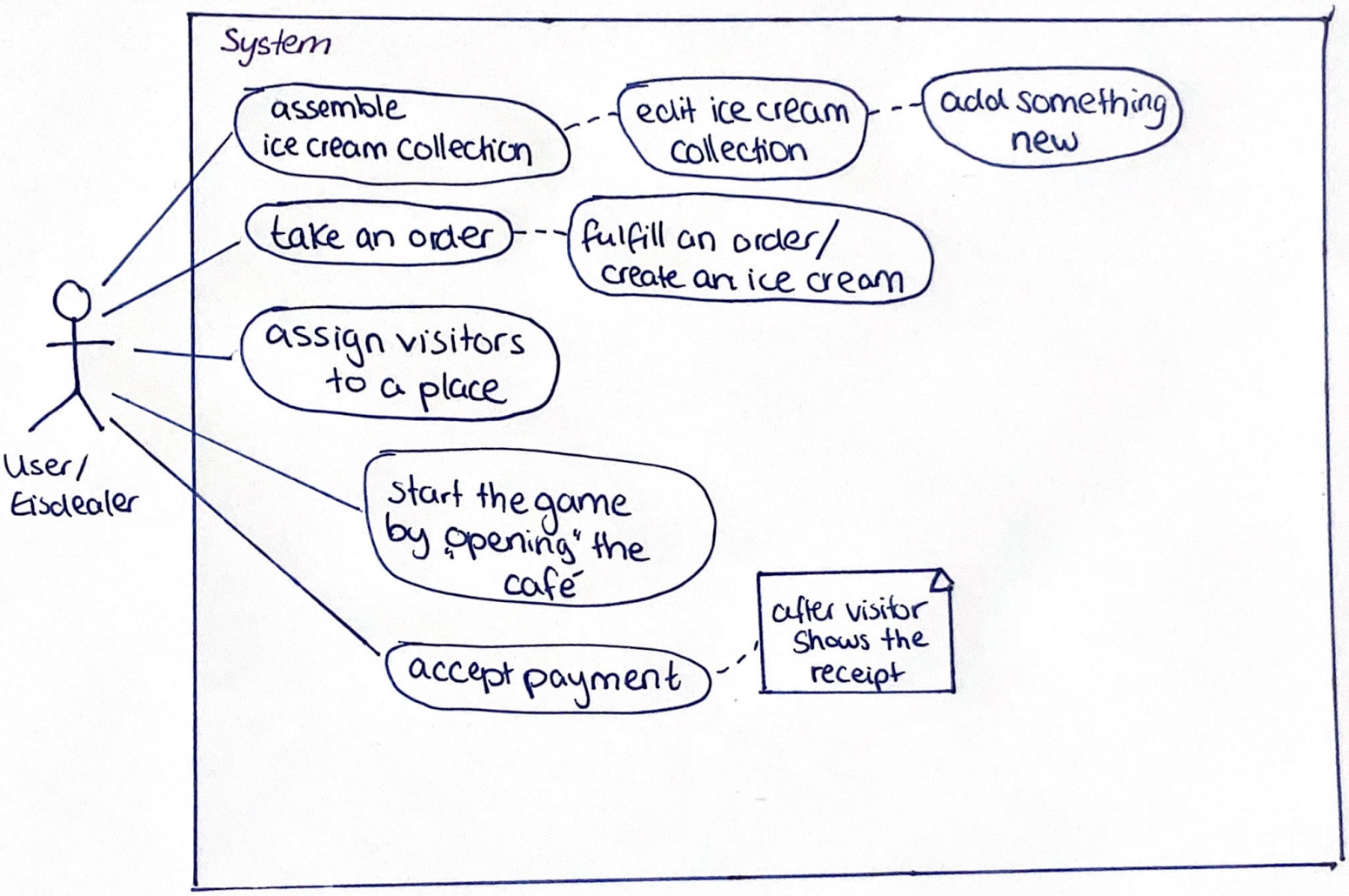
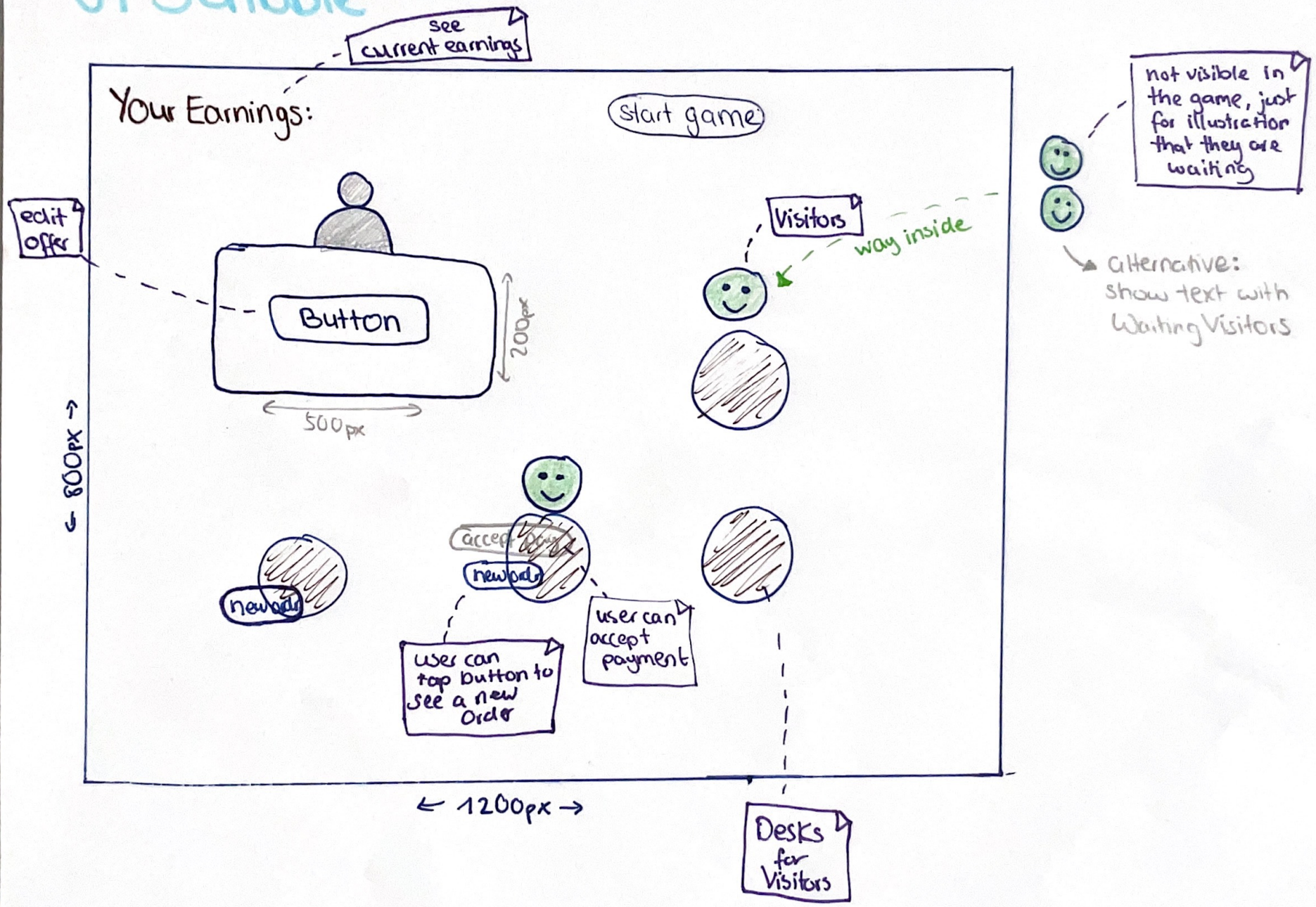
Konzeption

