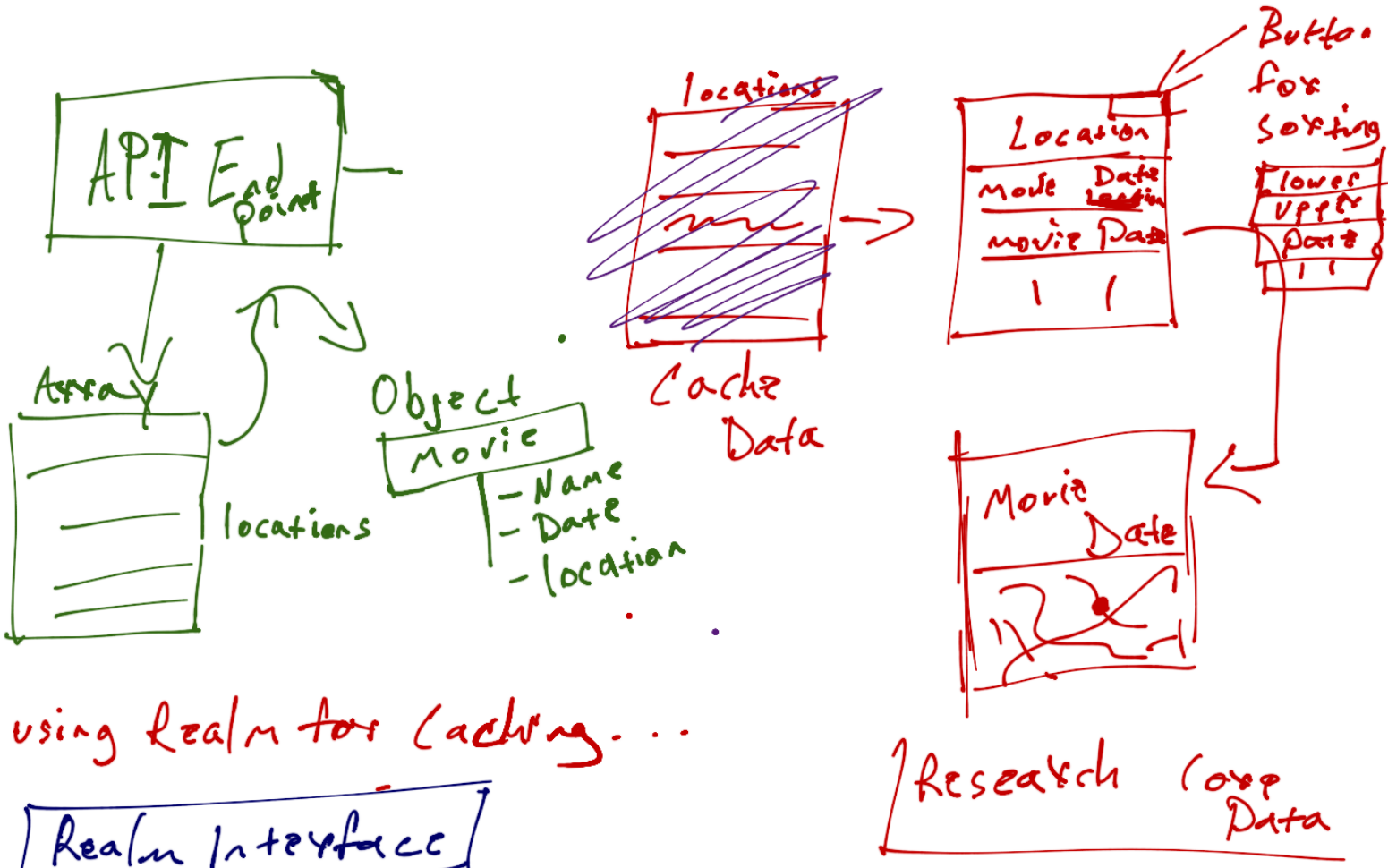



if using Realm for caching...

