

Überlegungen Eisdealer :

Welche Objekte interagieren miteinander?

- Start Day - Button :
 - Kunden erscheinen, betreten Laden
(hier schon Bedingung für Anzahl Kunden im Laden)
 - Kunden geben Bestellung auf
 - ↳ mögl. Bestellungen müssen von Maggi DB gekollt werden (inkl. Preis)
 - Bestelldaten müssen auf Canvas dargestellt werden
 - Stimmung wird fortlaufend geändert
 - Ab 10s warten generiert
 - Ab erhält das Eis glücklich
- Select Elemente + Serve Button :
 - Select Elemente speichern Eingabe bis Serve Button gedrückt wird → dann löscht es Eingabe
 - Bei anderer Auswahl über Select Elemente als Bestellung + Klick auf Serve Button wird die Stimmung des Kunden verändert
 - Select Elemente verändern Eis auf Canvas
 - ↳ Bedingungen: `If Stenciatella { crea.fillStyle="white" }`
 - Alle Elemente Waffel, Sorte, Soße, ...
 - Klick auf Serve :
 - Kunde läuft nach links
 - Ess-Balkendiagramm erscheint + Eis erscheint
 - Stimmung des Kunden verändert sich
 - Nächster Kunde läuft vor und bestellt
 - Alle Kunden laufen weiter
 - Bestellung des aktuellen Kunden verschwindet
 - Bestellung des neuen Kunden erscheint

Uunde ist fertig mit essen:

- zwischen 5-10s Esszeit
- Balkendiagramm + Eis verschwindet
- Uunde läuft raus
- Wasserbestand erhöht sich
- Preis über x€, Uunde gewinnt

Tagesablauf:

- Ab Start-Day 1 Minute Tag (set Timeout)
- Nach 1min können keine Uunden mehr den Laden betreten
- Färeschild Schild + Close Store Button erscheinen
- Uunde auf Close Store:
 - Dealer geht
 - Kasse = 0€
 - Hintergrund schwarz
 - Close store Button ~~ist~~ verschwindet
 - Start Day Button erscheint
- ↳ Dealer erscheint erst bei Ulich auf Start Day

Was ist nicht möglich beim Eisdealer-Game:

- Mehrere Uunden können gleichzeitig bestellen
- Uunden können vor Start Day nach Close Store bestellen
- ~~Uunden~~ Uunden können nicht mehrere Eis bestellen
- Nur beschränkte Anzahl an Uunden, max. 1 Sauce/Topping
- Uunden können den Laden nicht betreten wenn 3 Personen in der Schlange stehen, erst wenn auf Serve gelichtet wurde und ein Uunde geht (rotierendes System) Array für Uunden + für Uunden in Warteschlange + Essende Uunden (max. 2)
- Es können nicht mehr als zwei Uunden essen → d.h. zwei Positionen für essende Uunden

Welches Erlebnis soll der Nutzer bei der Anwendung haben?

- Der Nutzer soll das Gefühl vermittelt bekommen im Interesse des Eisdealers zu handeln.

Er soll versuchen die individuellen Bedürfnisse der Kunden zu befriedigen und ein profitables Geschäft zu führen.

Der Nutzer muss durch Interaktion mit dem Programm das Gefühl bekommen einen Laden zu führen, der ohne ihn den Bach runter gehen würde.

Anleitung für die Interaktion mit der Anwendung:

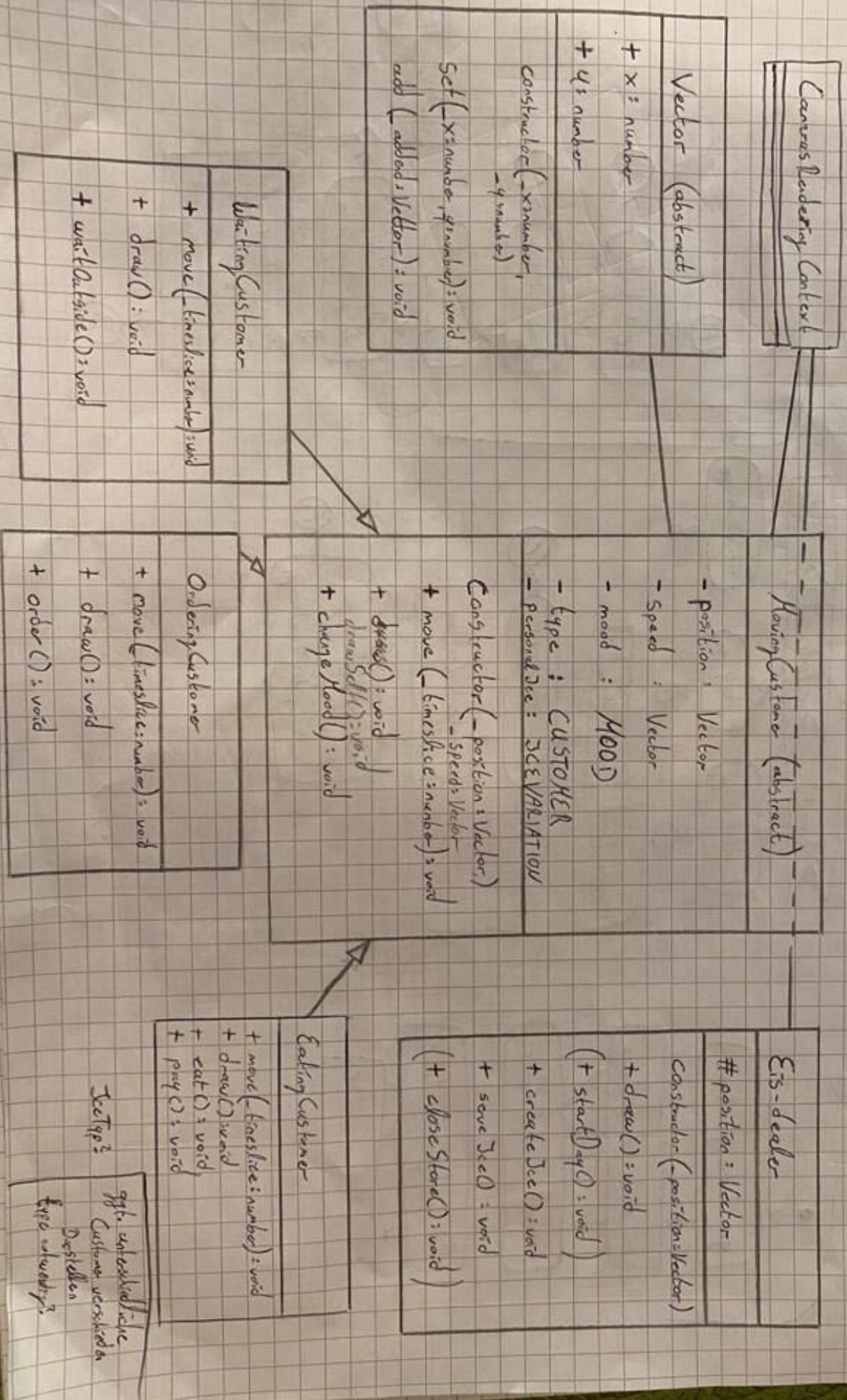
- Tag-Starten Button betätigen, um einen Tag in der Eisdiele zu simulieren (wird Tag)
 - Kunden stürmen ab diesem Zeitpunkt die Diele und geben Bestellungen auf
- Bestellung wird angezeigt
- Nutzer muss durch das Interagieren mit dem System (durch Buttons) die Bestellung fertigstellen
- Ist die Bestellung fertig kann über einen anderen Button die Bestellung den Kunden übergeben werden
 - An diesem Zeitpunkt gilt:
 - Ist die Bestellung richtig oder falsch → Stimmung des
 - Kunde bezahlt das Eis / Isst Eis → Kunden wird verändert
 - ↳ Vorrat/inventar steigt
- Kunde verlässt die Diele, nächster Kunde tritt an die Theke
 - selbes Spiel von vorn (d.h. Kunde wählt selbst random Eis aus)
- Nach einer gewissen Zeit ist der Tag zu ende
 - der Gewinn des Tages wird angezeigt
 - über einen Button kann der Nutzer die Eisdiele schließen (wird Nacht)

Wie, wann und wodurch müssen Funktionen aufgerufen werden?

- Durch Klick auf Start Day Button :
 - Day Funktion (Timeout, Kunden-Bestellung/-Bewegungen, ...) • Alle Draw Funktionen
- Handleload durch window load :
 - DrawCounter Funktion
 - Start Day Button
- Dealer / Nutzer muss nach Bestellung des Kunden, das Eis zusammenstellen, durch Klick auf Serve Button :
 - EatProcess Funktion → bei Ablauf : Cash Funktion / Go Funktion
 - AllKustomer-Move Funktion
 - Order Funktion
- Close Store Button ruft EndOfDay Funktion auf :
 - Keine Kunden mehr
 - Kasse wird = 0€
 - Dealer geht
 - Hintergrund schwarz
 - Close store Button verschwindet / Start Day Button erscheint

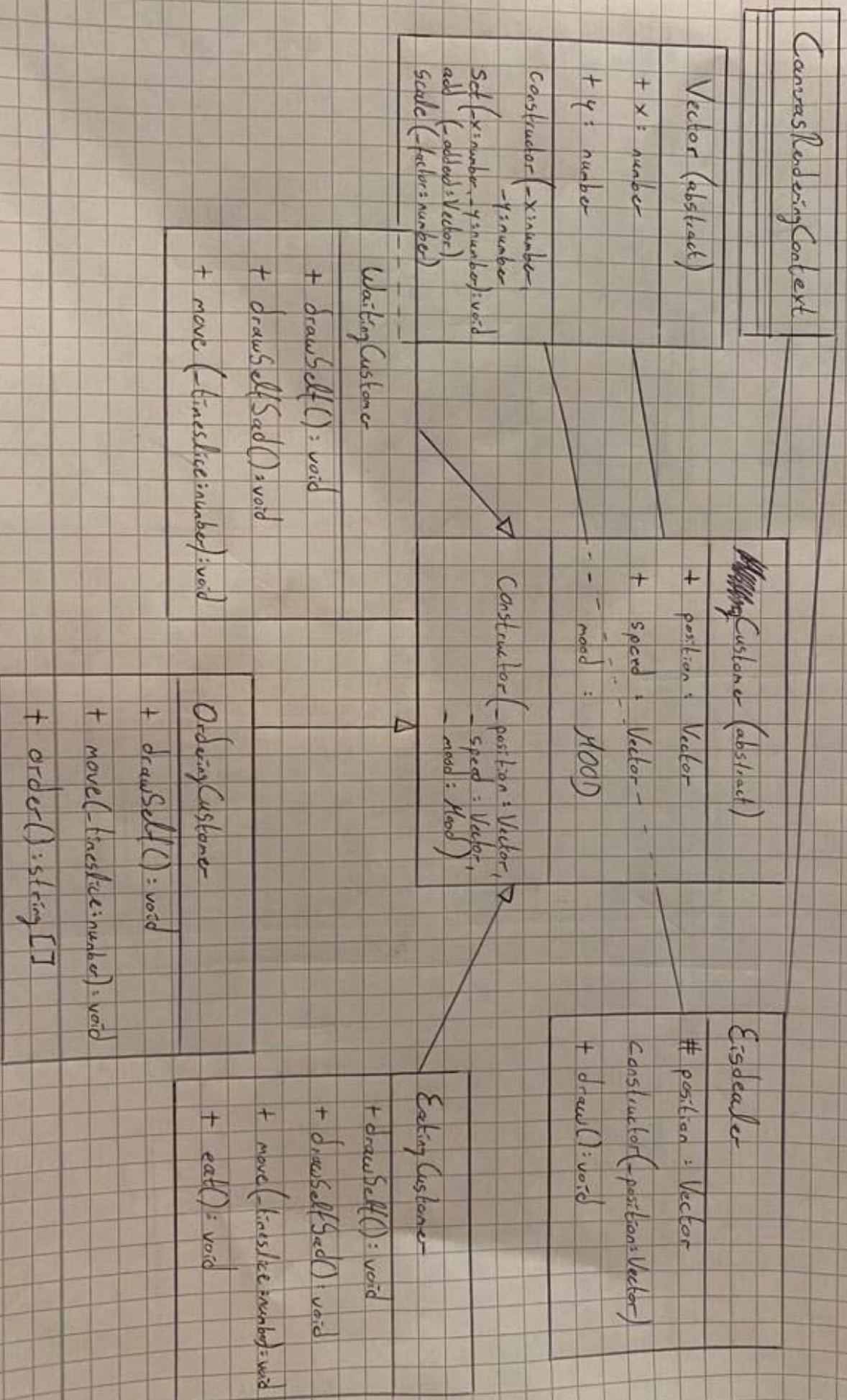
Unklarheiten :