Eisdealer-Game

Use Case Diagram

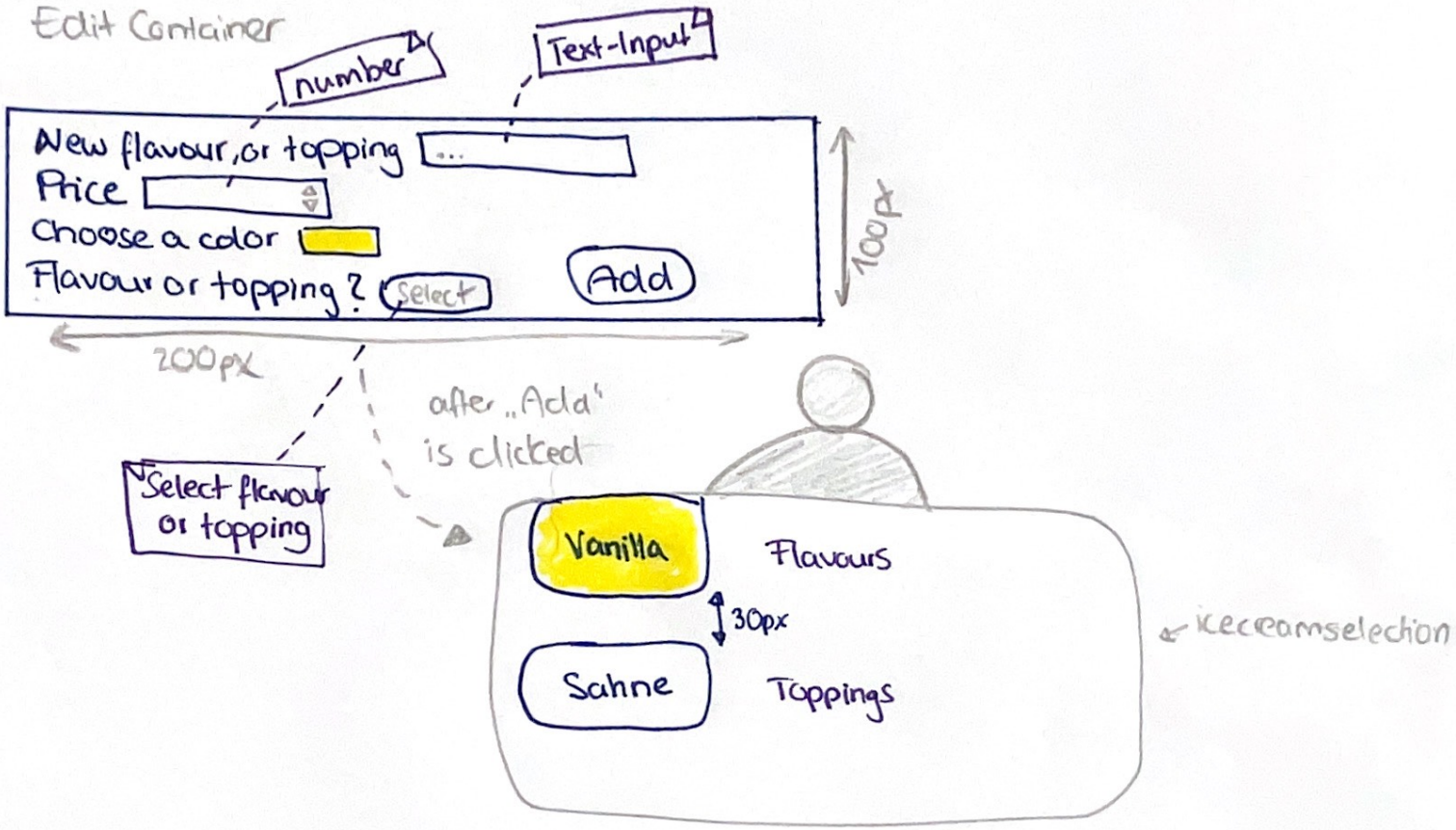


UI Scribble


see current earnings
Your Earnings:
start game
not visible in the game, just for illustration that they are waiting
edit offer
Button
200px
500px
Visitors
way inside
alternative: show text with Waiting Visitors
← 800px →
newOrder
user can tap button to see a new Order
user can accept payment
← 1200px →
Desks for Visitors

