

Agenda

- 1) Schema design
- 2) Project Structure
- 3) Design Tic Tac Toe

How to approach schema design

Cardinalities

Rules

- { 1:1 \Rightarrow id of one side on other side
- 1:N] \Rightarrow Id of one side on Many side
- M:1]
- N:M \Rightarrow Mapping table

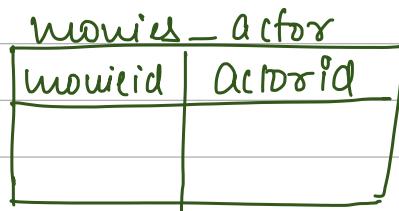
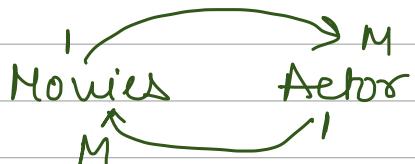
eg



inst



Batch



Steps to convert class design to schema design

- 1) For each class (Entity) in the class diagram, create a table
- 2) For each primitive (Simple \rightarrow primitive datatype + String) attribute of class, create a column in respective table

3) for Non primitive attributes, find the cardinality of relation & apply its rule

29

Class Movie {
 MovieId; }
 date :] Primitive
 Title] Primitive
 List<Actor> actors] Non
 Primitive

Class Actor {
Actor id;
Name;
dob;
} Actor

Mouiee

Actor id	Name	Dob

Movie — Deton
M: M

Movie_actor

Moweid	Detroid

(Collections) should be considered as Non-primitive
of Primitives

Inst of
id
Name;
ListBatch>

$$\begin{array}{r} 1 : M \\ \hline 1 : 1 \end{array}$$

Batch d
id:
Name:
Current:

Inst

id	name	

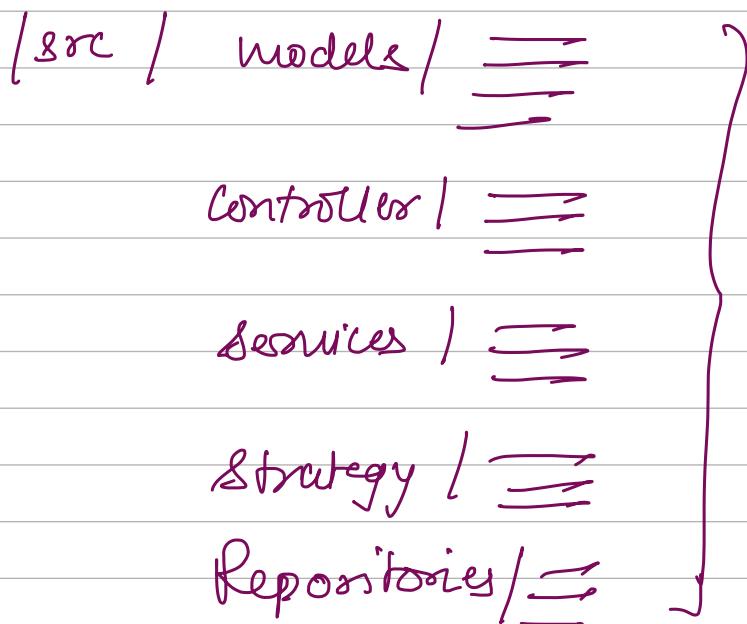
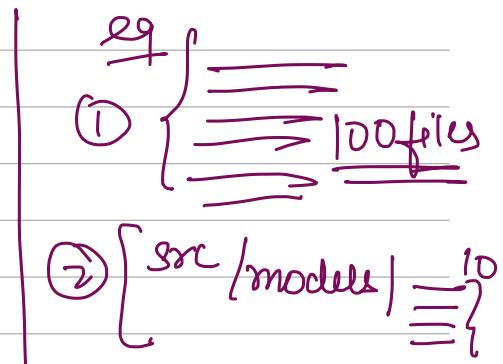
Batch

id	name	instid

How to structure code

1) Project structure

Our code base should have proper structure i.e. different folders for different type of files

