId selected
 - Retrieve the GuestId for this booking
 - Input data into database :
 - Delete or classify this BookingId.
 - Create temporary bill
- Else
 - Display free table
- If there are free tables for number of people
 - Update flag TableAvailability on database
 - Create temporary bill

61.New order



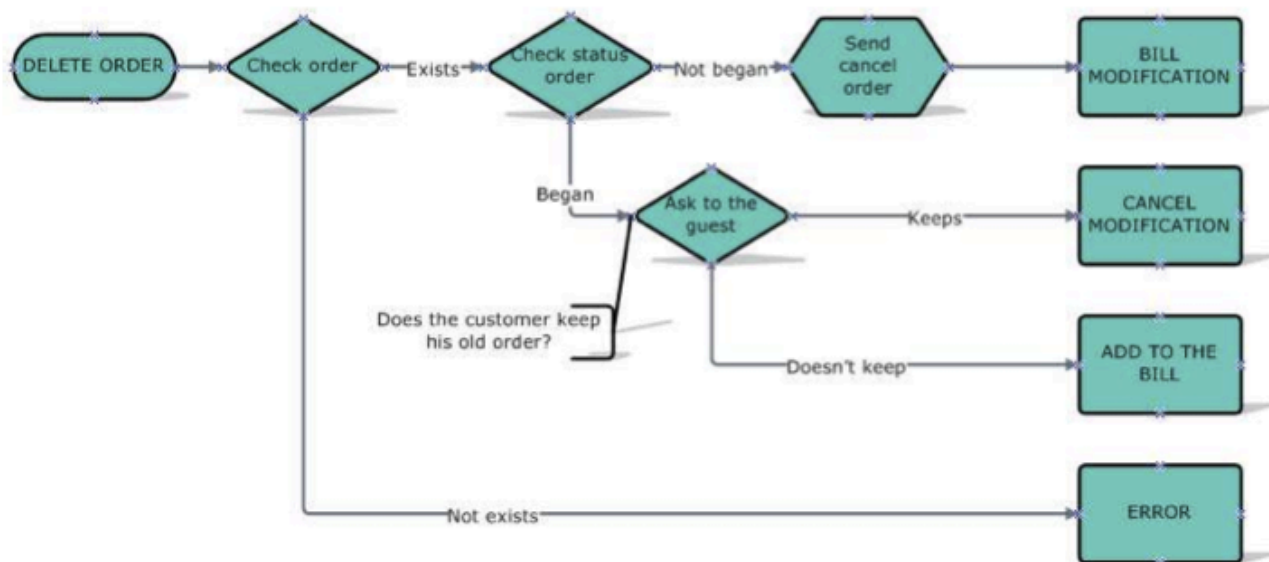
62. Modification of an order



@param GuestName, GuestSurname

- Ask GuestName and GuestSurname
- Treatments of information and display the corresponds orders
- If there is correspondences into the list box of bookings
 - Select the order line into the listbox of order
 - Check status order
 - If the order is not started yet
 - Click on the button modification of order
 - Recuperation of OrderId selected
 - Retrieve data of order for the guest
 - Open New windows and display information
 - Input new data into champs
 - Click on the Ok button
 - Update data of the database
 - Else
 - Ask to the guest is he wants to keep his order
 - If he wants to keep is order
 - Close window
 - Else
 - Call Mev61 (New order)
 - Else
 - If guest wants a new order
 - Call Mev61 (New order)

63. Delete an order



@param GuestName, GuestSurname

- Ask GuestName and GuestSurname
- Treatments of information and display the corresponds orders
- If there is correspondences into the list box of bookings
 - Select the order line into the listbox of order
 - Check status order
 - If the order is not started yet
 - Click on the button Cancel order
 - Recuperation of OrderId selected
 - Retrieve data of order for the guest
 - Open New windows and display information
 - Click on the Ok button
 - Update data of the database
- Else
 - If guest wants a new order
 - Call Mev61 (New order)