UI Scribble

Containers



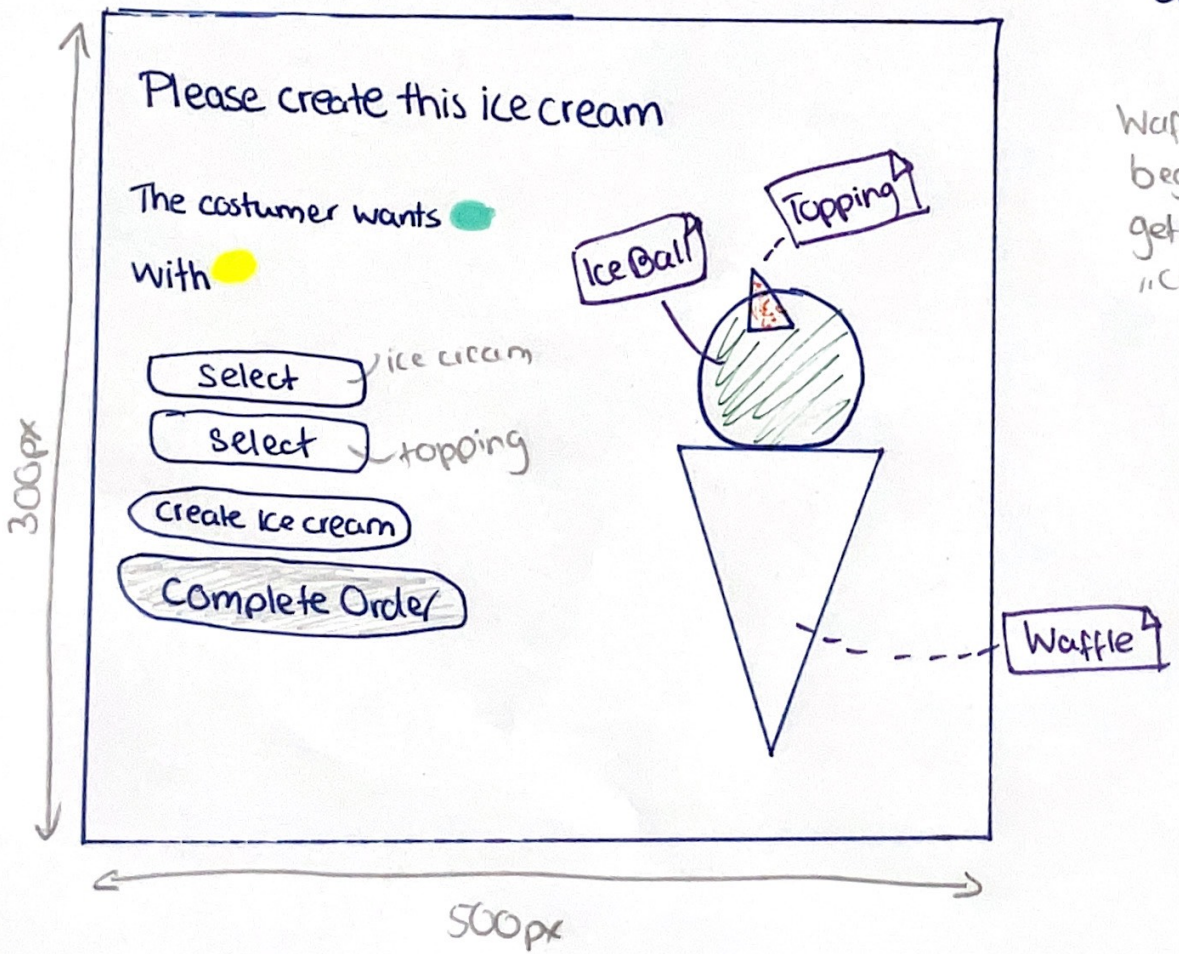
UI Scribble

Containers

= name of the ice cream order

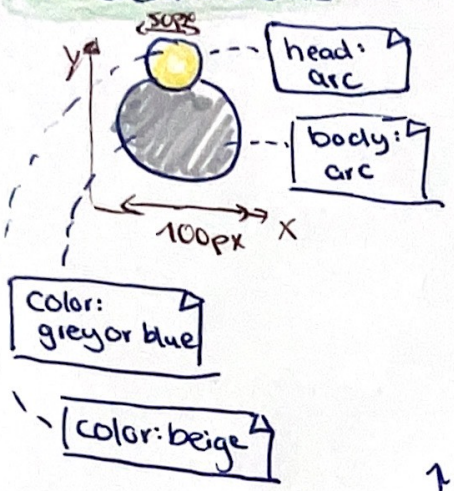
= name of the Topping order

FulfillorderContainer

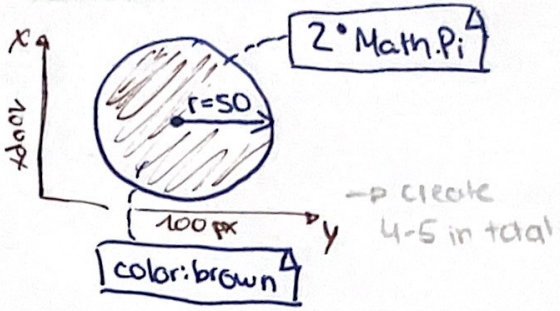


Waffle is visible from the beginning, ice cream and topping get visible, when the user clicks „create ice cream"

Ice Cream Dealer

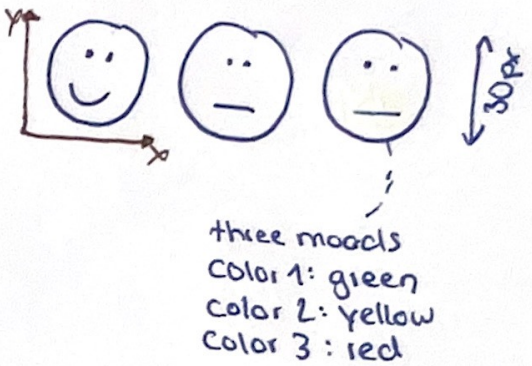


Standing Desk



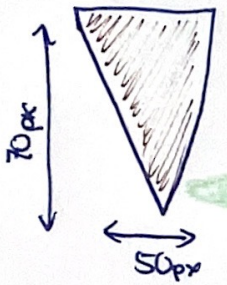
→ create 4-5 in total

Visitors



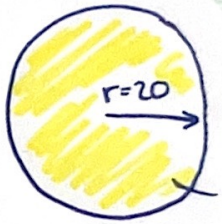
three moods
color 1: green
color 2: yellow
color 3: red

