

Discussion on Diagrams

27th Sept

Employee $\left\{ \begin{array}{l} \text{staff} \\ \text{cook} \end{array} \right.$

~~Can~~ Customer $\left\{ \begin{array}{l} \text{Comet} \\ \text{Non-Comet} \end{array} \right.$

→ eg has, has

- relationships should be meaningful.
- use terminology of food dept eg. for food joints
- relationships + attributes
- where is cell phone? → hardware phone, hardware kiosk
- where is ticket coming from?
- "System" should not be in Domain Model

→ Add admin in VCD

→ Payment Authorization System is also an actor

→ U.Cases should be at same level of abstraction.

→ ★ Place Order is main purpose of app.

Level 0

Customer → Place order

staff → process order

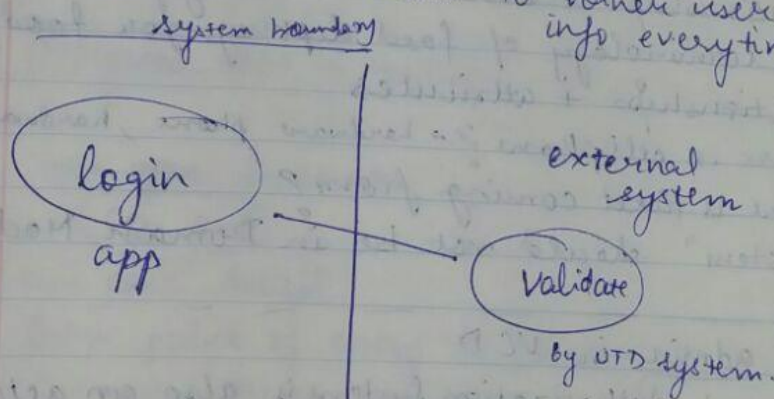
admin → maintain

chef → prepare order.

has "INCLUDES" ← Level 1 (highest level of abstraction) → main system capabilities

Customer - Placing order < app
~~Customer~~ taking ticket

login? → create account/register/login needed in app if we don't want to bother user with info everytime.



no login means app is downloaded & used (no validation).

staff — Process order — Cancel order
process order

chef — Prepares food — Receive info on screen
indicate when order ready

IMP part of detailed use cases

- display ^{wait} time
- display service number
- triggered by time/system & returned to user.

both can be use cases \therefore they are initiated by system & can be a use case.

* time is an actor \therefore it pushes system to interact with user.

brief
casual
fully-dressed

} use-case description

→ Priorization of Use Cases : Critical for user & critical from perspective of software developer

a) Place Order → fully dressed (don't put everything in one template)

↳ clean & organized

textual description: main scenario (step by step)
↳ happy day scenario

- you can have "if, when" ^{etc} still be in main flow
- for exp. exceptions → you don't maintain in same flow

Black box SSD should be a direct translation of your happy-day scenario i.e. textual description

ref login

→ means you have a reference black box for login

→ prioritization of exceptions.

ticket → e-ticket

payment system

admin

employee → staff

Imp

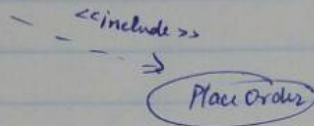
Place order includes make payment

include → when you are re-using

↳ when you have more than one use-case.

if you have exceptions during make payment

Browse food joins



SSD → inconsistencies

↳ actors not same for SSD & UCD.

- X sending order before payment
- loop needed for more orders.
- exceptions (simple)
- different SSD for smaller parts.
- employee packing food based on what?
 - chef indicates staff what order is ready.

→ incomplete SSD.

→ each & every detailed step in SSD.

Friday

- ~~that~~ → UCD (all 2-3 levels),
- UC Prioritization
- Place order → fully dressed (+ includes & extends)
- ↳ happy day scenario → SSD.