64.An order is prepared



@param OrderId

- Select the order line into the list box of orders
- Click on prepared order button
- Recuperation of **OrderId**
- Update the **OrderType** on « prepared » into the database
- Update the list box of orders for clean the delete of the screen the prepared order

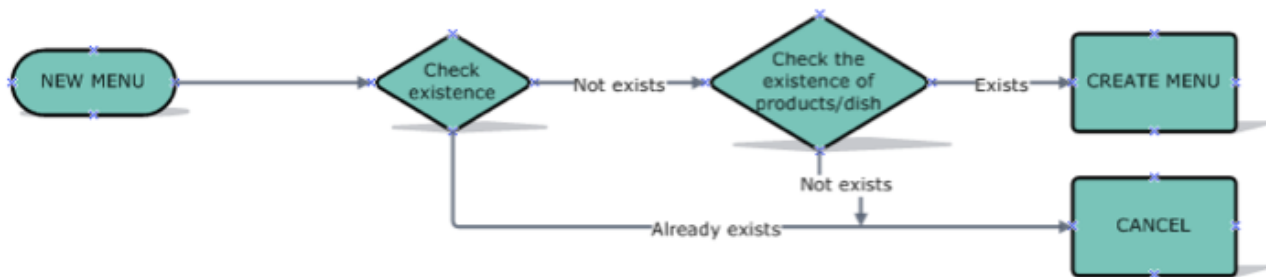
65. A guest leaves



@param TableId

- Select the **TableId** into the list box of tables
- Click on available button
- Retrieve of **TableId** selected
- Retrieve of **GuestId** of the table selected
- If guest have temporary bill
 - Transfer **TableId** to *Mev75*
- If there isn't a reservation of this table after
 - Put the **TableAvailability** flag to « available »

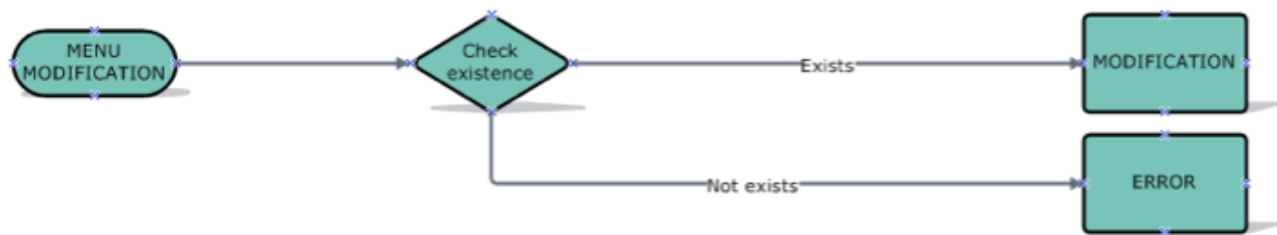
66. New menu



@param none

- Click on new menu button
- Open a new windows
- Disable the ok button while the **MenuLabel** is not enter
- Display all dish of hotel into the list box of dishes
- Input the **MenuLabel** into the field « Menu Label »
- On change of field « Menu Label » Call M12 (Check the existence of **MenuLabel**)
- If the menu exist
 - Display an error pop up (« This menu label is already used »)
 - Enter a new **MenuLabel** into the field « Menu Label », or cancel operation
- Select a **MenuType** into the list box of menu type
- While a dish was selected the MenuPrice are recalculated
- Select all dishes composed the new menu
- Update and display list boxes of dishes
- Click on the Ok button
- Insert data into the database :
 - **MenuLabel** ;
 - **MenuType** ;
 - **MenuPriceHT** ;
 - **MenuPriceTTC** ;
 - **MenuTVARate**.