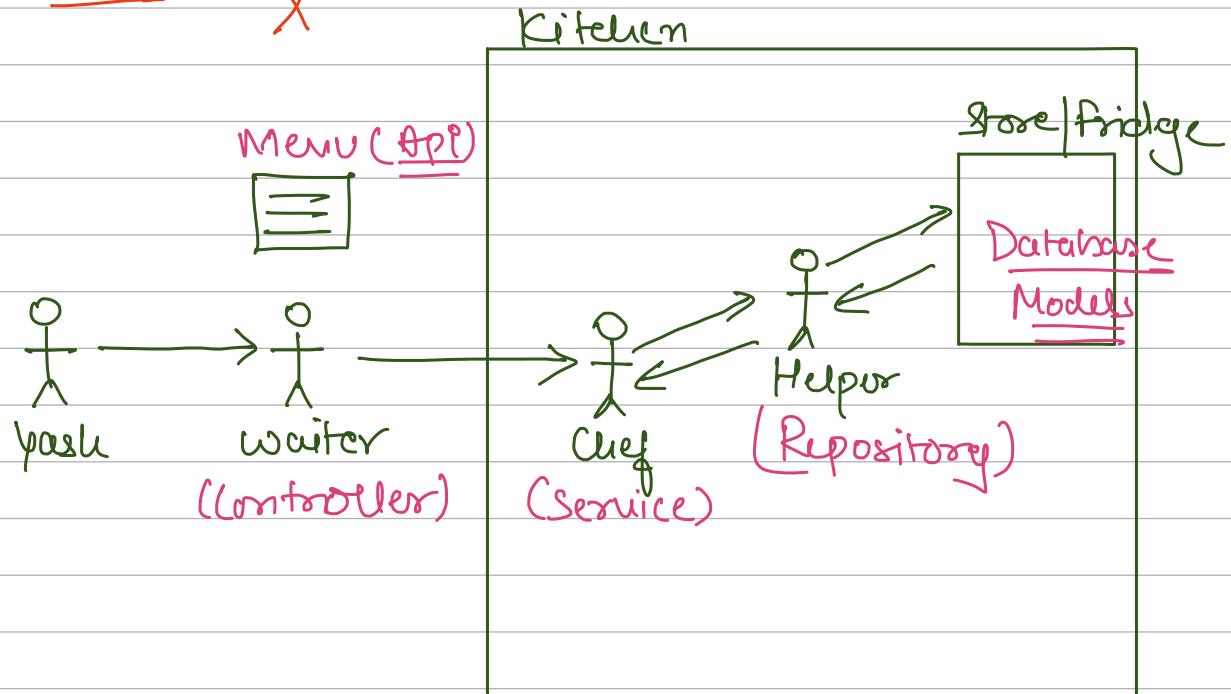
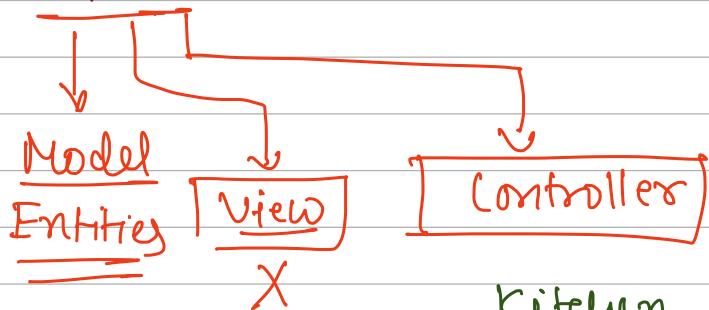


2) At least some of the requirements should be in working state.

→ Code all Models → Classes that we came up in class diagram

→ Start coding req by Req.

