Agenda :-

1. class diagram
2. schema
3. How to code
4. Tic Tac Toe Requirements.

MVC

Any physical entity that can write behaviour is a pen.

Supporting gel pen, ball pen, fountain pen.

Gel and ball pen will have a refill, others will not have a refill, and refill will have ink.

Different inks have different colours.

For every pen, need to store the brand, name and price.

Pen can be either closed via cap or via click buttons.

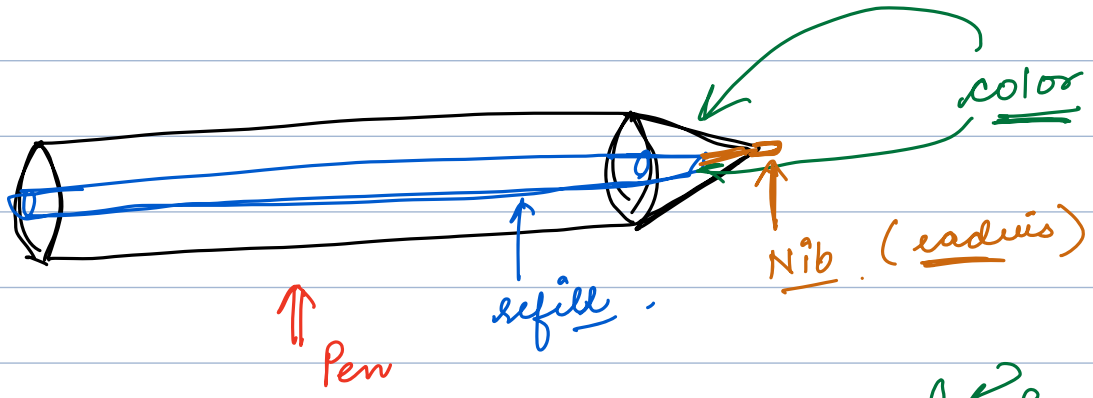
Often refills have different nibs radius, do we have to store the radius of nibs?

The fountain pen itself has nibs.

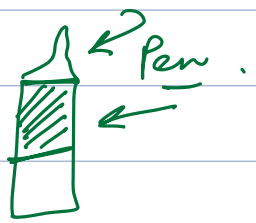
Pens should be able to write.
We can close/open the pen.
We should be able to refill pens.