Eis-dealer : Class-Diagram (alt)

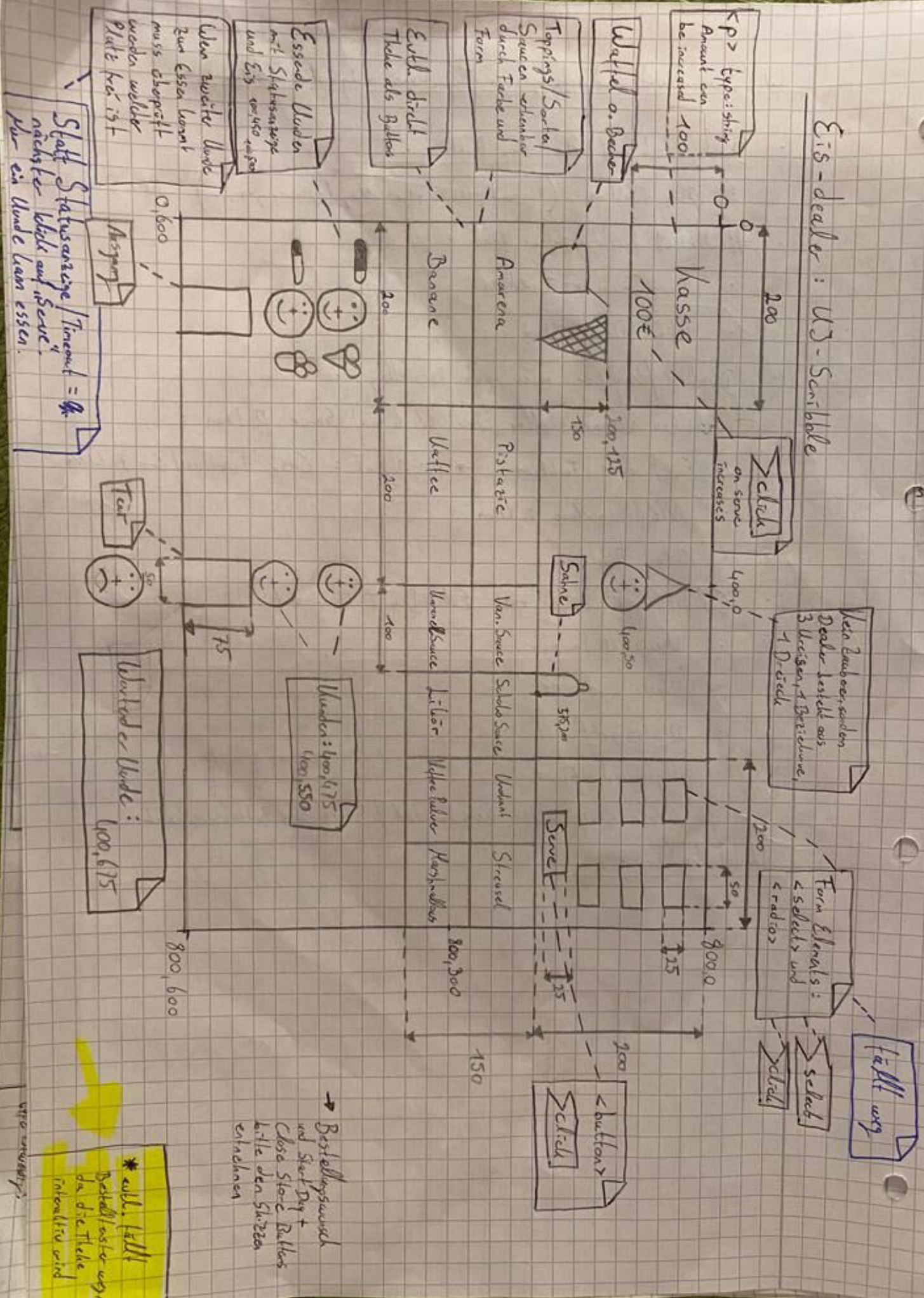


gibt unterschiedliche
Customer-Verhalten
Darstellung
Typen an?

Eis-dealer : Class + Diagram (aktuell)



Er is dealer: WS-Scribble

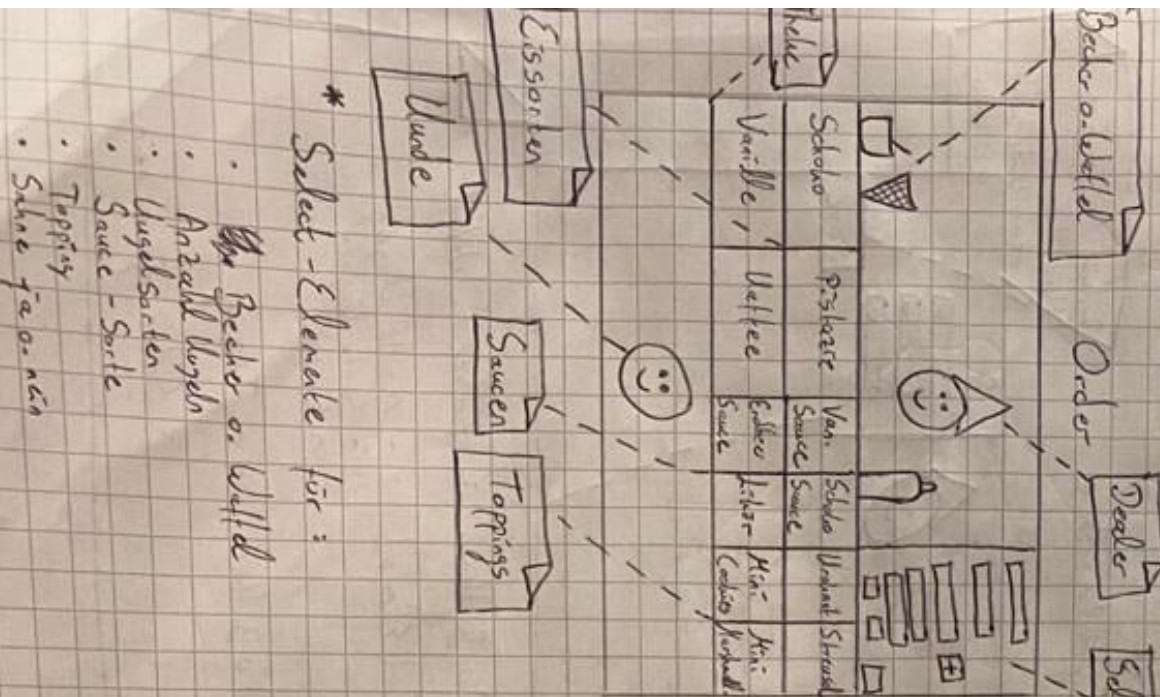


→ Bestellungsverwech
und Short Day +
Close Store Rollas
bilden den Switzer
entnehmen

- * evtl. fällt Bestellfrist weg, da die Teile internativ wird

Eis-Dealer: Skizzen

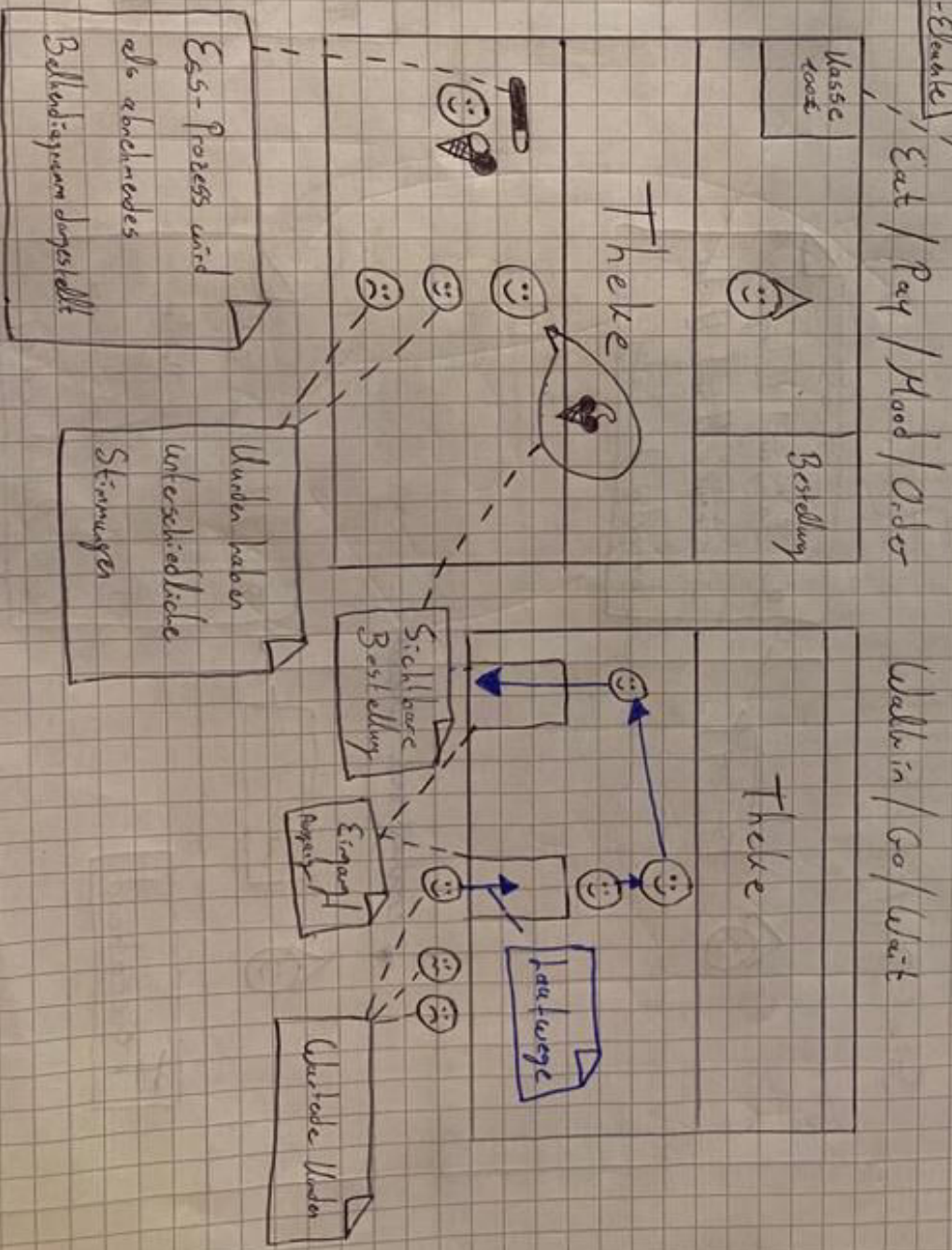
Kunden - Aktivitäten:



* Select-Elemente für:

- Bei Becher o. Waffel
- Anzahl Mengen
- Luge Sorten
- Sauce - Sorte
- Topping
- Sahne ja o. nein

Wasserkopf wird durchgehend angelegt, erhöht sich, wenn Kunde fertig gegessen hat

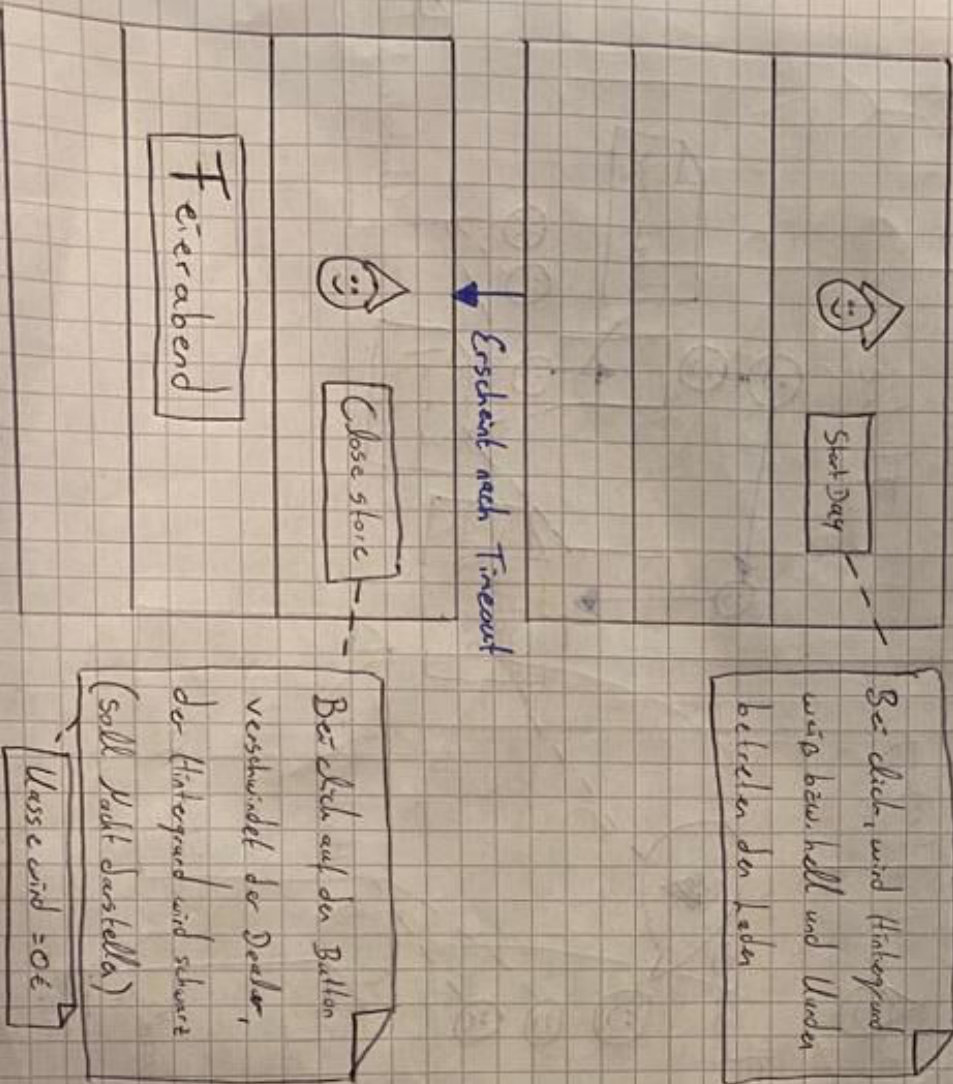


over 1/5 word

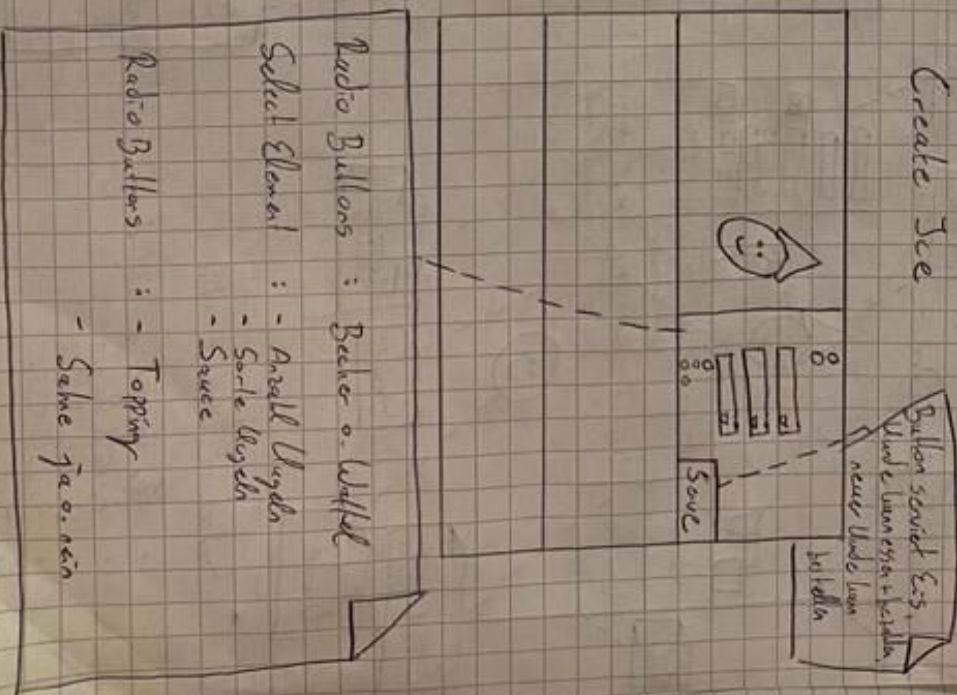
unwissen type network?