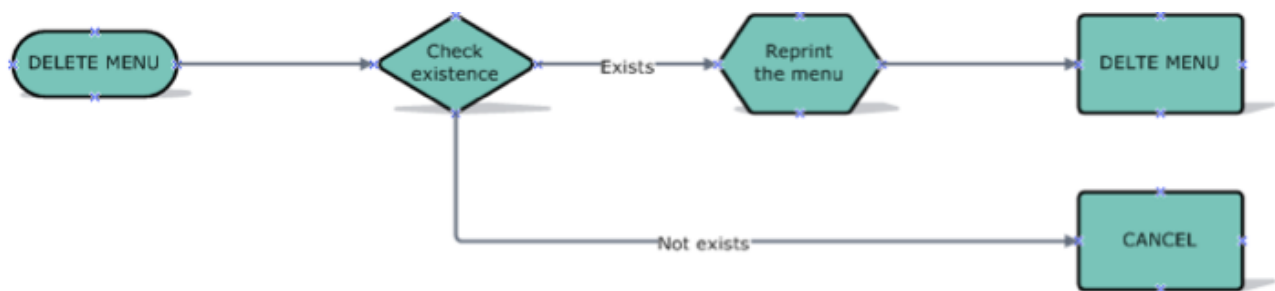
67. Modification of a menu



@param MenuId

- Select **MenuLabel** into the list box of menus
- Click on the modification menu button
- Recuperation of menu informations
- Open a new window
- Input old menu data into field of window
- While no change as done, disable the ok button
- Input new datas into fields
- If is the « Menu Label » field
 - Call M13 (Check the existence of **MenuLabel** excepted him (param this))
 - If the menu exist
 - Display an error pop up (« This menu label is already used »)
 - Enter a new **MenuLabel** into the field « Menu Label », don't change the **MenuLabel**, or cancel operation
 - Else
 - Enable the Ok button
- Add or delete dishes with the add button and delete button
- Update and display list boxes of dishes
- Click on the ok button
- Update data of the database

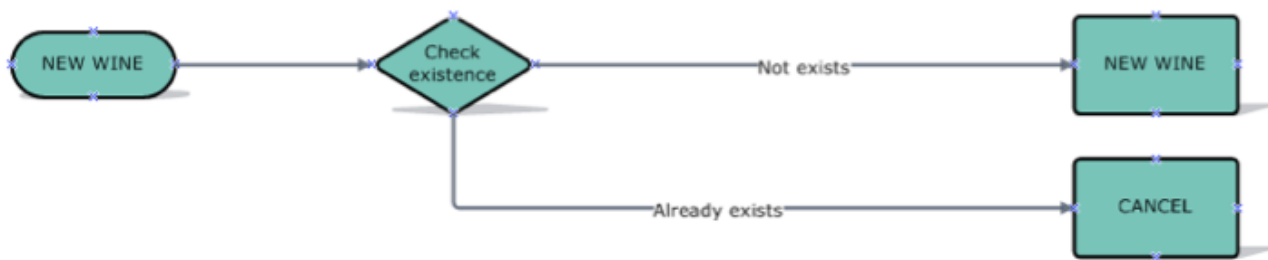
68. Delete a menu



@param MenuId

- Select **MenuLabel** into the list box of menus
- Click on the delete menu button
- Recuperation of **MenuId**
- Delete data menu of the database
- Display an pop up (« **MenuLabel** is delete »)
- Update and display list box of menus