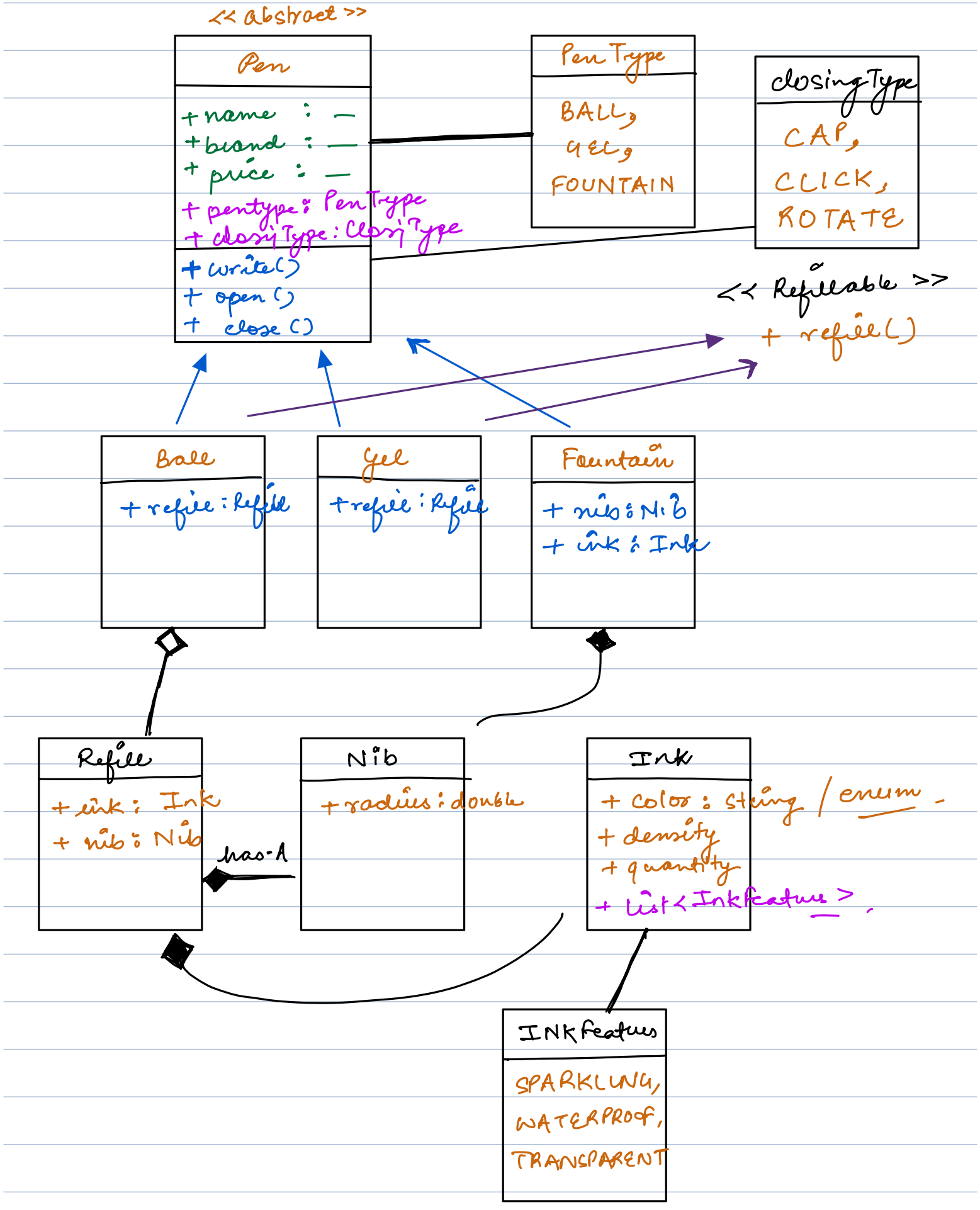
class diagram

1. Visualization
2. using nouns in the requirements.



Ball, Gel, Fountain




<< abstract >>
Pen
+ name : —
+ brand : —
+ price : —
+ pentype : PenType
+ closiType : ClosiType
+ write()
+ open()
+ close()
PenType
BALL,
GEL,
FOUNTAIN
ClosingType
CAP,
CLICK,
ROTATE
<< Refillable >>
+ refill()
Ball
+ refile : Refill
Gel
+ refile : Refill
Fountain
+ nib : Nib
+ ink : Ink
Refill
+ ink : Ink
+ nib : Nib
has-A
Nib
+ radius : double
Ink
+ color : string / enum
+ density
+ quantity
+ List<InkFeatures>
INKfeatures
SPARKLING,
WATERPROOF,
TRANSPARENT

How to approach Schema design ?

① For each class that represents entity, create a table

```
class Movie {
    String name;
    int releaseYear;
    List<Actor> actors;
}
class Actor {
    ==
}
```

M : M
M : 1

Movies

id	name	releaseY

Actors

id	name

movie_actors

movie_id	actor_id

② each primitive attributes of these classes will go as a col^m in table

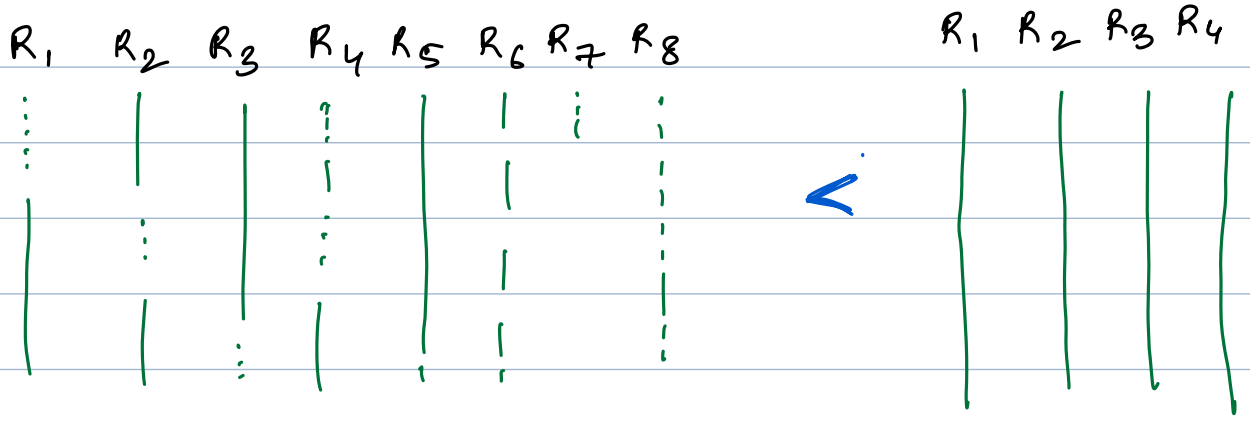
③ non-primitive attributes.

↓

cardinality

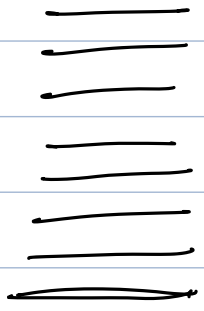
How to code ? 8-10 req (1-1½ hr) ?