Dealer - Aktivitäten

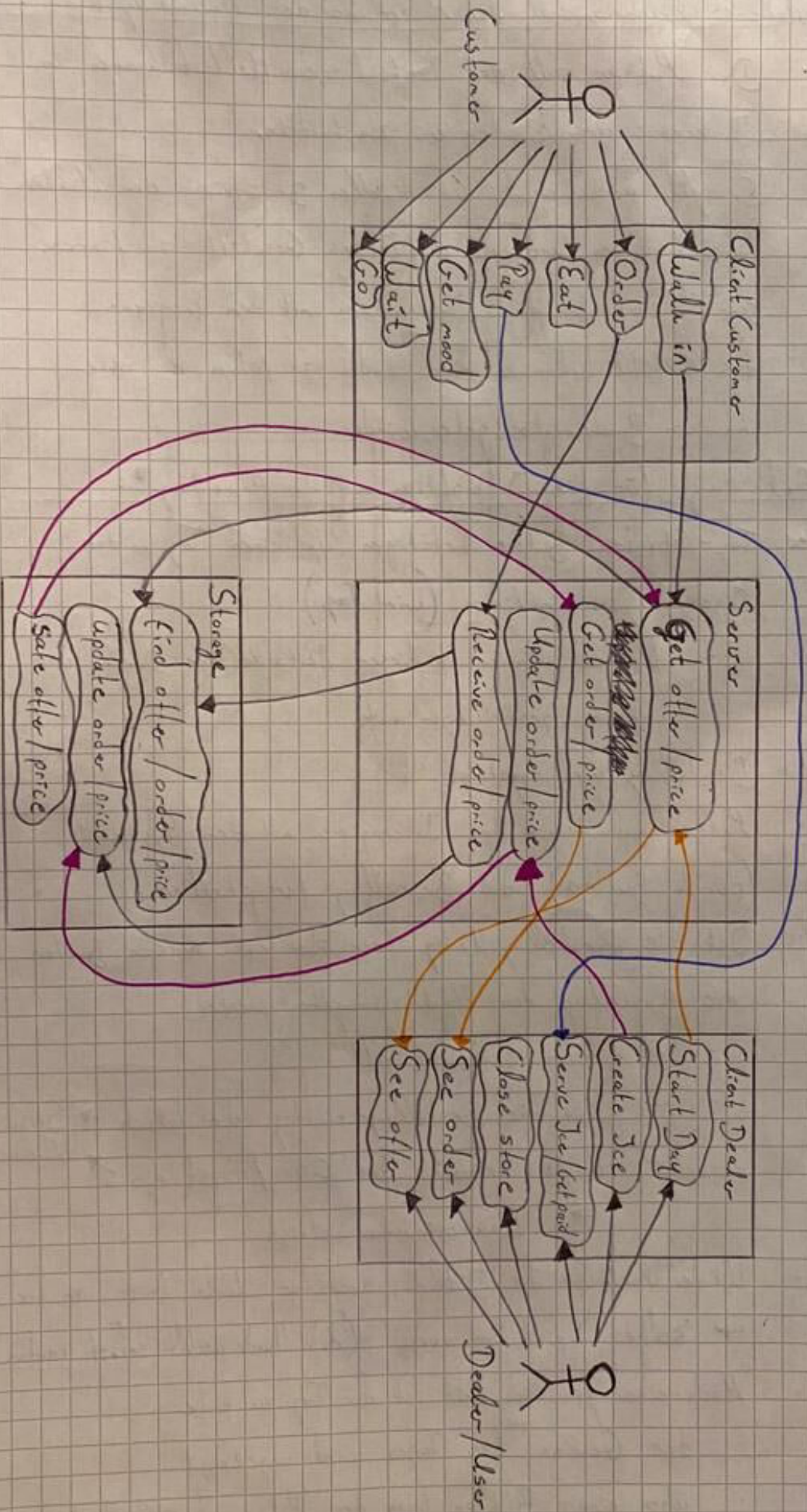
Start Day / Close Store



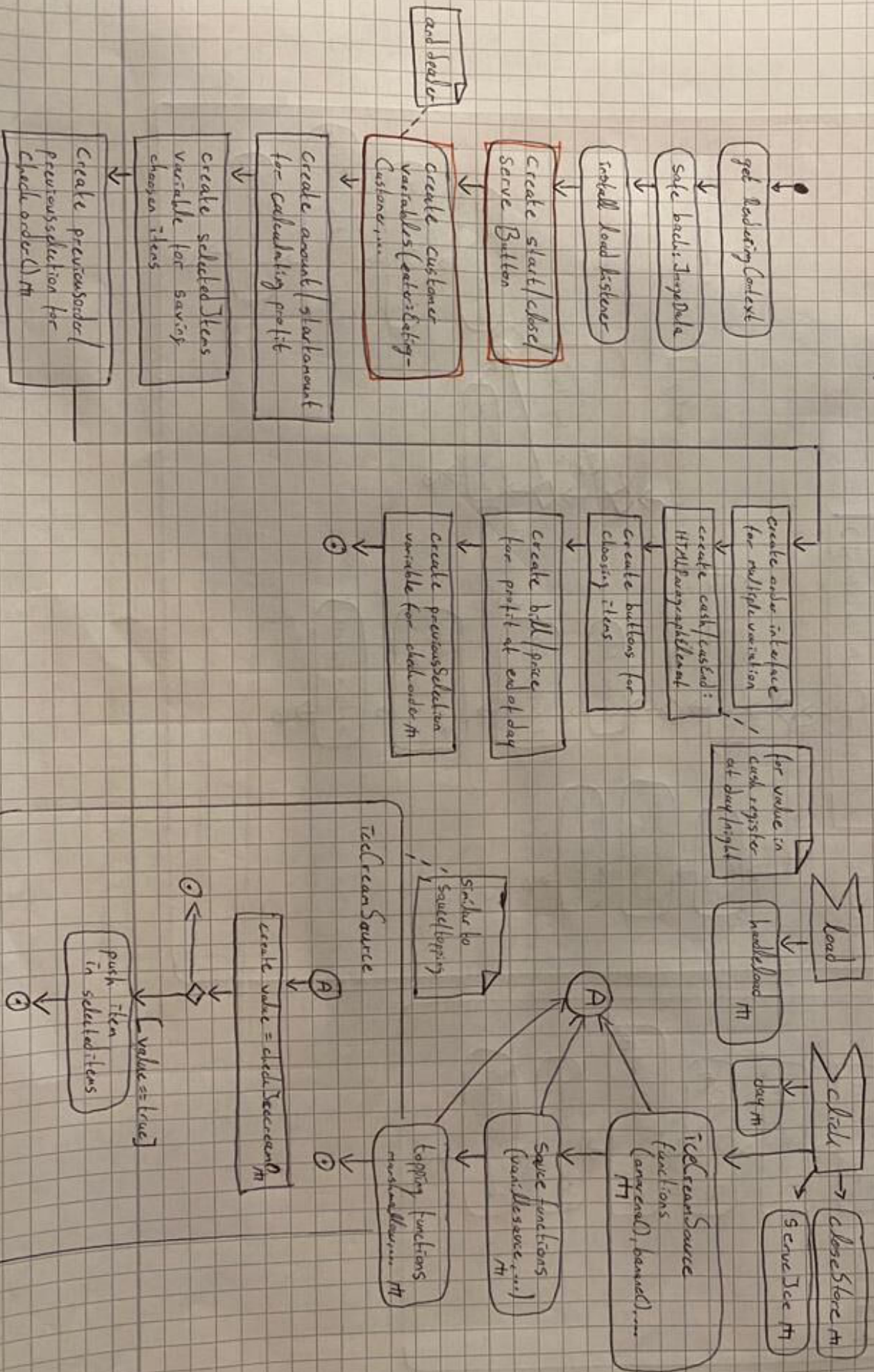
Create Ice

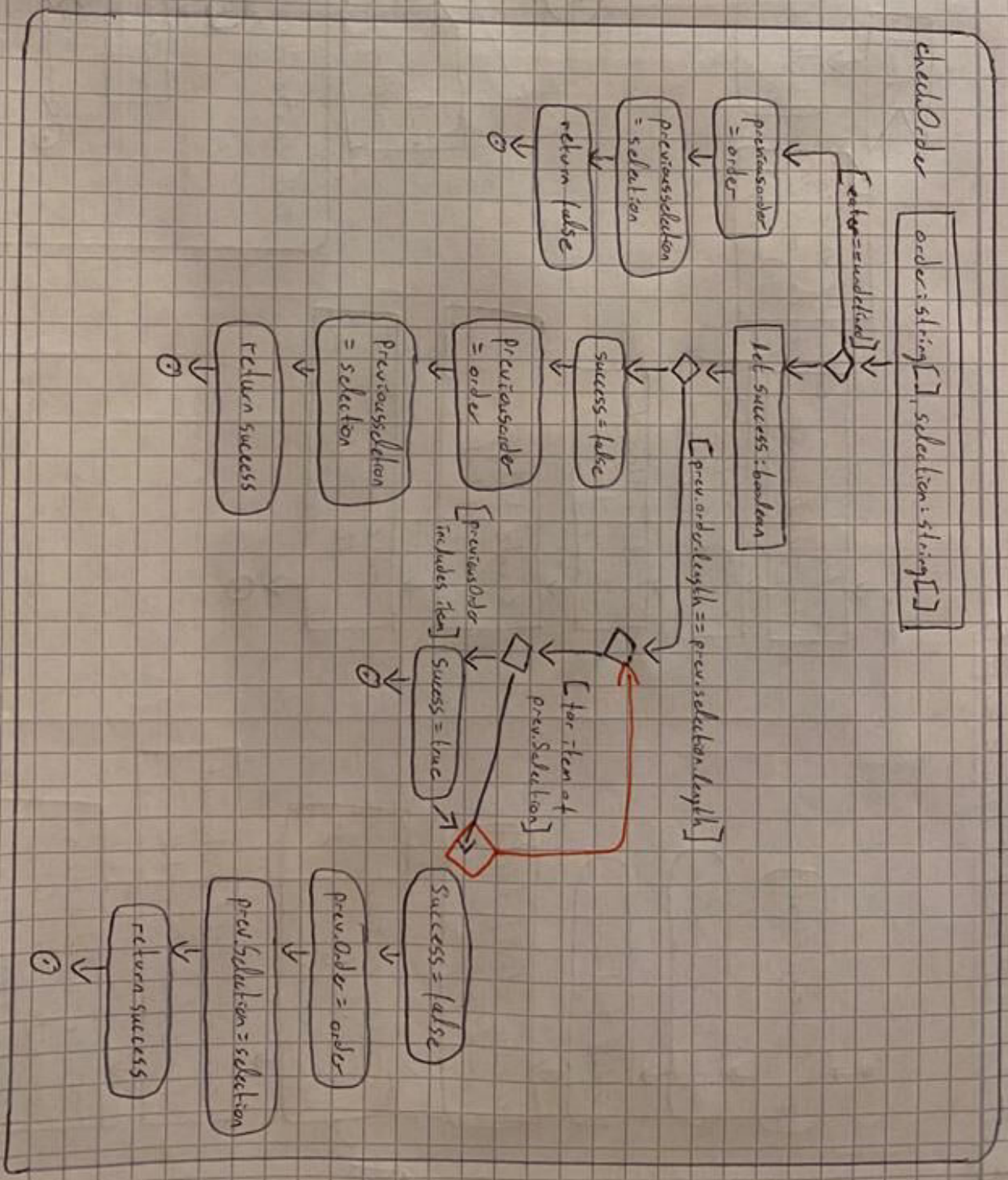
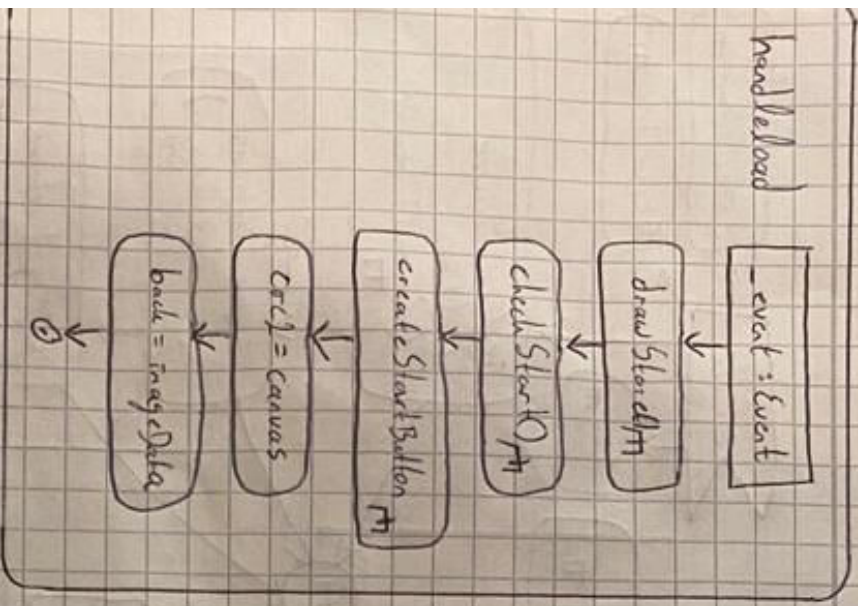