Waffle



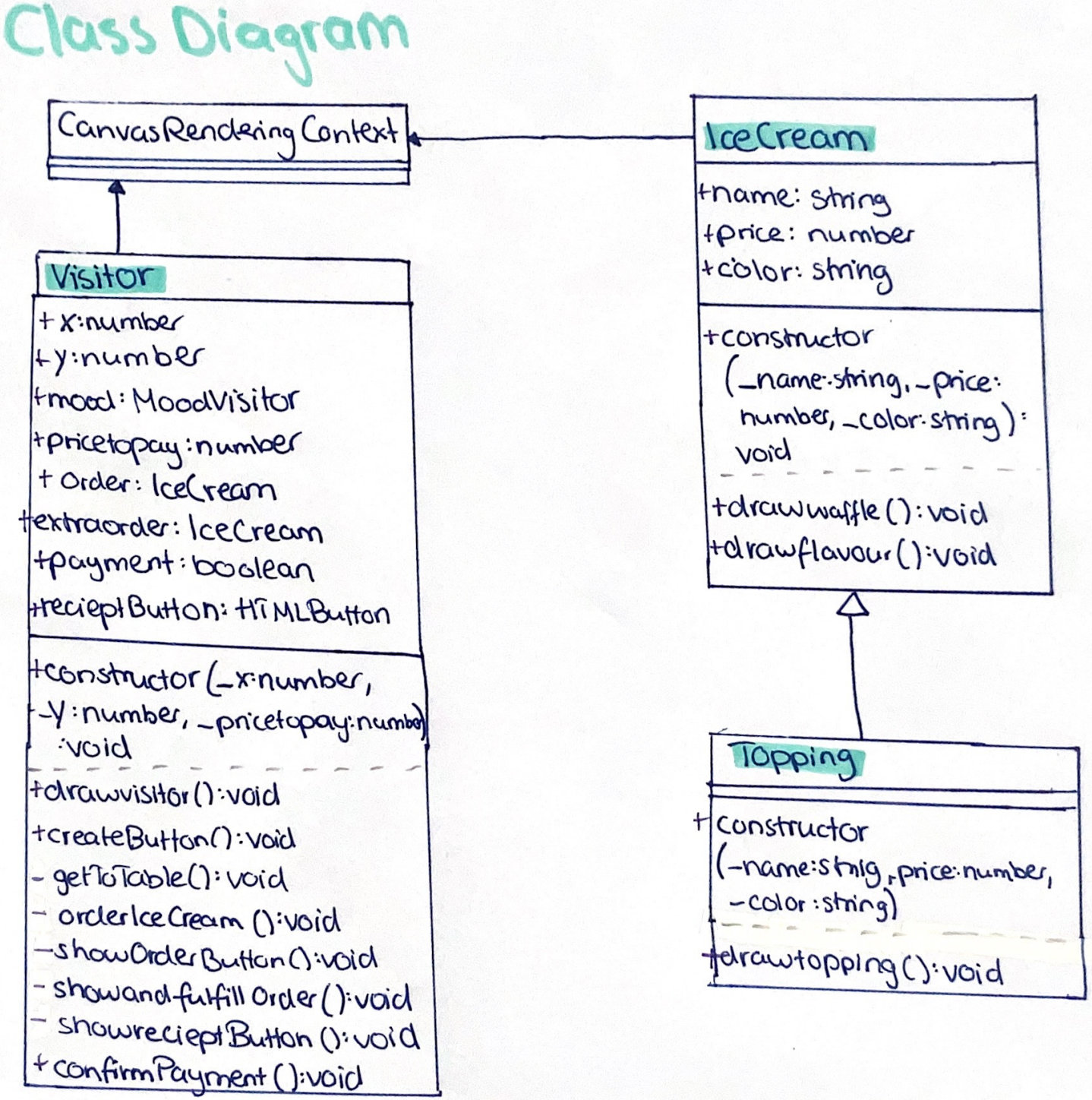
Topping

Ice Ball

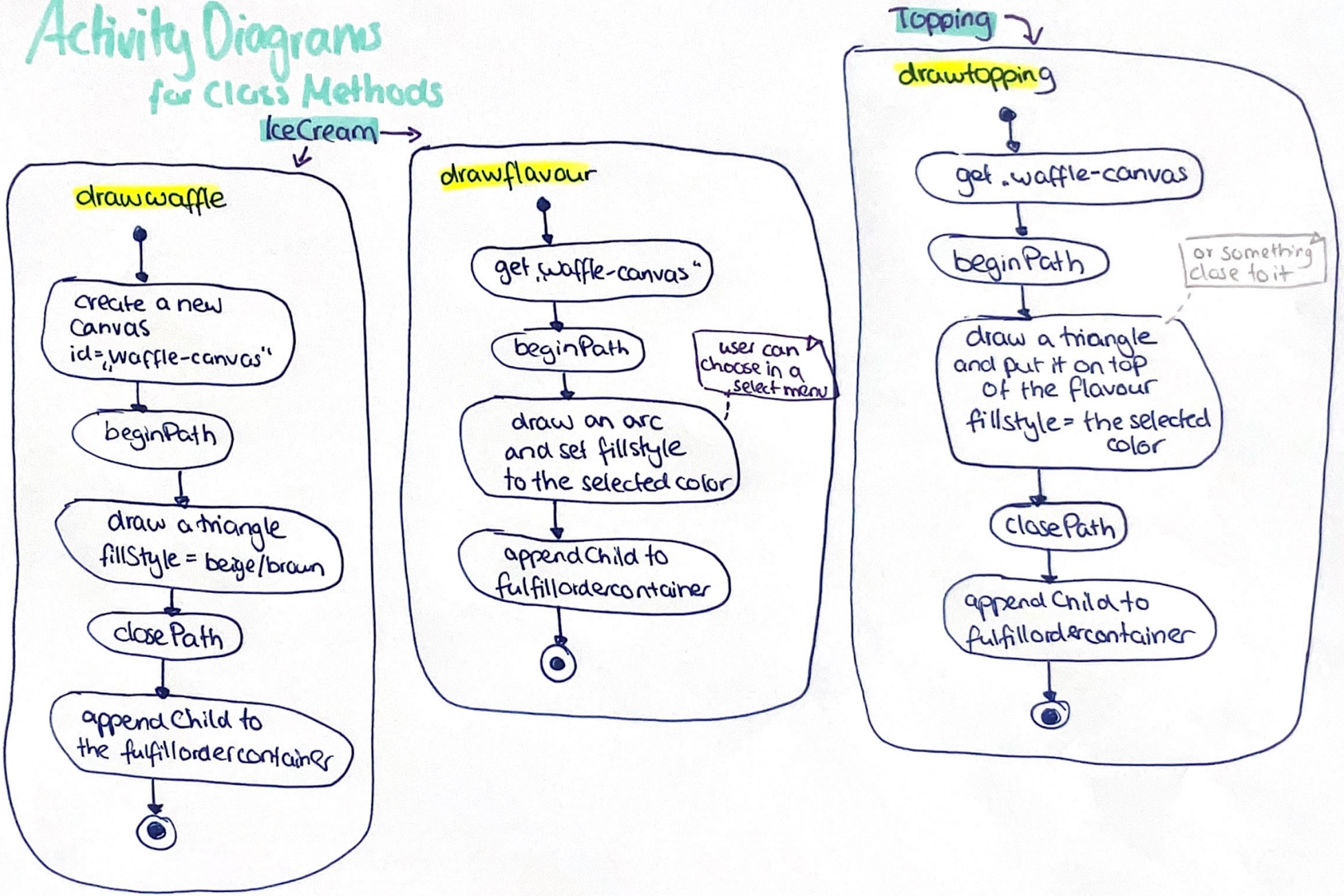


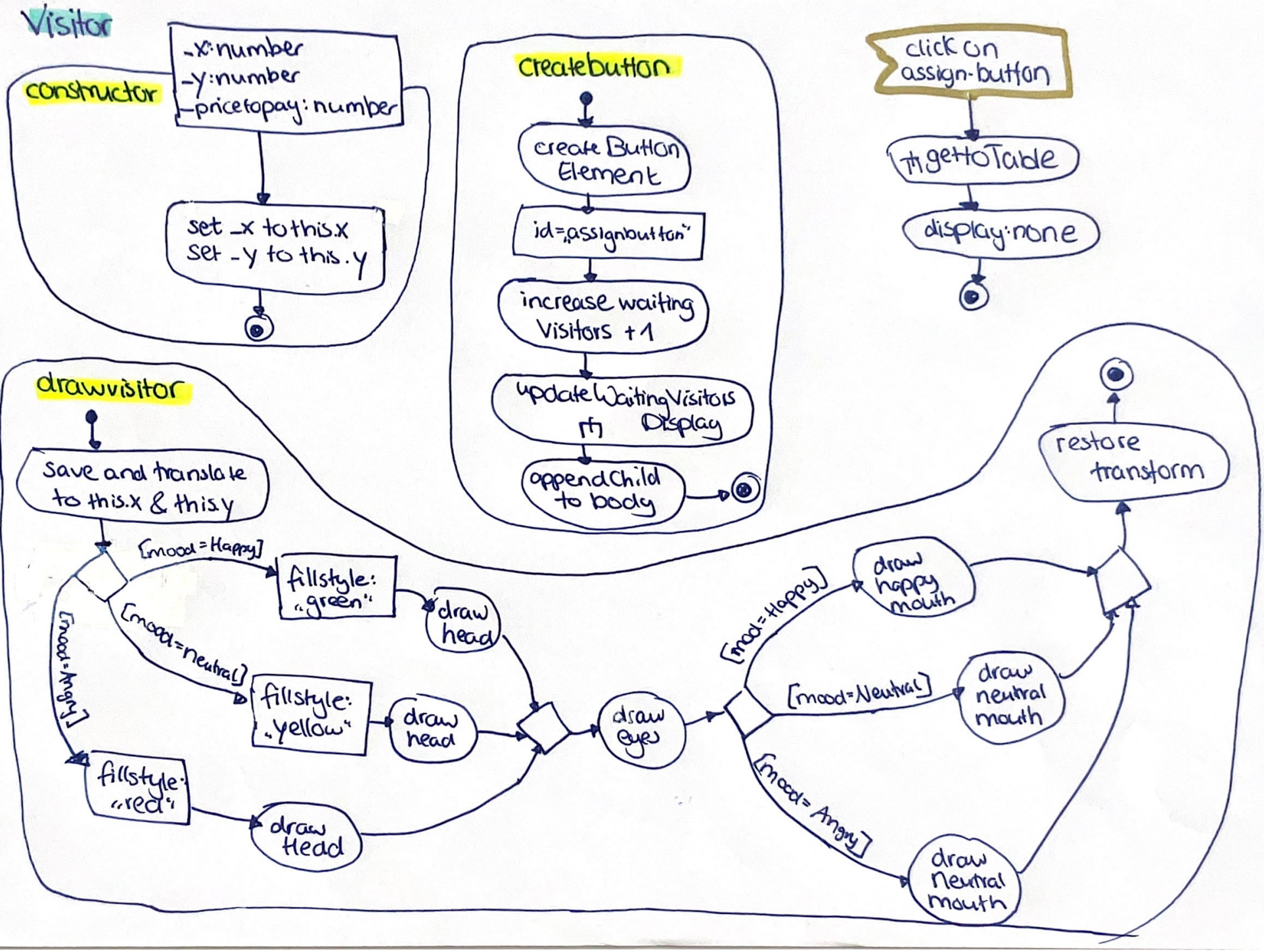
fillstyle: selected color

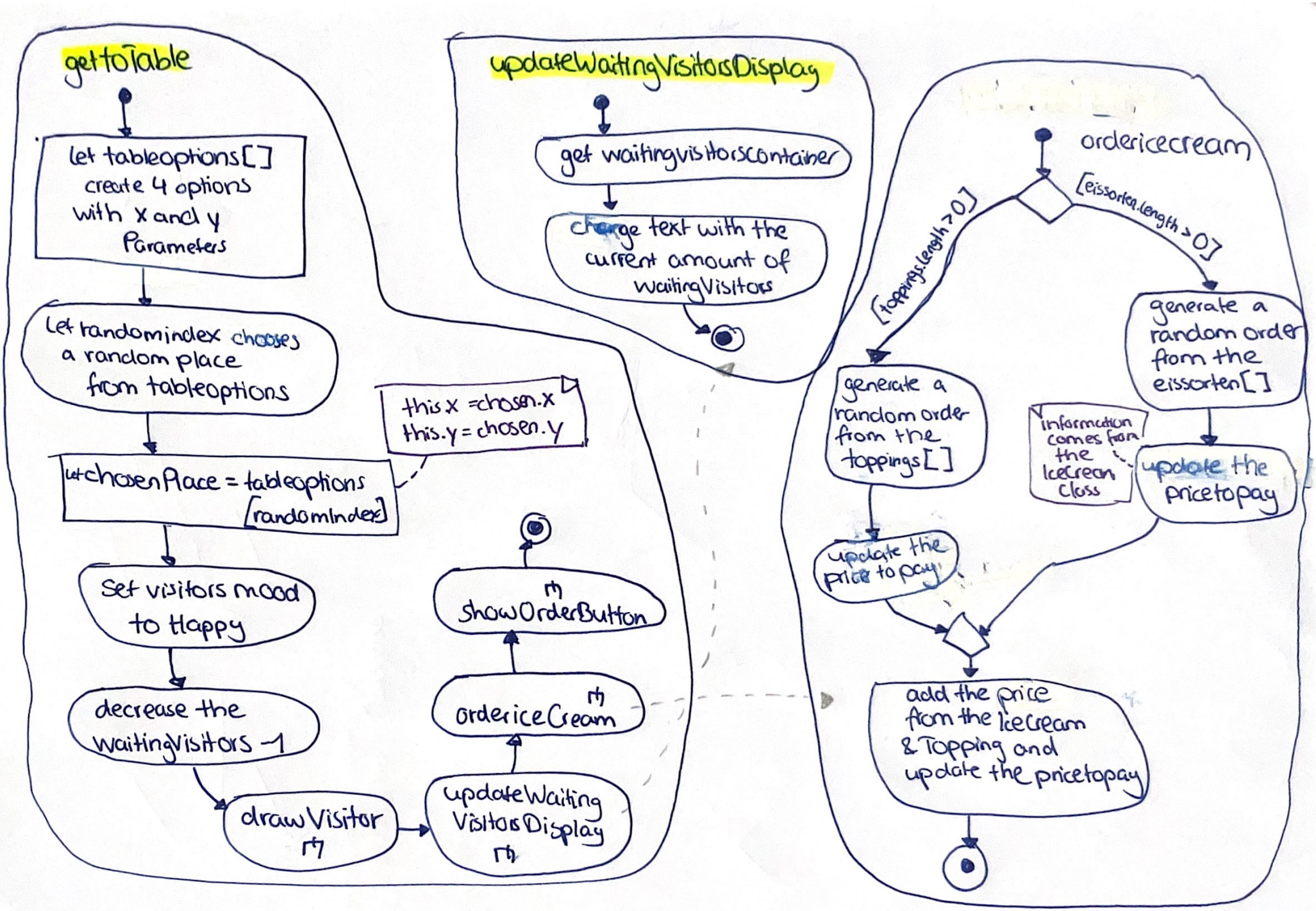
Class Diagram

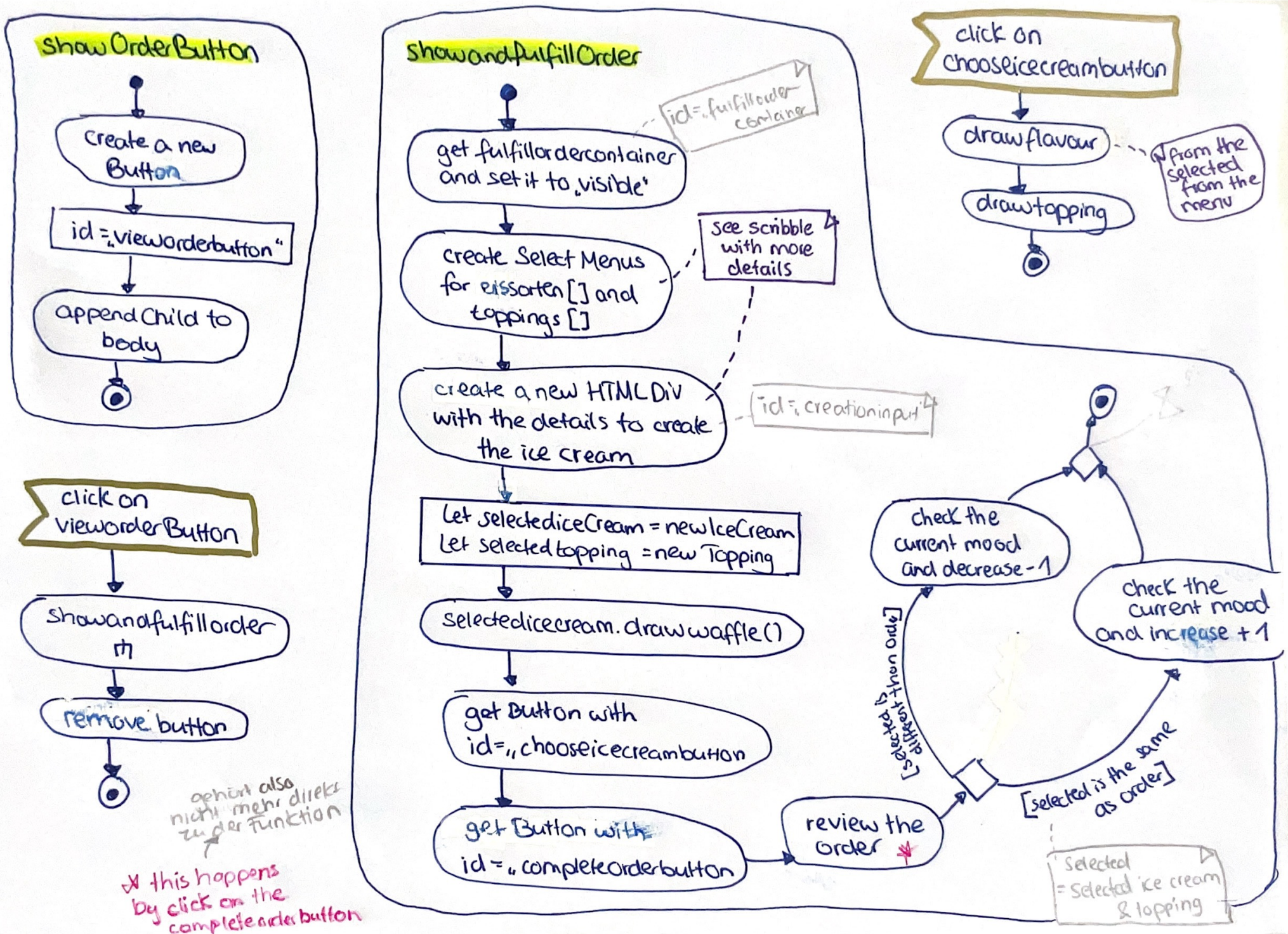


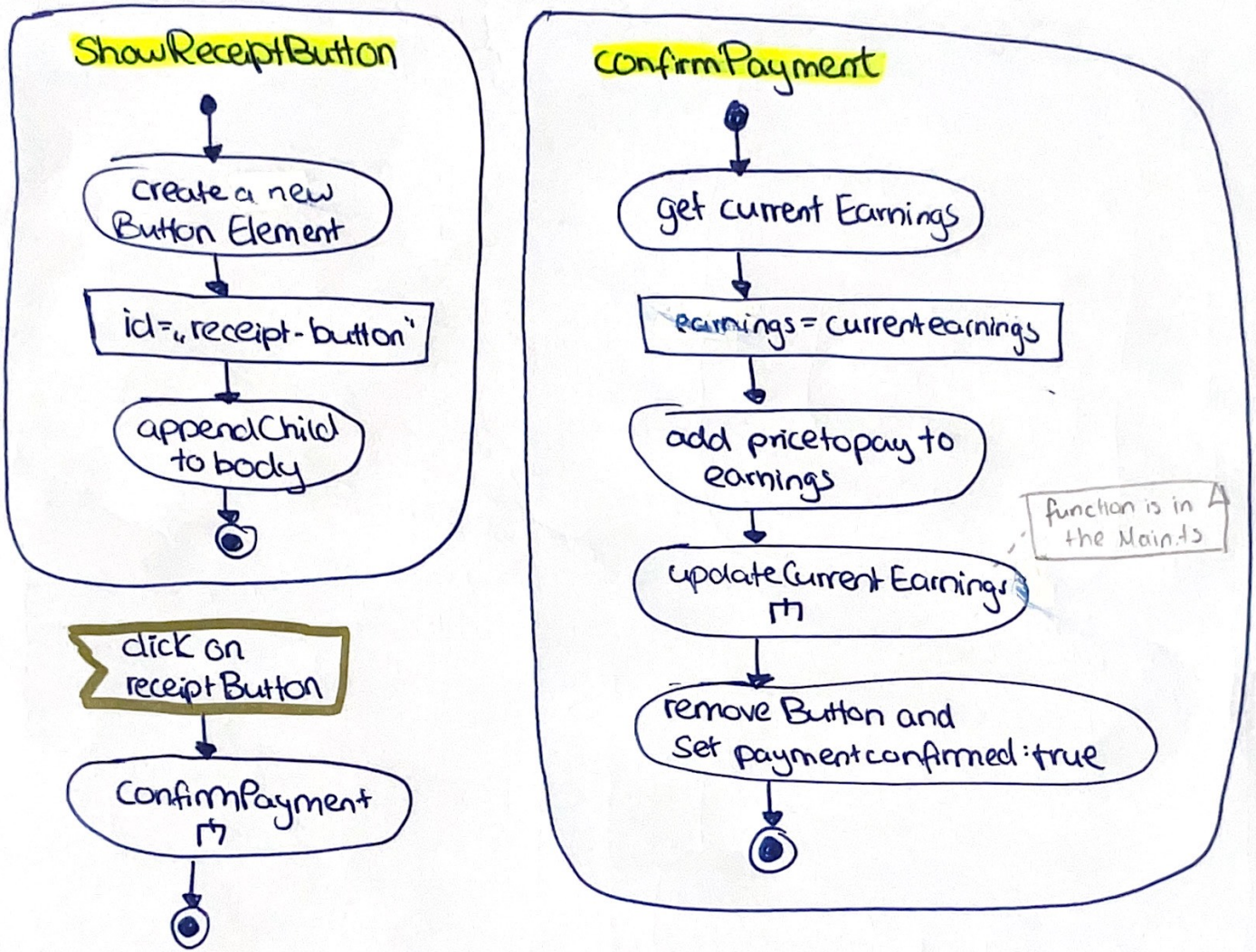
Activity Diagrams for Class Methods




Visitor
constructor
-x:number
-y:number
-pricetopay: number
set _x to this.x
set _y to this.y
createbutton
create Button Element
id="assignbutton"
increase waiting Visitors +1
updateWaitingVisitors fn Display