① Atleast some of the req must be working end to end.



② code structure

src/



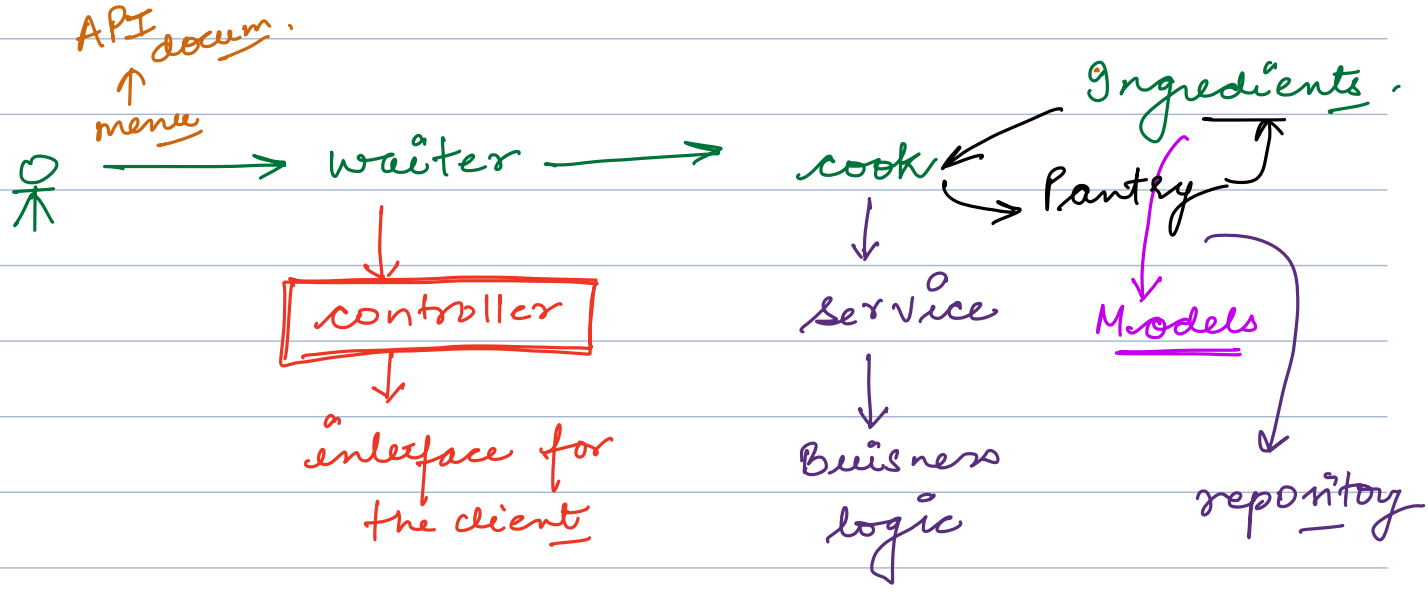
MVC architecture

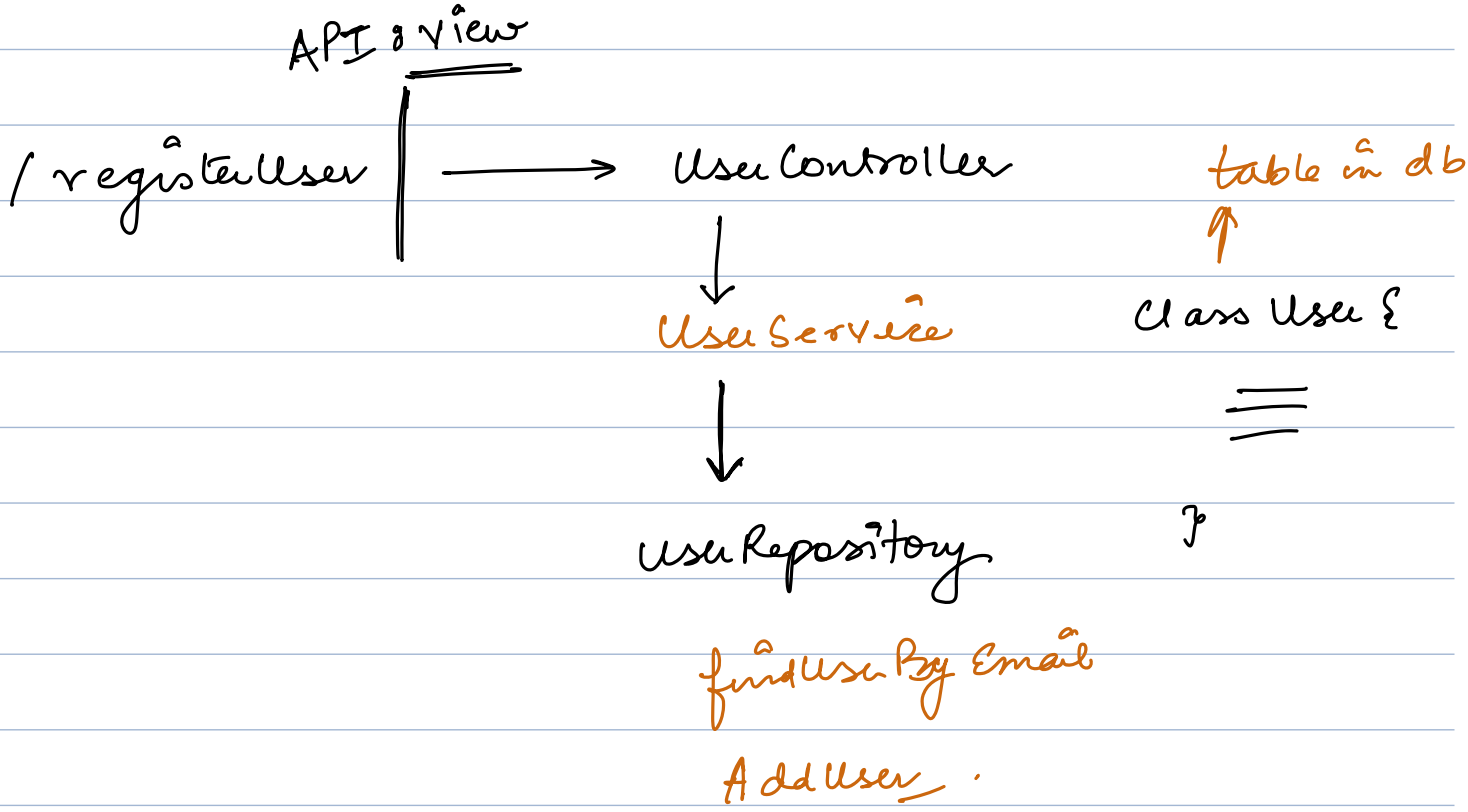
M : models

V : View

C : controller

MVC architecture





Tic Tac Toe

Step 0 : Overview

Know.

Not aware.

Software system ✓

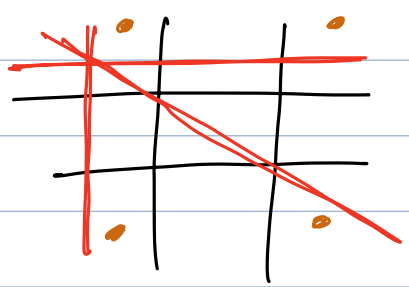
Persist X

input Hardcode + command.

Break: 10:38 pm

Req gathering / clarification

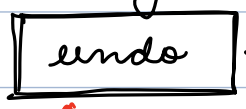
1. size of board : n * n
2. no of players : n - 1
3. Each player can chose their own symbol
   - validation ← { 2 players chosing the same symbol }
4. Can I have bots ? **Yes**
5. can my bot have a difficulty level ?