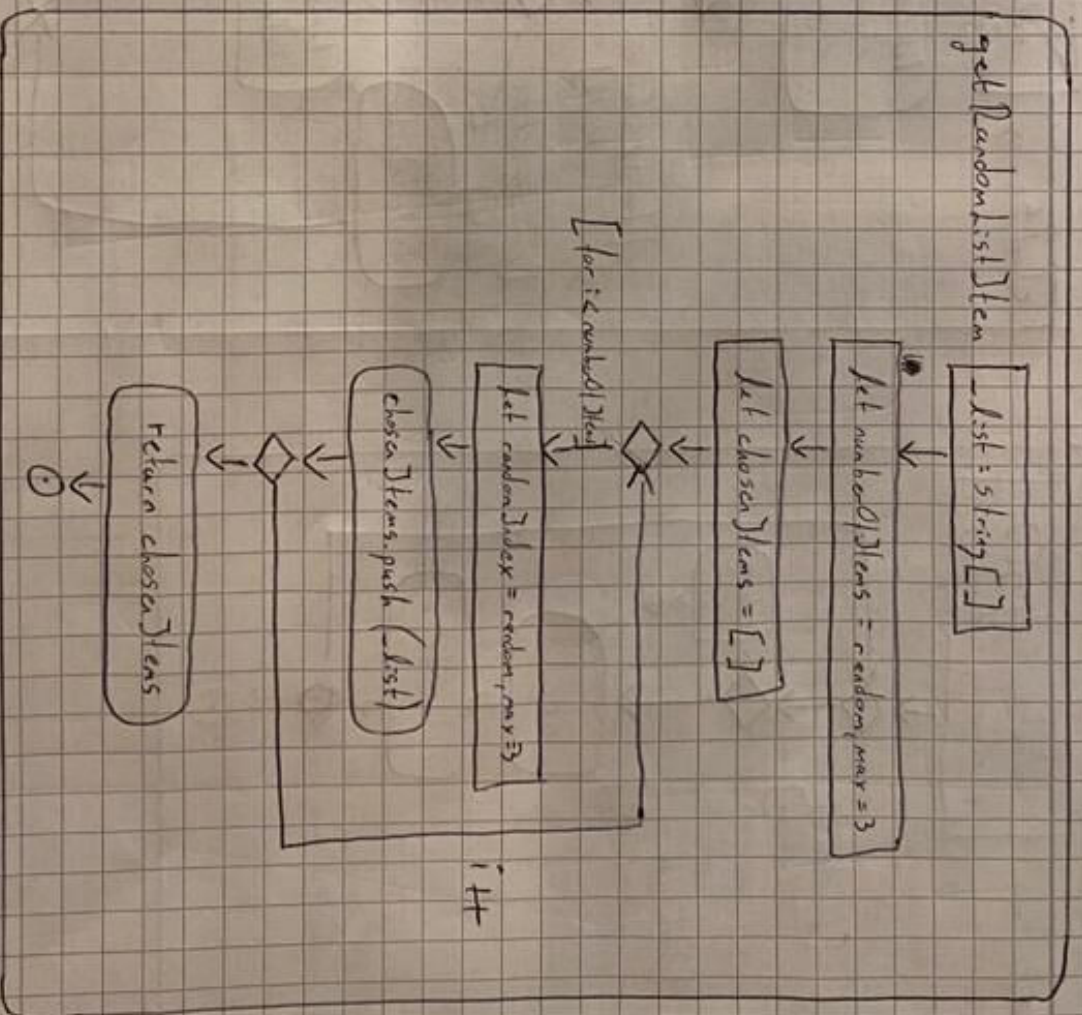
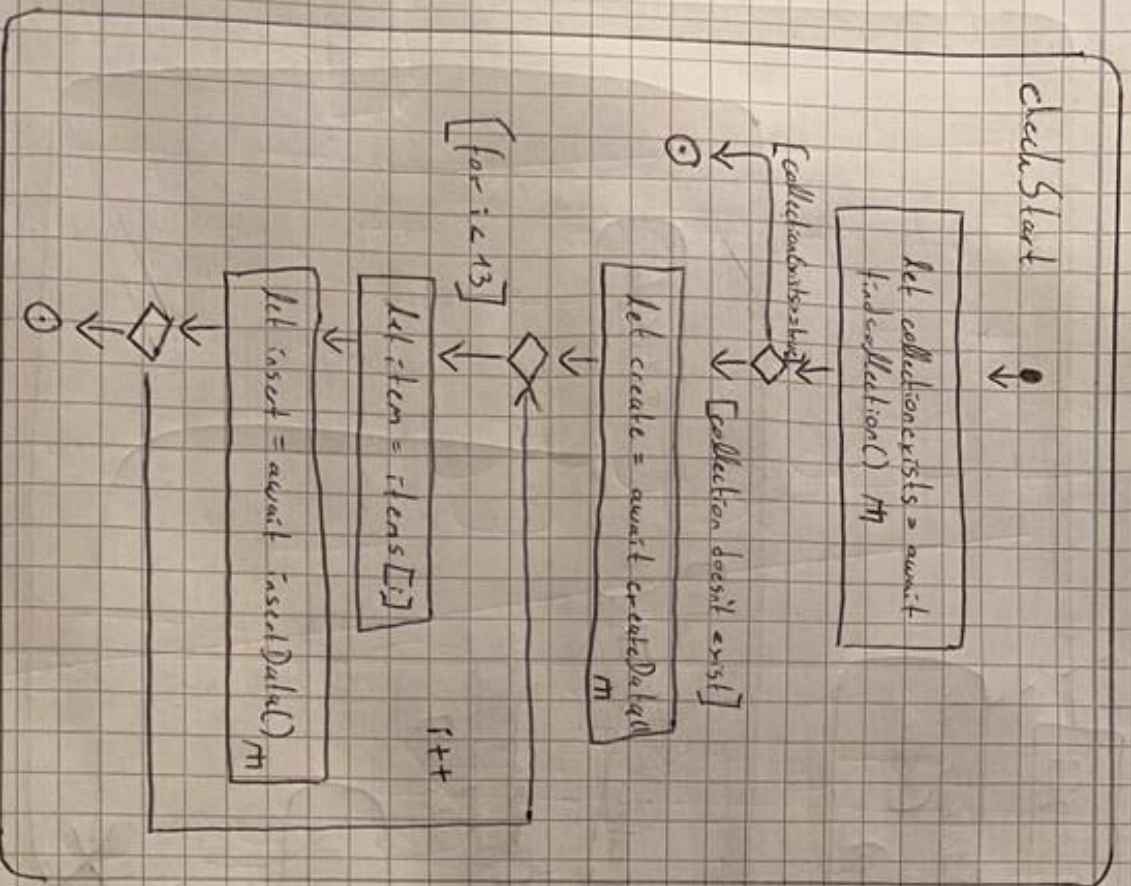
Eis-Dealer : Use-case Diagram / Anwerdungsfall diagramm



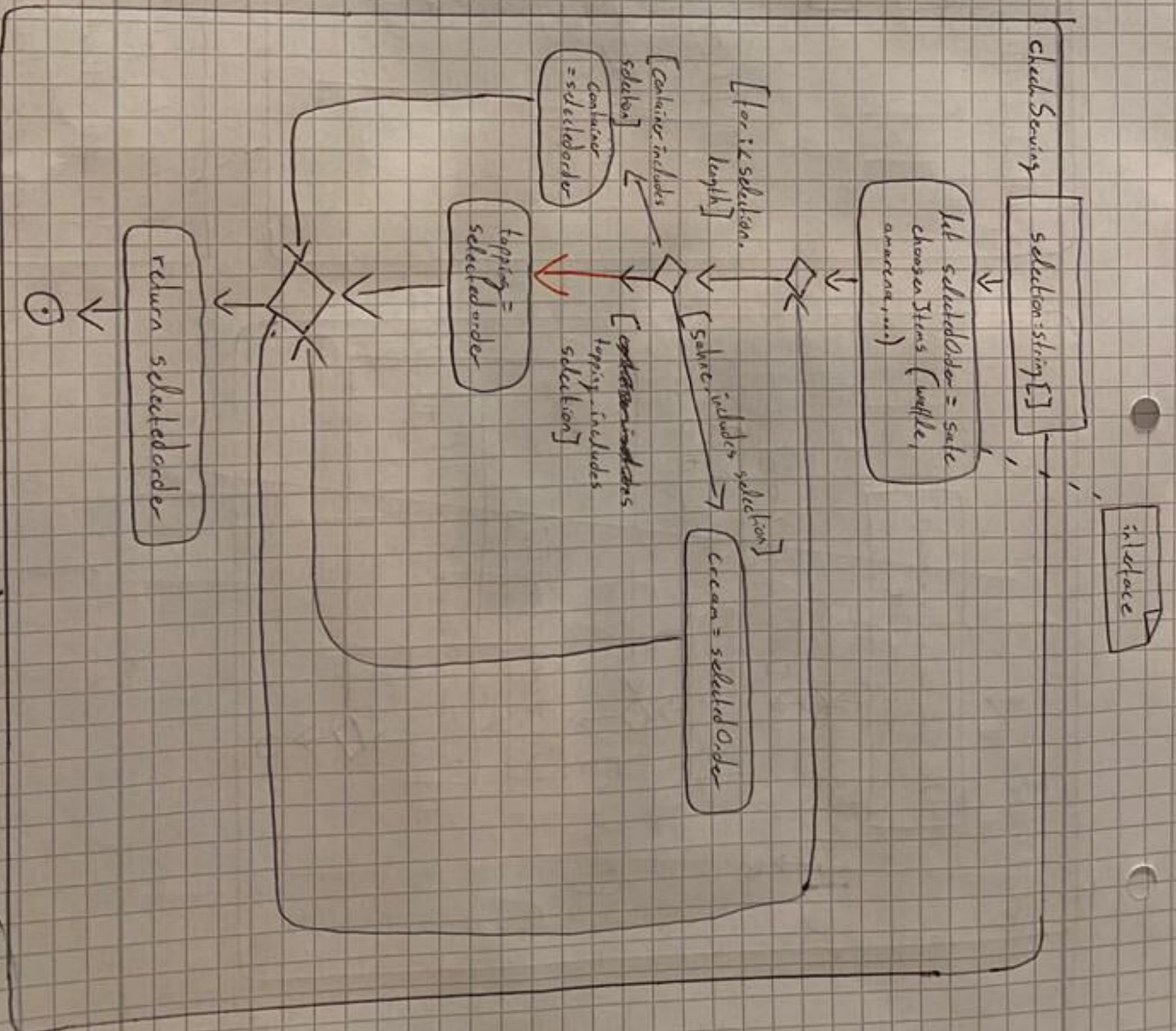
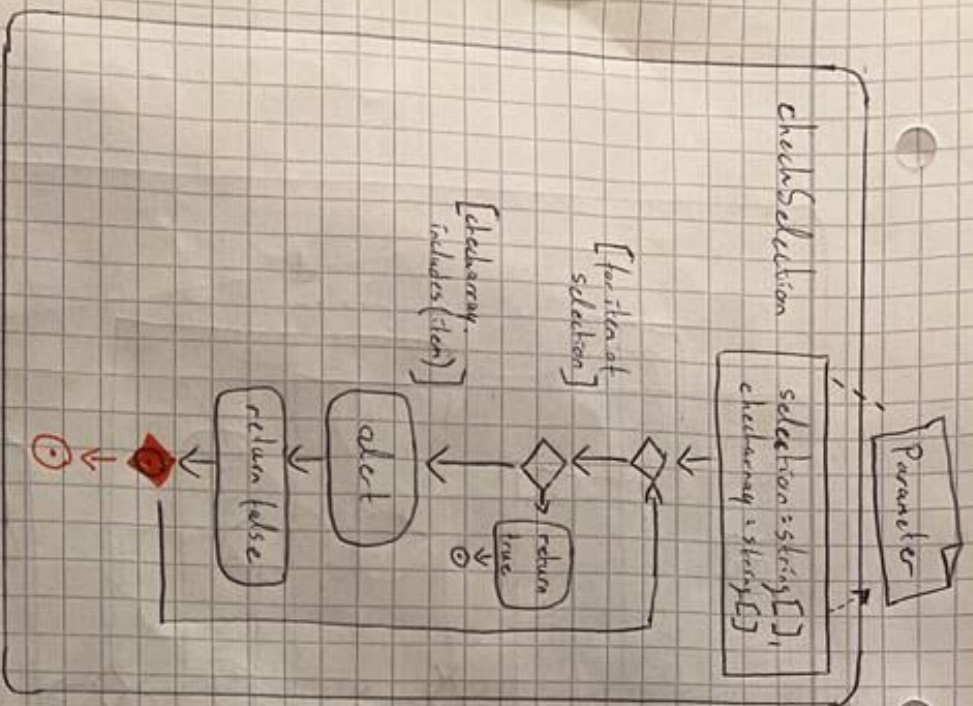
Endeasler : Activity Diagram scripts