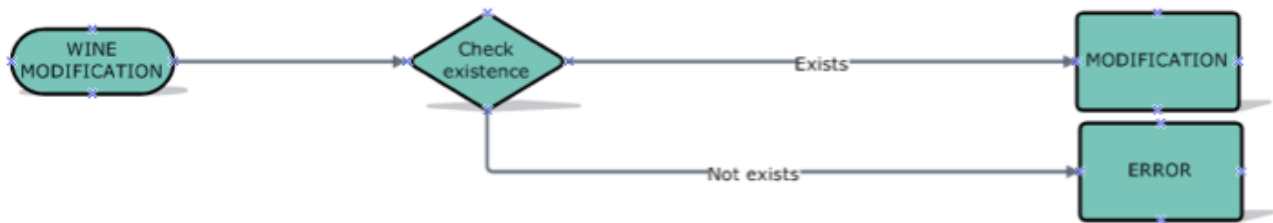
69. New wine



@param none

- Click on the new wine button
- Open a new window with field correspond to wine
- Input disable button 'OK'
- Input new datas into fields
- **WinePriceTTC** is calculate automatically with **WineTVARate**
- **WineProcurementAmount** is calculate automatically with **WineProcurementThreshold**
- **WineNbInStock** is facultative
- If is the « Menu Label » field
 - Call M14 (check **WineLabel** existence)
 - If the wine exist
 - Display an error pop up (« This wine label is already used »)
 - Enter a new **WineLabel** into the field « Wine Label » or cancel operation
- While all field are not informed except facultative field, the Ok button remains disable
- Click on the ok button
- Insert data of the database :
 - **WineLabel** ;
 - **WineColor** ;
 - **WineYear** ;
 - **WineCountry** ;
 - **WineRegion** ;
 - **WineTemperature** ;
 - **WineAlcoholContent** ;
 - **WineSparklingOrFlat** ;
 - **WineBottleType** ;
 - **WineBottleVolume** ;
 - **WineGettingDate** ;
 - **WinePriceHT** ;
 - **WinePriceTTC** ;
 - **WineTVARate** ;
 - **WineProcurementAmount** ;
 - **WineProcurementThreshold** ;
 - **WineNbInStock** (default zero) ;
- Update and display list box of wines

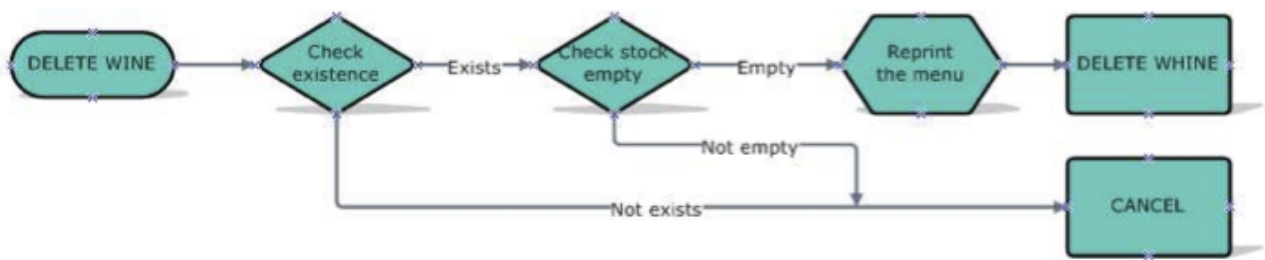
70. Modification of a wine



@param WinId

- Select **WineLabel** into the list box of wines
- Click on the modification wine button
- Recuperation of **WinId**
- Retrieve wine informations
- Open a new window
- Input old wine data into field of window
- **WineNbInStock** is not editable
- While no change as done, disable the ok button
- Input new datas into fields
- If is the « Wine Label » field
 - Call M15 (check **WineLabel** existence excepted him (param this))
 - If the wine exist
 - Display an error pop up (« This wine label is already used »)
 - Enter a new **WineLabel** into the field « Wine Label », don't change the **WineLabel**, or cancel operation
 - Else
 - Enable the Ok button
- Click on the ok button
- Update datas of the database

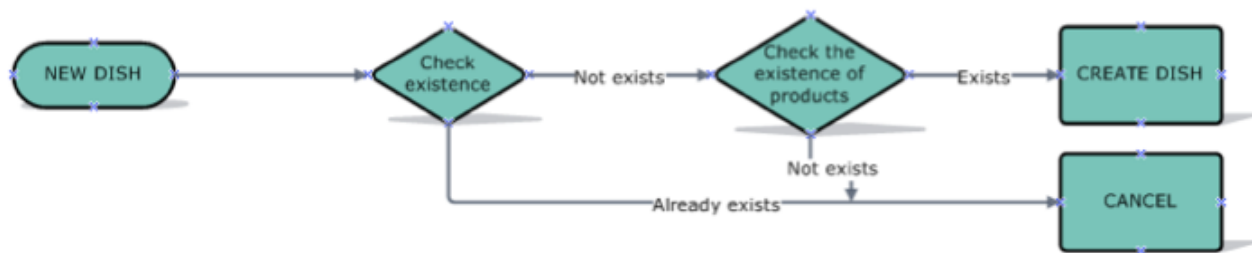
71. Delete a wine



@param Wineld

- Select **WineLabel** into the list box of menus
- Click on the delete menu button
- Recuperation of **Wineld**
- Call M16 (Check the Number in Stock of the **WineLabel**)
- If **WineNbInStock** is equals zero
 - Delete data wine of the database
 - Display an pop up (« **WineLabel** is delete »)
 - Update and display list box of wines
- Else
 - Display an error pop up (« **WineLabel** cannot be deleted : quantity in stock »)