appendChild to body
click on assign-button
#gettoTable
display:none
drawvisitor
save and translate to this.x & this.y
[mood=Happy]
fillstyle: "green"
draw head
[mood=neutral]
fillstyle: "yellow"
draw head
[mood=Angry]
fillstyle: "red"
draw Head
draw eyes
[mood=Happy]
draw happy mouth
[mood=Neutral]
draw neutral mouth
[mood=Angry]
draw Neutral mouth
restore transform

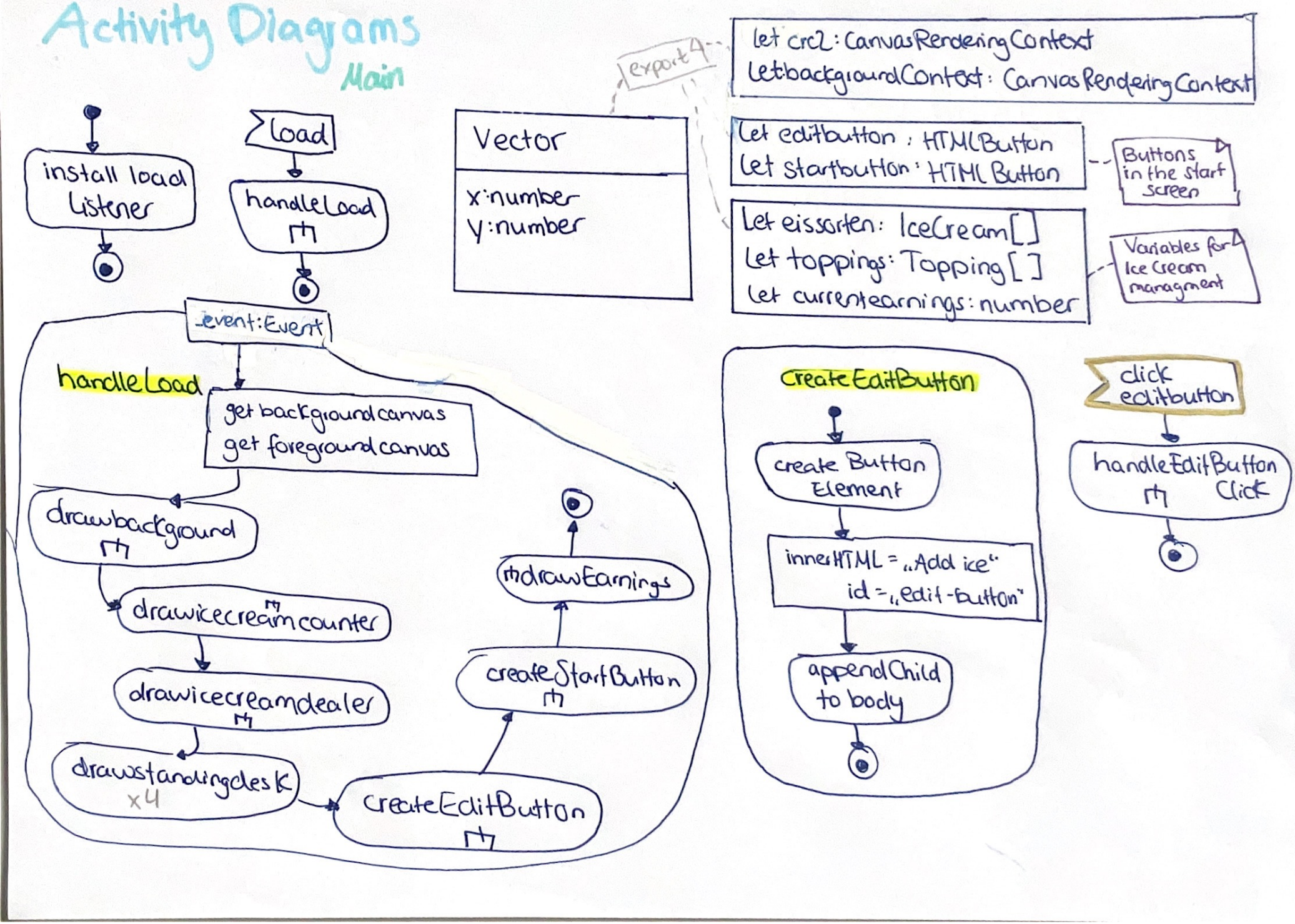

gettoTable
let tableoptions[]
create 4 options
with x and y
Parameters
Let randomindex chooses
a random place
from tableoptions
this.x =chosen.x
this.y = chosen.y
let chosenPlace = tableoptions
[randomIndex]
Set visitors mood
to happy
decrease the
waitingVisitors -1
drawVisitor
updateWaiting
VisitorsDisplay
ordericeCream
showOrderButton
updateWaitingVisitorsDisplay
get waitingvisitorscontainer
change text with the
current amount of
waitingVisitors
ordericecream
[toppings.length > 0]
[eissorten.length > 0]
generate a
random order
from the
toppings[]
generate a
random order
from the
eissorten[]
information
comes from
the
IceCream
class
update the
pricetopay
update the
price to pay
add the price
from the IceCream
& Topping and
update the pricetopay