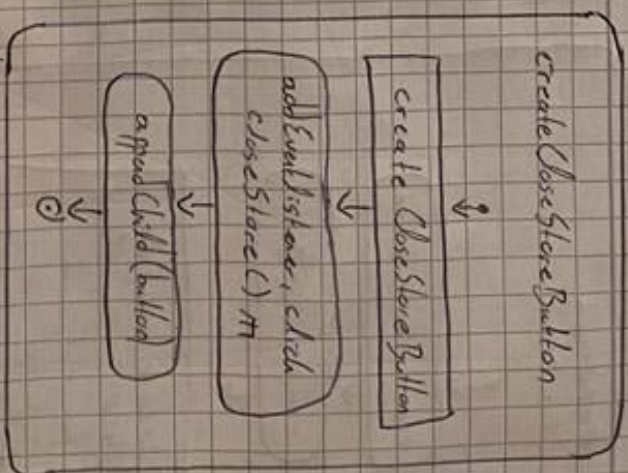
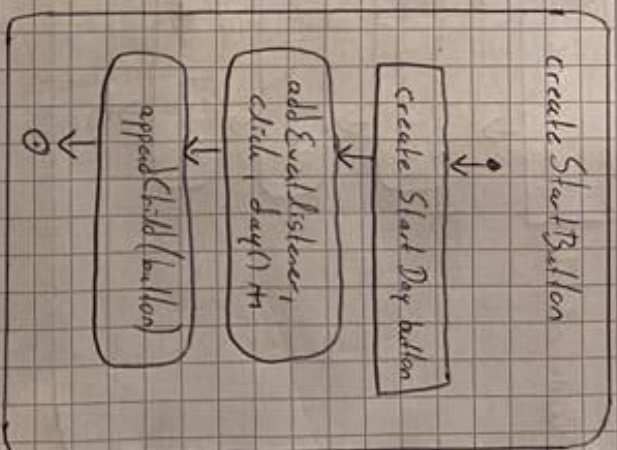
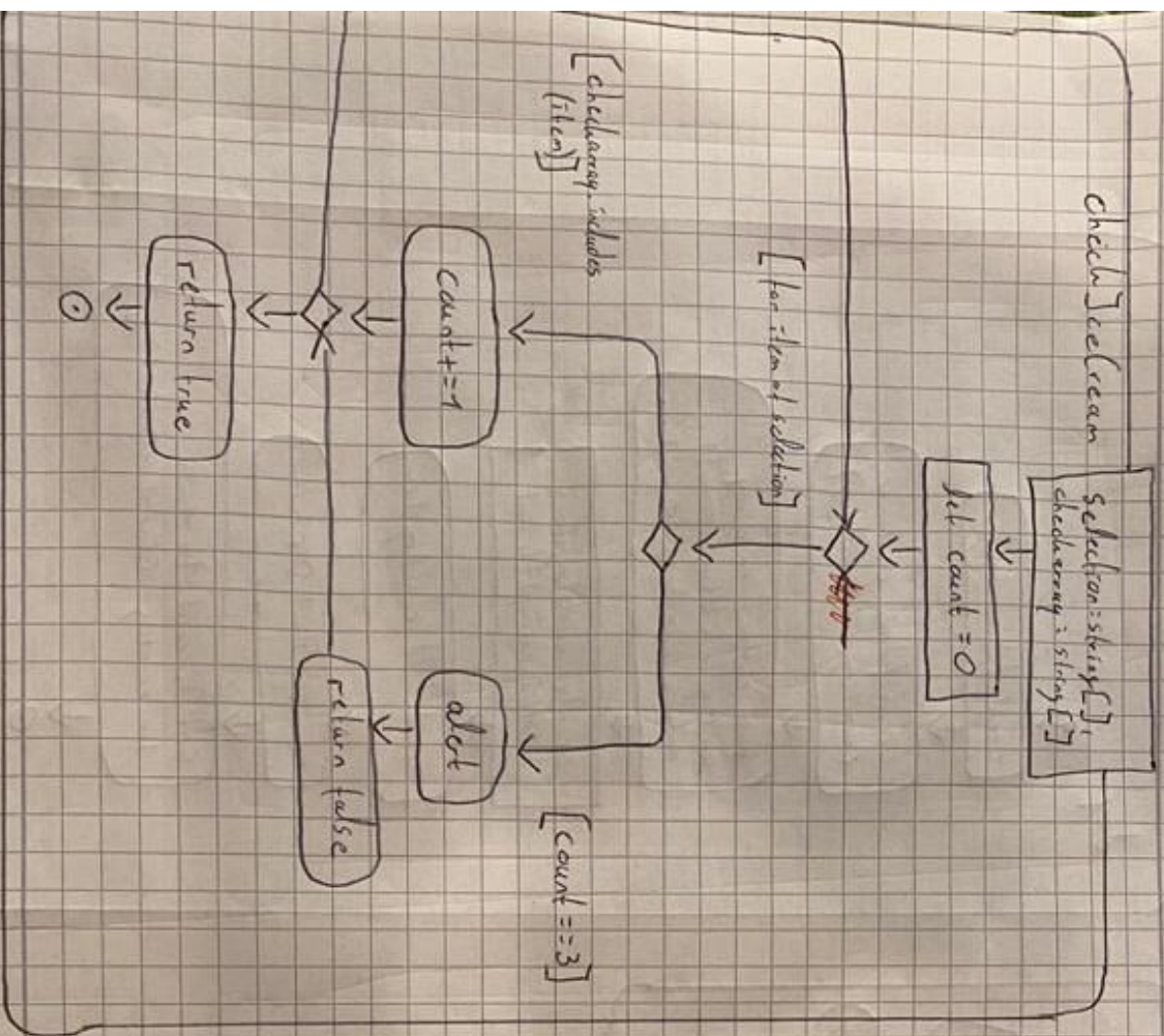






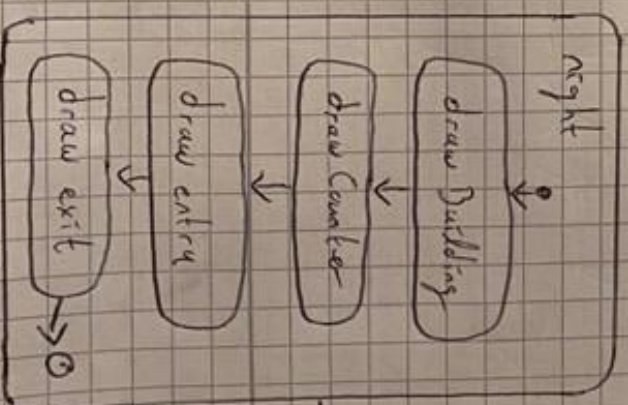
Similar to get1RandomItem
but ~~that~~ without loop, because
only one item can be chosen

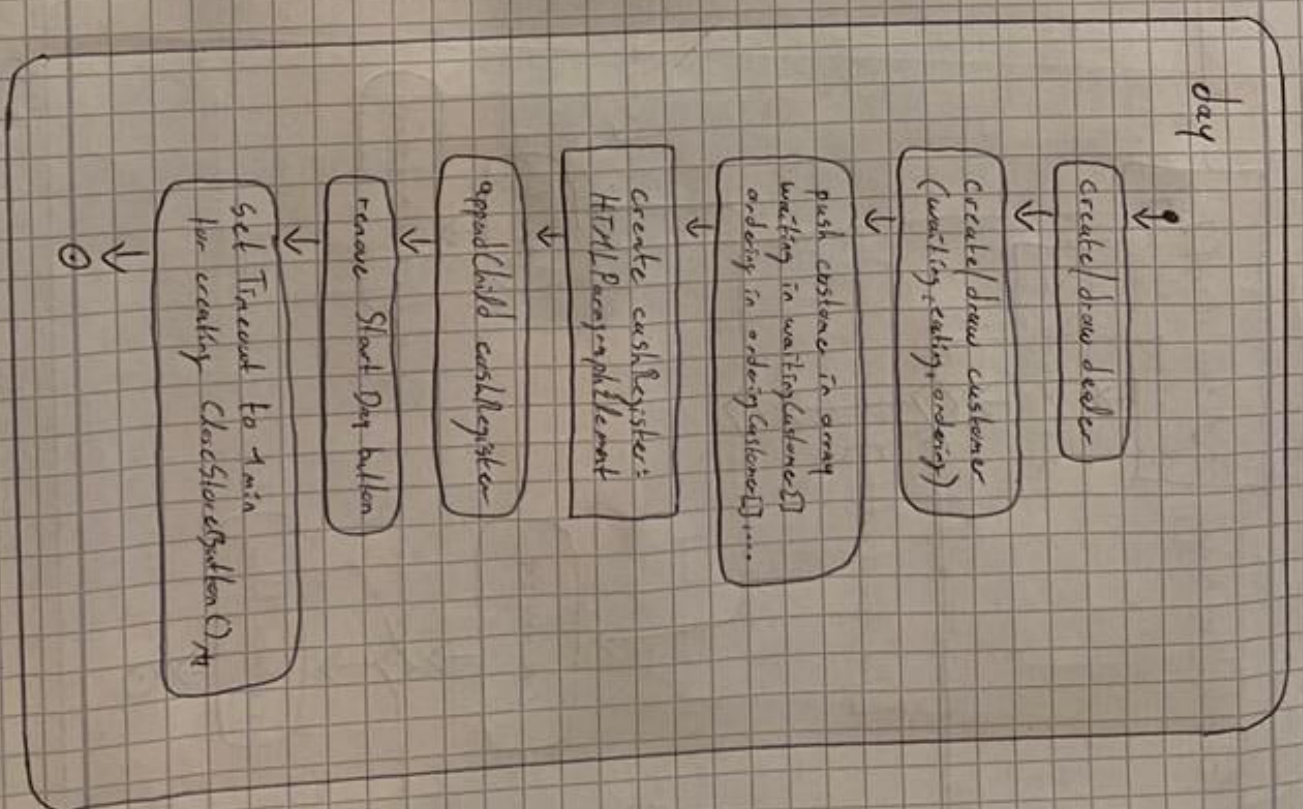
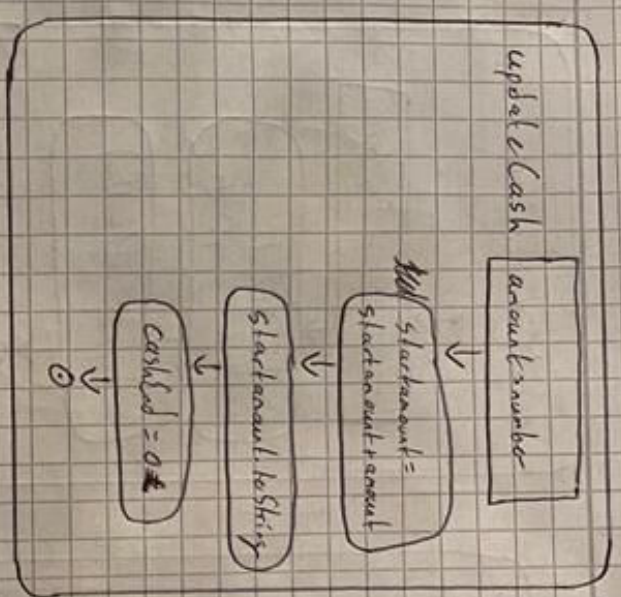
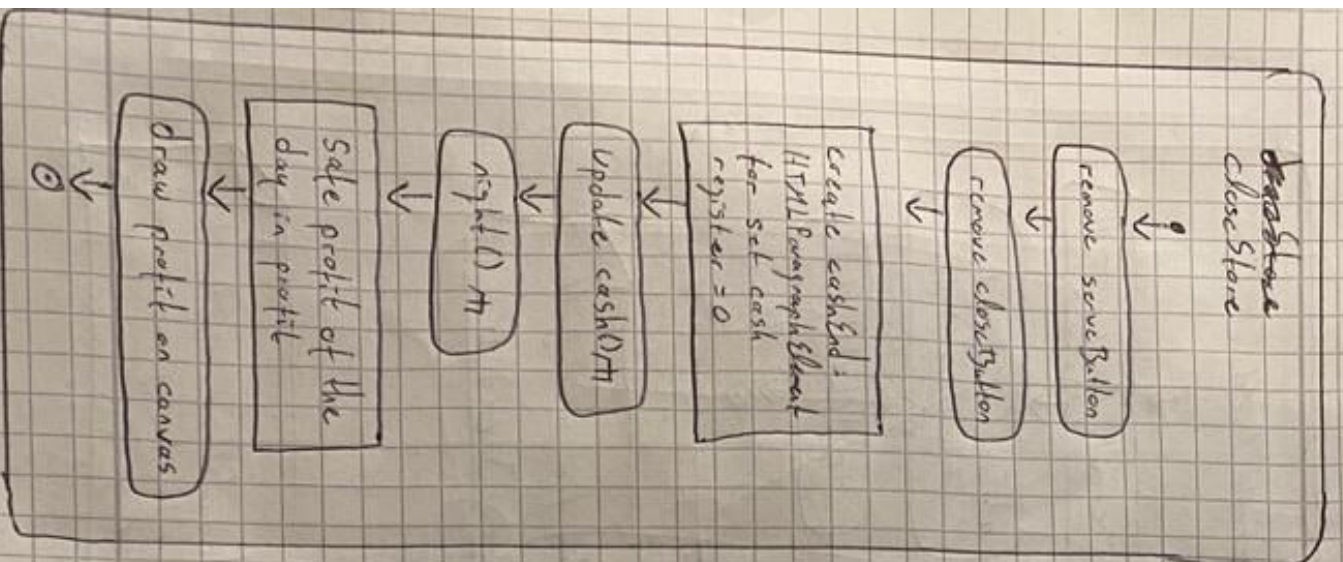