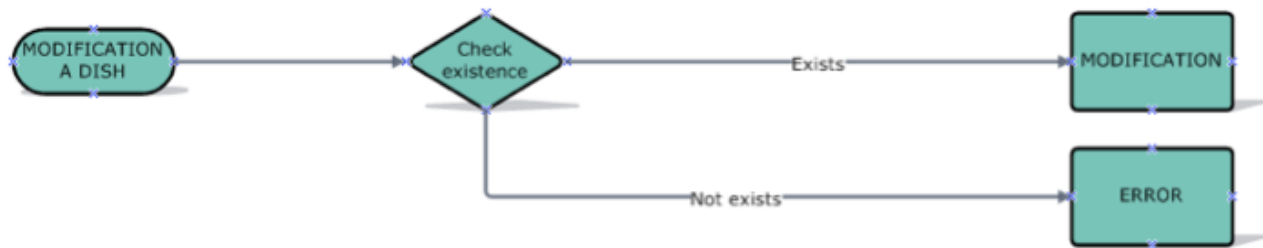
72. New dish



@param none

- Click on the new dish button
- Open a new window with field correspond to dish
- Display all product of hotel into the list box of products
- Input disable button 'OK'
- While all field are not informed and no product are selected, the Ok button remains disable
- Input new datas into fields
- **DishPriceTTC** is calculate automatically with **DishTVARate**
- If is the « Dish Label » field
 - Call M17 (check **DishLabel** existence)
 - If the dish exist
 - Display an error pop up (« This dish label is already used »)
 - Enter a new **DishLabel** into the field « Dish Label » or cancel operation
- Select all products composed the new dish with the add button and delete button
- Update and display list boxes of product
- Click on the ok button
- Insert data of the database :
 - **DishPriceHT** ;
 - **DishPriceTTC** ;
 - **DishTVARate** ;
 - **DishLabel** ;
 - All product component in dish.
- Update and display list box of dishes

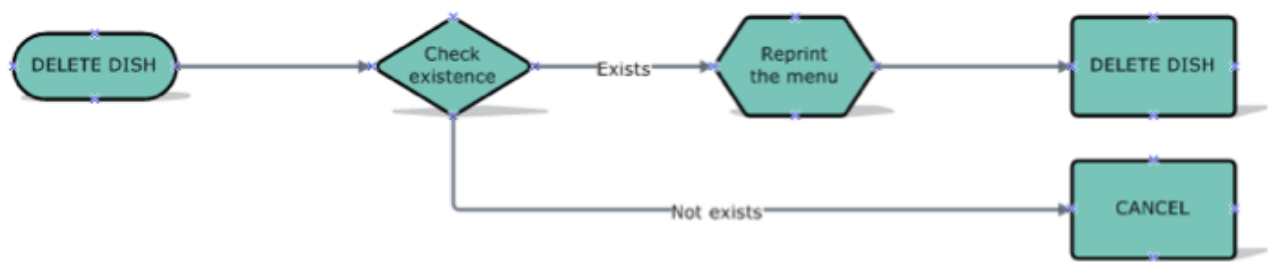
73. Modification of a dish



@param DishId

- Select **DishLabel** into the list box of dishes
- Click on the modification dish button
- Retrieve of **DishId**
- Open a new window
- Input old dish data into field of window
- While no change as done, disable the ok button
- Input new datas into fields
- If is the « Dish Label » field
 - Call M18 (Check the existence of **DishLabel** excepted him (param this))
 - If the dish exist
 - Display an error pop up (« This dish label is already used »)
 - Enter a new **DishLabel** into the field « Dish Label », don't change the **DishLabel**, or cancel operation
 - Else
 - Enable the Ok button
- Add or delete products with the add button and delete button
- Update and display list boxes of products
- Click on the ok button
- Update data of the database