MVC Architecture \Rightarrow to follow SRP



Controller: Interface through which client interacts with the system

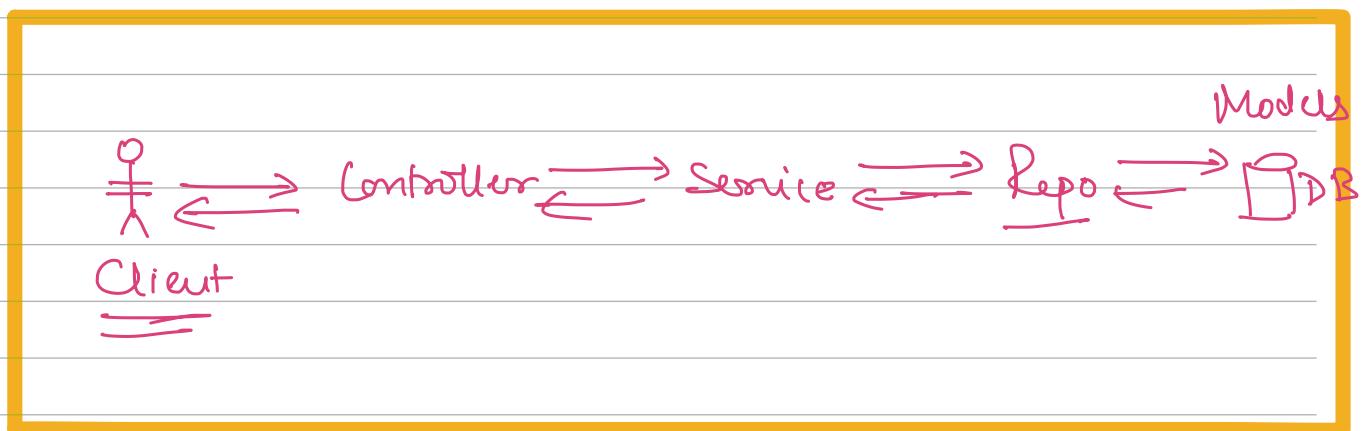
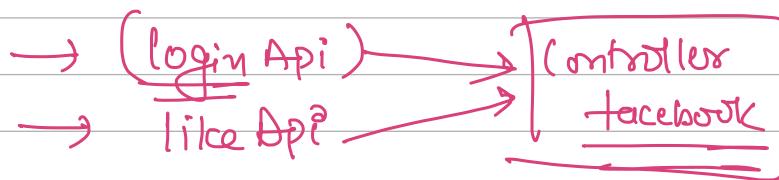
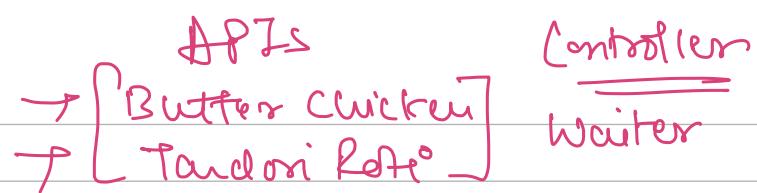
(very light weight)

Takes input from the client and passes onto service

Service : Algorithm | Business logic

Repository: Classes that will be used to interact with DB
(DAD)
↳ Data Access Object

Models : Tables in DB



Design Tic Tac Toe

Overview


 know the system

 Don't know the system

input = CMD or API

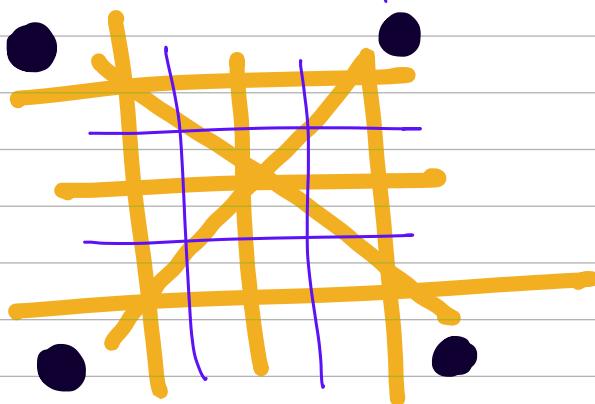


Note: Features that we are going to come up as requirement for TTT will almost be same for other games also.

Requirement gathering

- 1) Size of board should be $N \times N$ ($N \geq 3$)
- 2) No of players will be k ($k \geq 2$ & $1 < k < N$)
- 3) Players can choose symbols
(No 2 players should have same symbol)
- 4) Will there be a Bot? YES
 - ↳ Computer player (Max 1 bot is allowed)
- 5) Bot can have difficulty levels
 - ↳ easy | Normal | Hard
- 6) Who will start game?
 - ⇒ List of players
 - ⇒ At the start of the game, we will randomise the list and player will play in that order

7) How check winner?



- User will win the game if same symbol across row | col | diag.
- User will win the game if same symbol across all corners

8) When game will end?

- Someone wins the game
- Draw

g) UNDO

Unlimited UNDO's

X	O	
X		-
O		

Nikita
Ogarkina

leaderboard X

toeoffit X

leaving X

timelimit X

genreAssist X

History X

Reset X

cells Blocked X

User diagrams

Assignment



} Perfactory

if (pen = Addcel)

return new Addcel;

new smooths?
new lines?

Addcel {
 } Writing w/
 | strategy = new smooth()

write()

{