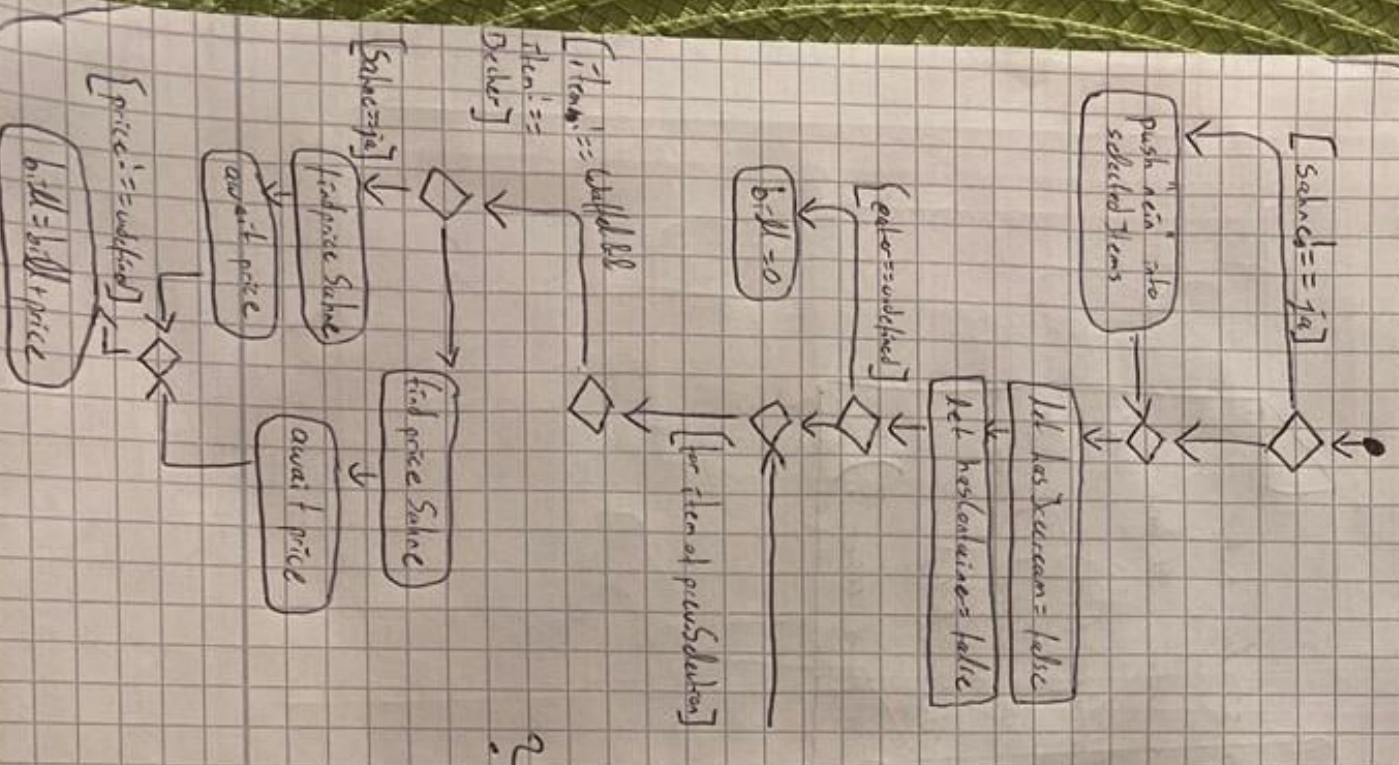
Similar for `serveJcecream`
`serveJcecream()` in