74. Delete a dish



@param DishId

- Select **DishLabel** into the list box of dishes
- Click on the delete dish button
- Retrieve of **DishId**
- Delete data dish of the database
- Display an pop up (« **DishLabel** is delete »)
- Update and display list box of dishes

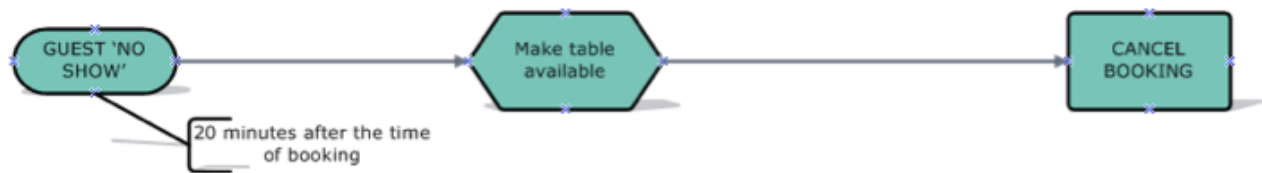
75. Ask an invoice



@param TableId

- Select the **TableId** into the list box of tables
- Click on invoice button
- Retrieve **GuestId** of the **TableId**
- Calculate the total cost of the temporary bill
- Insert on the bill of the guest into database :
 - **BillLineDate**
 - **BillLinePrice**
- Print temporary bill

76. A guest no show



@param none

- If a table is available 20 minutes after the DateOfBeginning of the booking
 - If there isn't booking after now and the **RestaurantSittingMediumLast** for table
 - Update the table availability on database to « available »
 - Transfer GuestName GuestSurname and DateOfBeginning to *Mev28 (Cancel a booking for a service)*

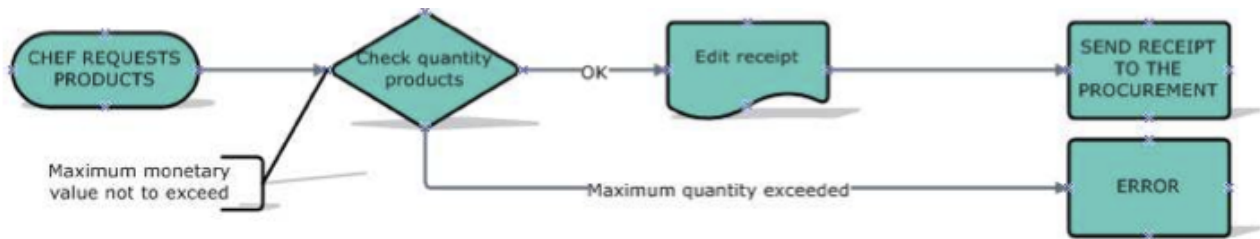
77.End of a day



@param none

- For all booking of each restaurant
 - Delete all booking of the day for the restaurant

78. The chef requests products



@param none

- Click on procurement button
- Open a new window
- Display all product of hotel into the list box of products
- Input disable button « Add »
- While a product and a quantity are not informed, the « Add » button remains disable
- Select the product into the list box of products
- Retrieve **ProductId**
- Retrieve and display product informations
- Calculate **ProductPurchasePrice** with the quantity asked (default one)
- Input data into the « Quantity » field
- Recalculate the **PRLineTotalCostTTC** with quantity and **ProductPurchasePrice**
- Click on the add button
- If the **DeptBudget** is inferior to the **PRLineTotalCostTTC**
 - Display an error pop up (« Maximum quantity exceeded »)
- Else
 - Update the **DeptBudget** with the **PRLineTotalCostTTC**
 - If there is a **ProcurementRequestId** already used by the user
 - Input data into database :
 - **ProductId** ;
 - **ProductLabel** ;
 - **PRLineQuantity** ;
 - **PRLineTotalCostTTC**.
 - Else
 - Input data into database :
 - **ProcurementRequestDate** ;
 - **ProductId** ;
 - **ProductLabel** ;
 - **PRLineQuantity** ;
 - **PRLineTotalCostTTC**.