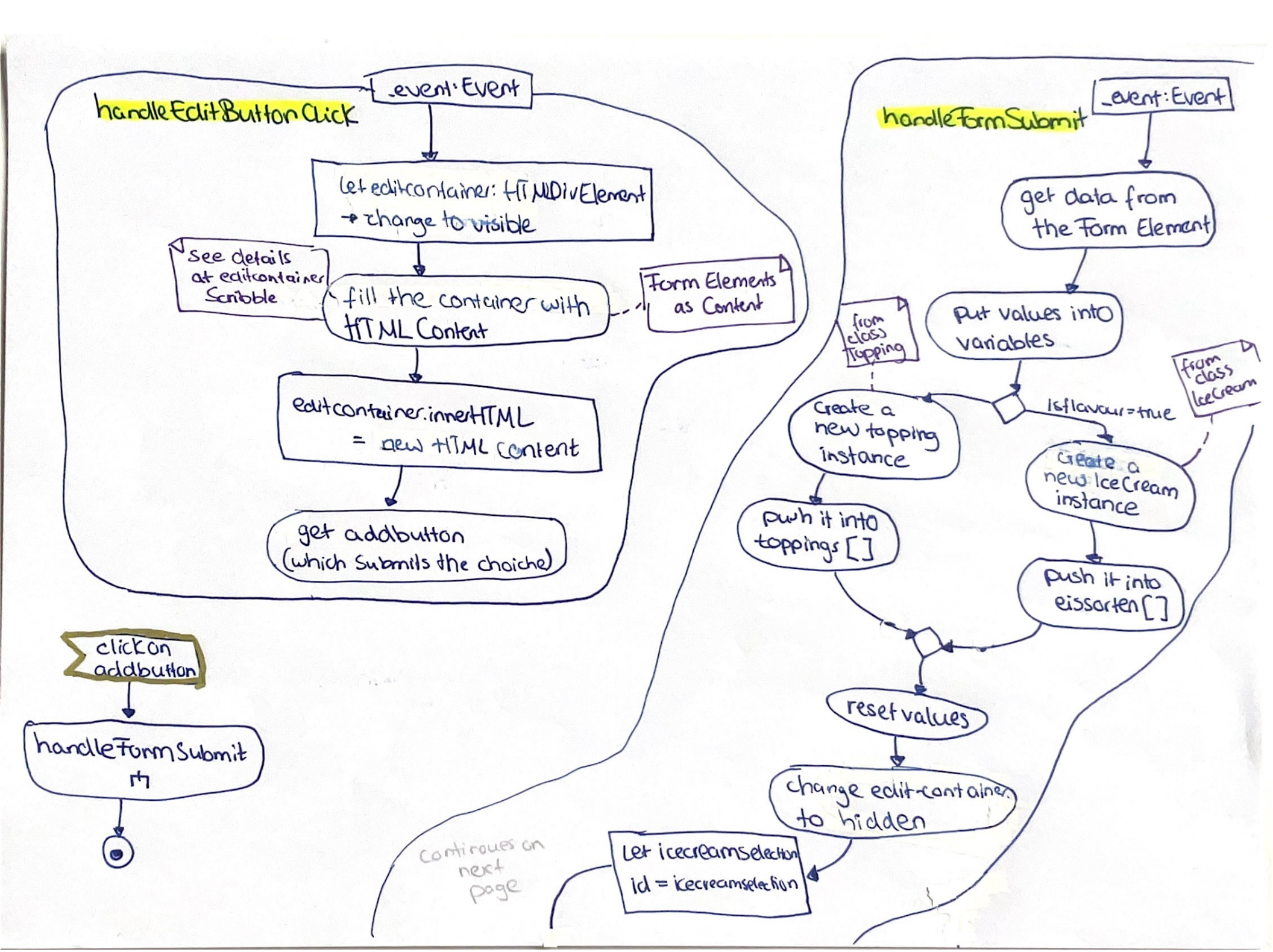

showOrderButton
create a new Button
id = „vieworderbutton“
appendChild to body
showandfulfillOrder
get fulfillordercontainer and set it to „visible“
id = „fulfillorder container
create Select Menus for eissorten [] and toppings []
see scribble with more details
create a new HTMLDiV with the details to create the ice cream
id = „creationinput“
Let selectediceCream = newIceCream
Let selectedtopping = new Topping
selectedicecream.drawwaffle()
get Button with id = „chooseicecreambutton
get Button with id = „completeorderbutton
review the order
click on chooseicecreambutton
drawflavour
drawtopping
from the selected from the menu
click on vieworderButton
showandfulfillorder
removebutton
check the current mood and decrease -1
check the current mood and increase +1
[selected is different than order]
[selected is the same as order]
Selected = selected ice cream & topping
gehört also nicht mehr direkt zu der funktion
* this happens by click on the completeorderbutton