drawStore similar but
white background instead
of grey

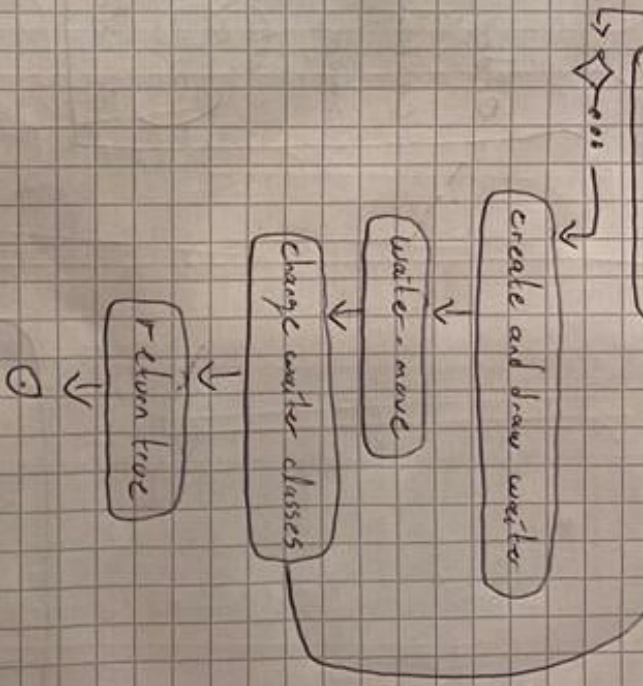




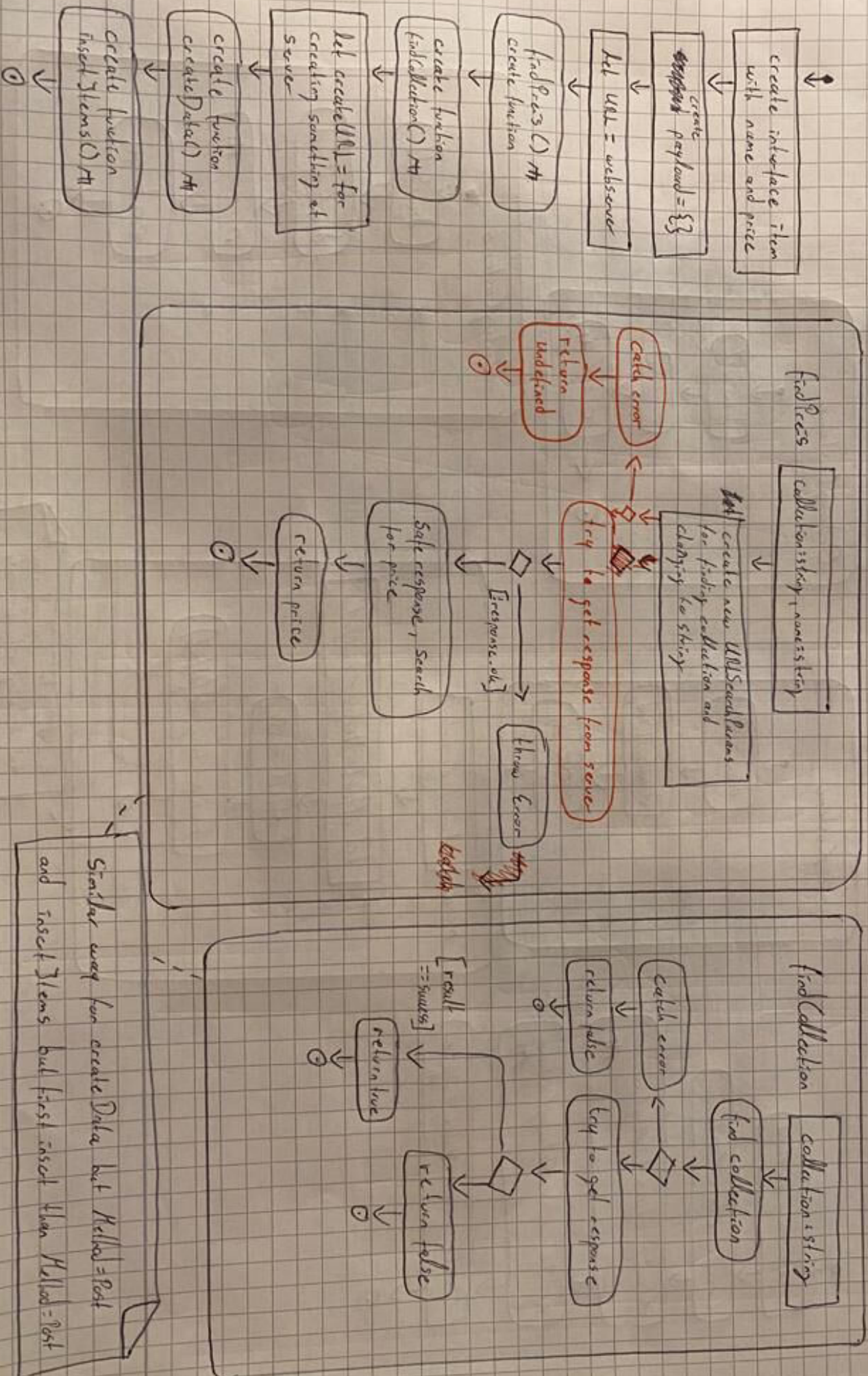
Seve Bee



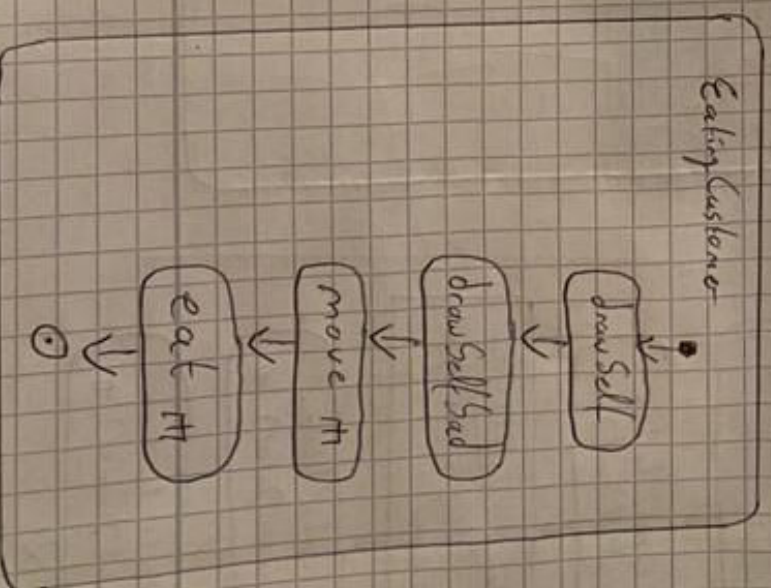
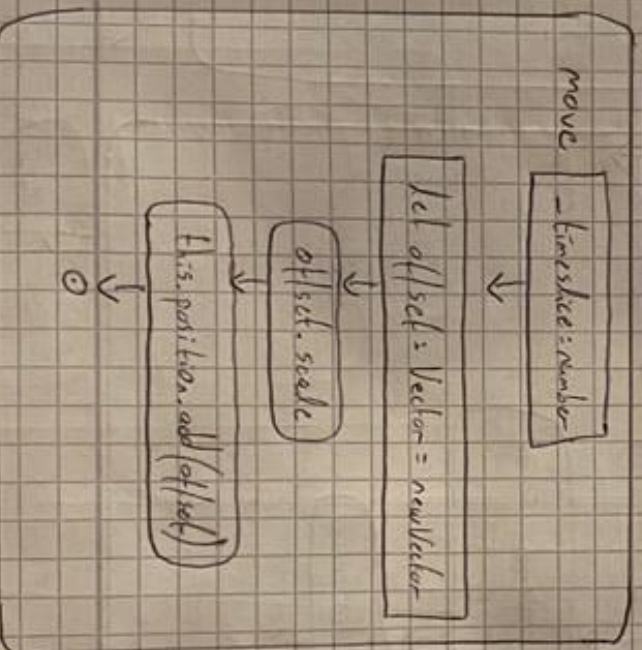
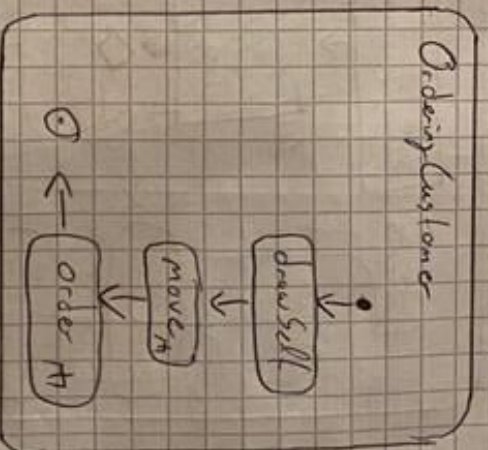
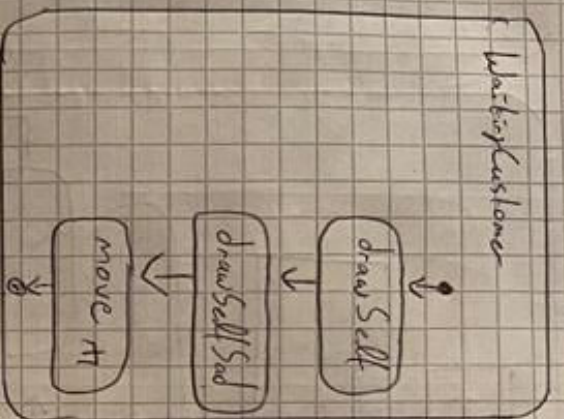
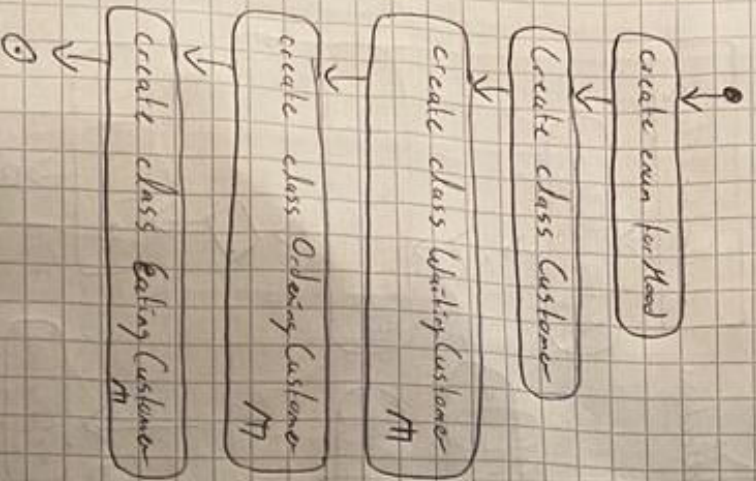
Set Interval



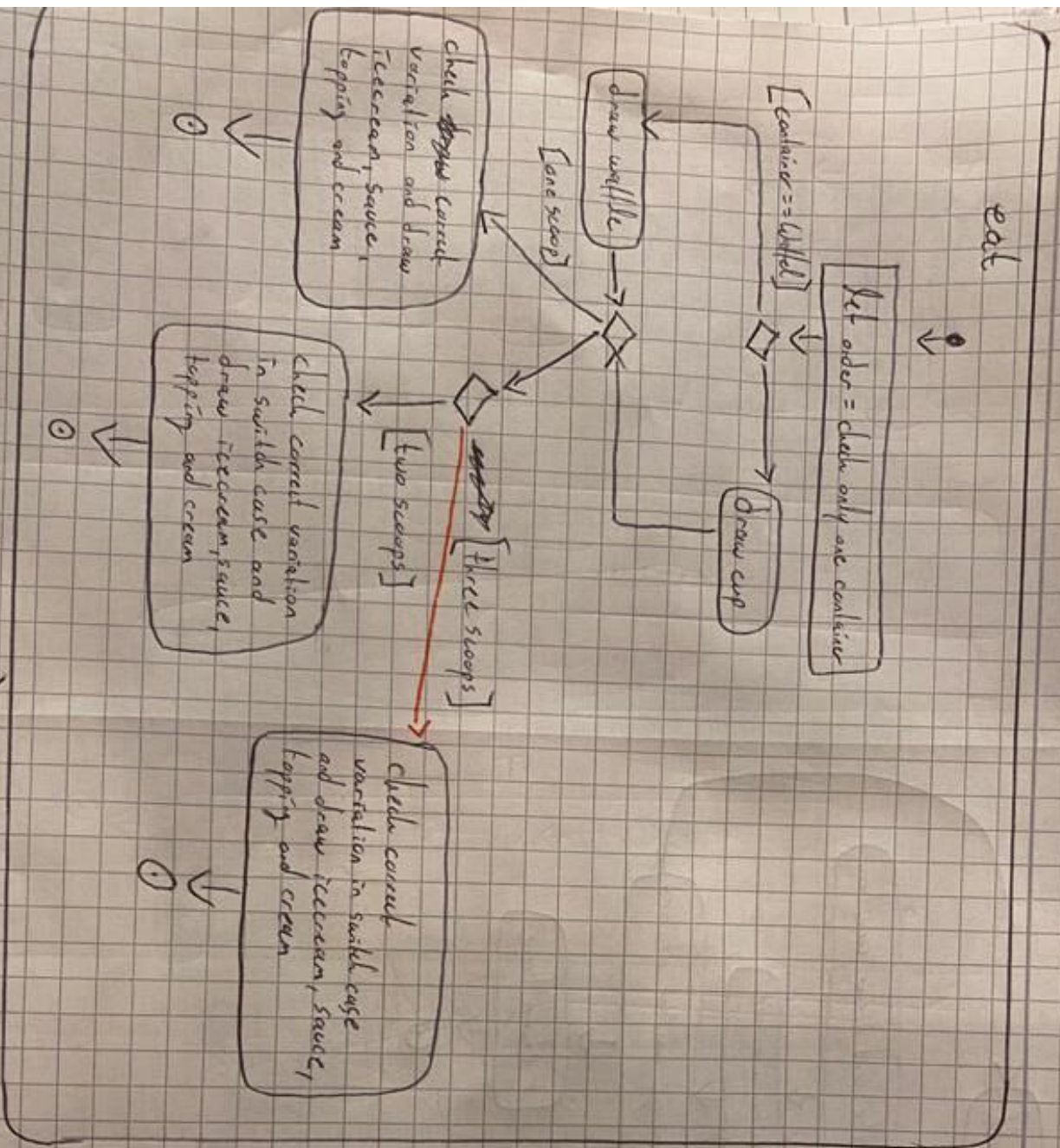
UML Activity Diagram: server side



Activity Diagram: Moving Customers



Similar to eatingCustomer and orderingCustomer only other call options



Order is similar but customers order will be saved in arrays for checking with selected items (is order identical to user selection from user?)