   Easy, Medium, Hard
6. Who will make the first move ?
   → random.
7. Players can't leave the game in between.
8. How the winner is decided ?
   - flexibility
   1. vertical
   2. hz
   3. diagonal
   4. corner



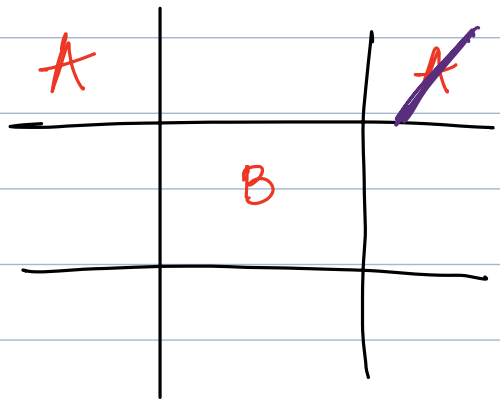
(9) Game → draw

(10) Undo feature

global.



Anyone can press it at any moment



undo
↑
B.

---

db.

- Leaderboard
- Tournament
- Timer — Frontend
- Autoplay.
- Pause / Resume / Reset
- Exit
- Some cells are blocked.
- Replay

- undo
- class diagram
- check winner

HW

↓

$n^3 \rightarrow n^2 \rightarrow n \rightarrow O(1)$

LLD 2 Mock

↓

LLD 1

+ missed the contest → attempt the reattempt.

+ 2-3 problems.

↓

- chess
- Snake & Board
- Tic Tac Toe