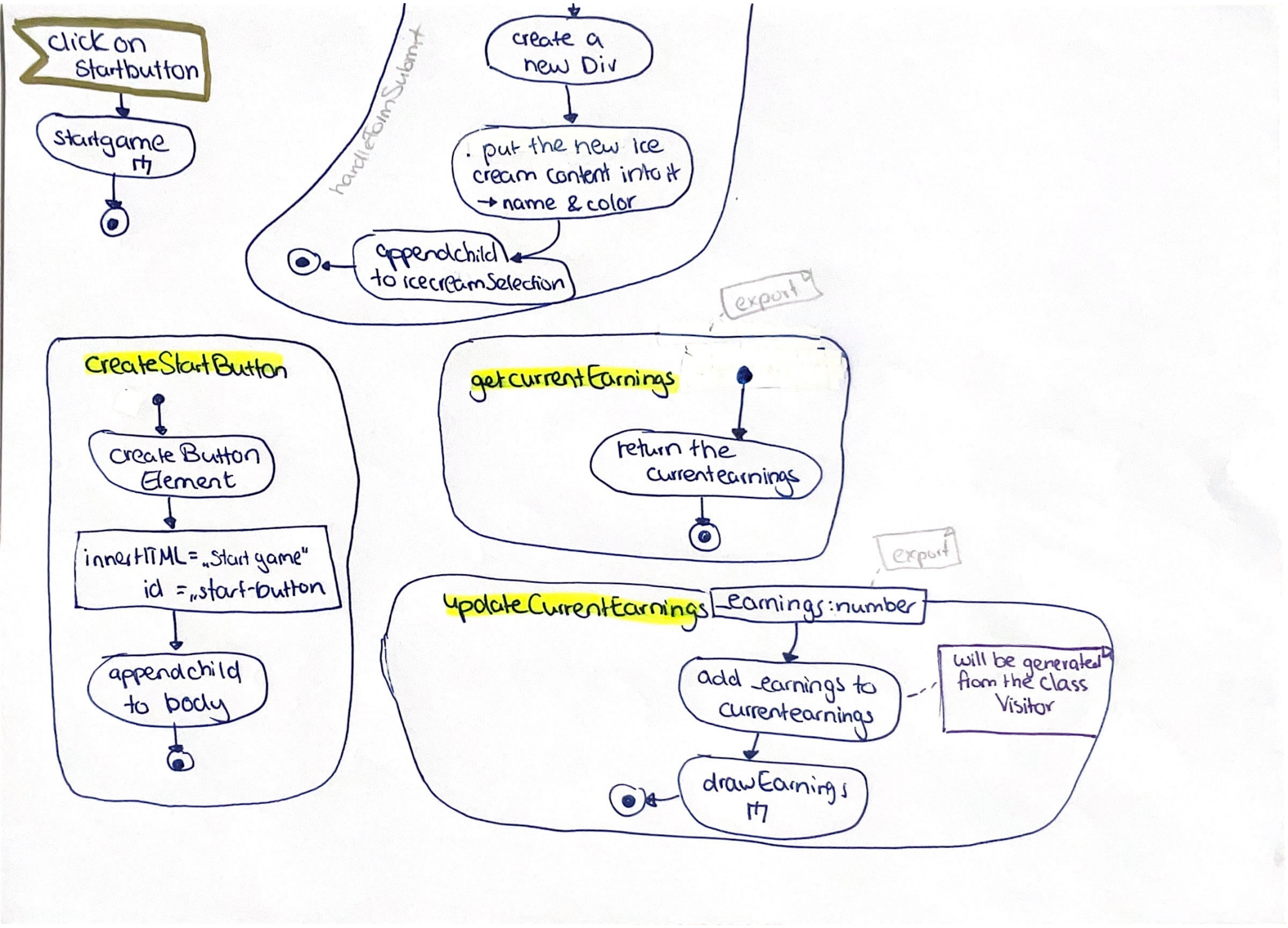

ShowReceiptButton
create a new Button Element
id="receipt-button"
appendChild to body
click on receiptButton
confirmPayment
confirmPayment
get current Earnings
earnings = currentearnings
add pricetopay to earnings
function is in the Main.ts
updateCurrentEarnings
remove Button and Set paymentconfirmed:true

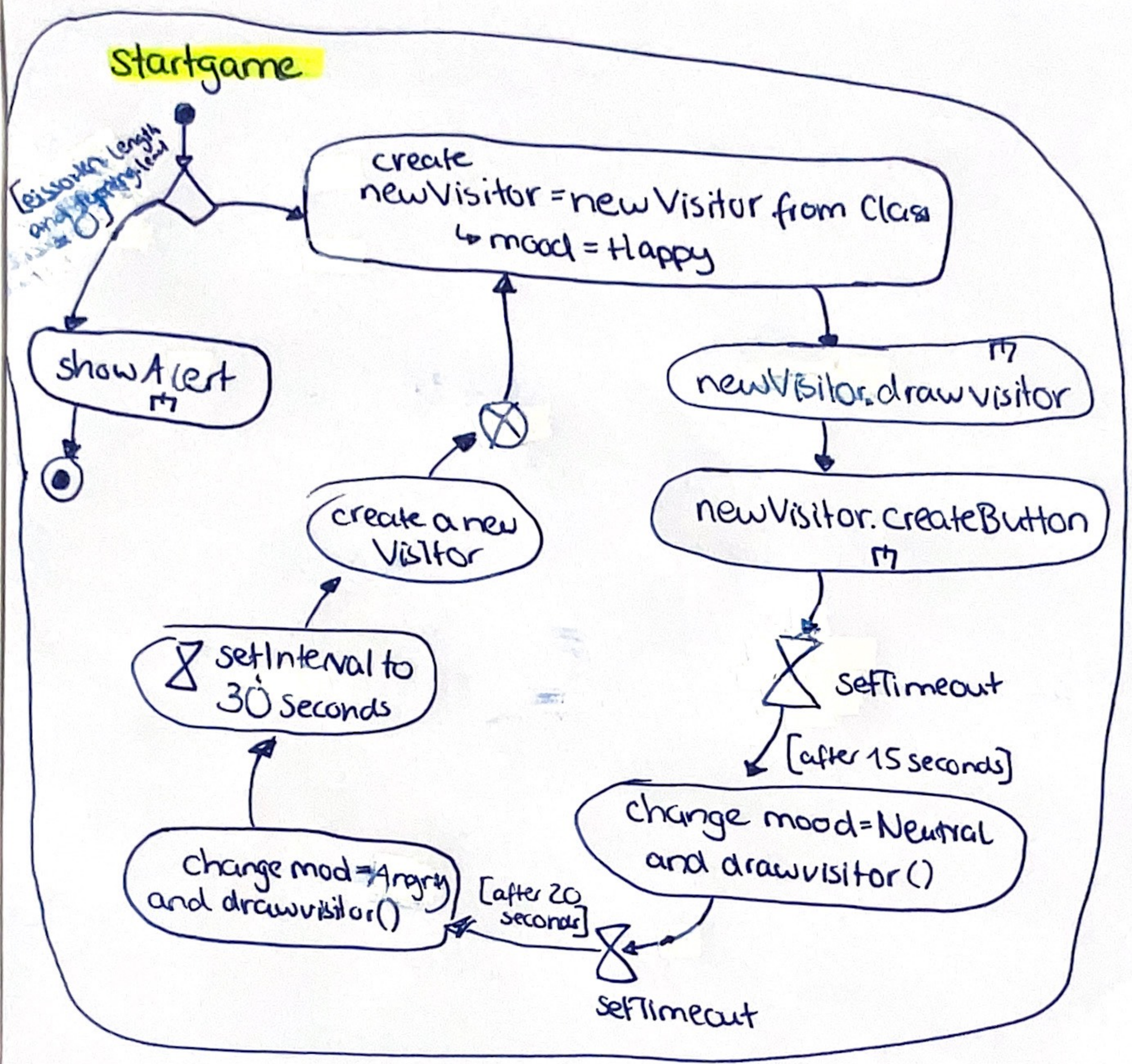
Activity Diagrams

Main




handleEditButtonClick
_event: Event
let editcontainer: HTMLDivElement
→ change to visible
See details at editcontainer Scribble
fill the container with HTML Content
Form Elements as Content
editcontainer.innerHTML = new HTML content
get addbutton (which Submits the choiche)
clickon addbutton
handleFormSubmit
handleFormSubmit
_event: Event
get data from the Form Element
Put values into variables
from class Topping
from class IceCream
Create a new topping instance
isFlavour=true
Create a new IceCream instance
puh it into toppings []
push it into eissorten []
reset values
change edit-container to hidden
Let icecreamselection id = icecreamselection
Continoues on next page


click on startbutton
startgame
handleFormSubmit
create a new Div
. put the new ice cream content into it → name & color
appendchild to icecreamSelection
export
CreateStartButton
createButton Element
innerHTML = „Start game"
id = „start-button
appendchild to body
getcurrentEarnings
return the currentearnings
export
updateCurrentEarnings
_earnings:number
add _earnings to currentearnings
will be generated from the Class Visitor
drawEarnings


Startgame
create
newVisitor = new Visitor from Class
mood = Happy
showAlert
newVisitor.drawvisitor
newVisitor.createButton
setTimeout
[after 15 seconds]
change mood = Neutral
and drawvisitor()
[after 20
seconds]
setTimeout
change mod = Angry
and drawvisitor()
setInterval to
30 seconds
create a new
Visitor


ShowAlert
Show alert
"Fill the counter..."

Static Background Elements