79. New receipt



@param LaundryKitId, LaundryKitDateOfArrival, LaundryKitNbTotalOfItems

- Click on “New bag” button
- Display “Add” disable button
- Display a new form :
 - Laundry kit identifier “**LaundryKitId**”
 - Date of arrival “**LaundryKitDateOfArrival**” : automatically filled with today’s date
 - Number of items “**LaundryKitNbTotalOfItems**”
 - Put the flag **LaundryKitTreaty** : to treat
- OnChange of fields : enable “Add” button
- Click on “Add” button
- Input data in database

80.A laundry request is ready



@param LaundryKitId

- Click on “Laundry kit treaty” button
- Display list of laundry kit who having their flag : to treat
- Select the line corresponding
- Retrieve **LaundryKitId**, **LaundryKitDateOfArrival**, **LaundryKitNbTotalOfItems** corresponding
- Display “Put on the bill” disable button
- Display a new form :
 - Laundry kit identifier “**LaundryKitId**” *not change value*
 - Date of arrival “**LaundryKitDateOfArrival**” *not change value*
 - Number of items “**LaundryKitNbTotalOfItems**” *not change value*
 - Price TTC “**LaundryKitTotalCostTTC**”
- OnChange of field “**LaundryKitTotalCostTTC**” : enable “Put on the bill” button
- Click on “Put on the bill” button
- Calculate the **LaundryKitTotalCostHt** with the **LaundryKitTVARate**
- Update data in database
 - Update flag “**LaundryKitTreaty**” to : treaty

81.Start work



@param none

- Click on the « laundry kit treated » button
- Open a new window
- Display all laundry kit of hotel into the list box of laundry kit with the flag «ready»
- If there is no laundry kit, display a pop up («there is no laundry kit ready today »)

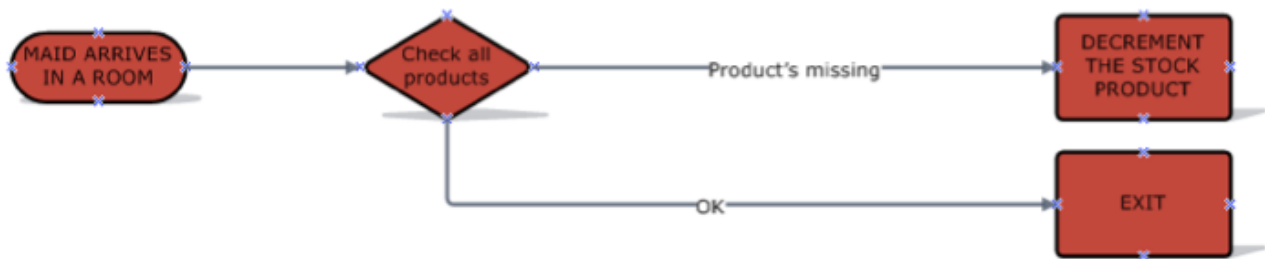
82.A bag has to be deposited



@param none

- Click on the « depose a laundry kit » button
- Open a new window
- Display all laundry kit of hotel into the list box of laundry kit with the flag «ready»
- If there is no laundry kit, display a pop up («there is no laundry kit ready today »)
- Select a laundry kit into the list box of the laundry kit
- Click on the depose button
- Retrieve the **LaundryKitId**
- Delete the laundry kit information to the database

83. A maid arrives in a room



@param none

- Select the RoomNumber into the list box of room
- Click on the « equipment of room » button
- Open a new window
- Retrieve **RoomId**
- Retrieve **RoomTypeId**
- Retrieve the equipment of the room type
- Display the list of equipment of the room type in a list
- If a product missing
 - Select the product into the list box of product
 - Click on add button
 - Retrieve ProductId
 - Decrement the ProductNumberInStock into the database