Realm Interface

- make Store

- get locations
- get Movies

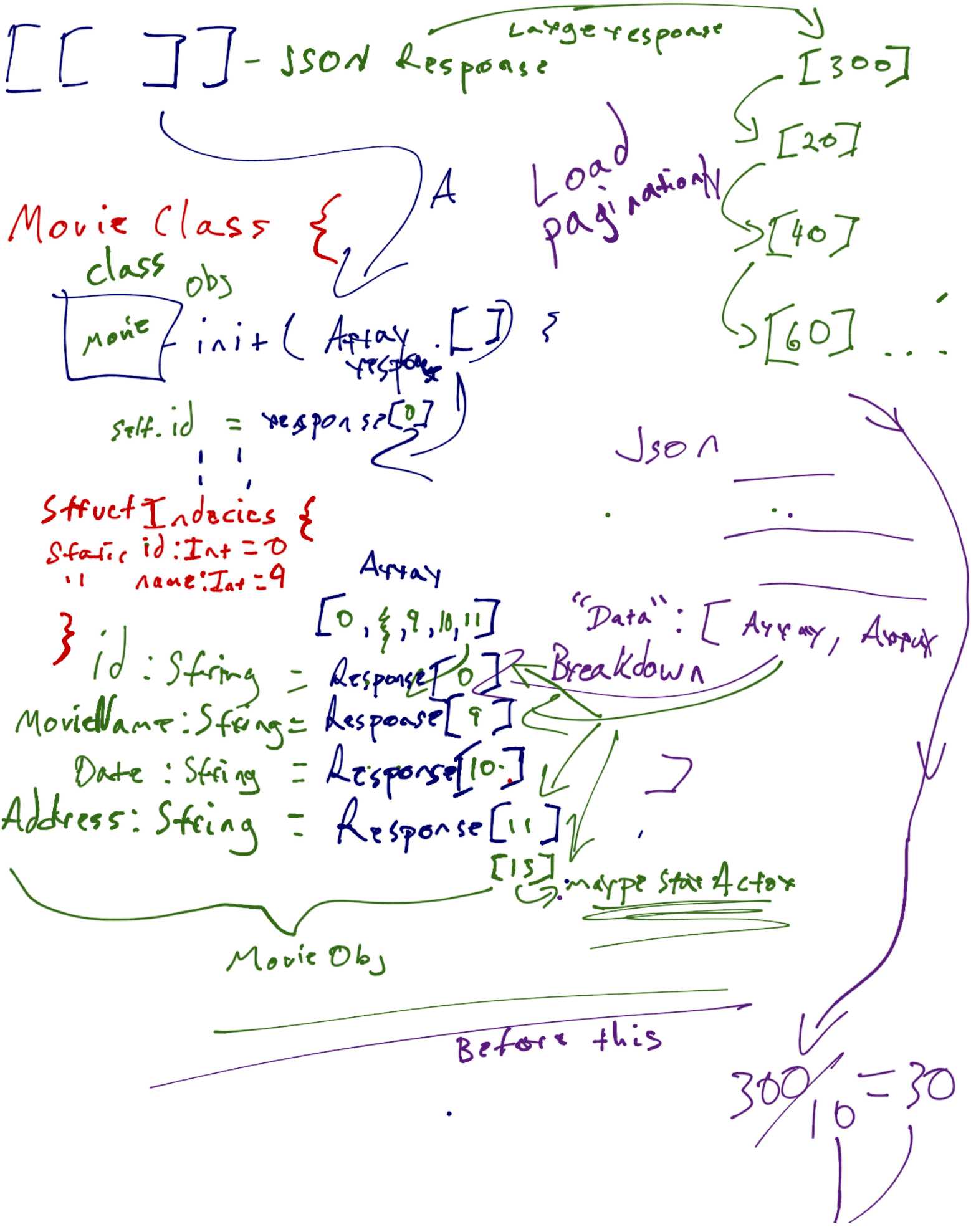
Hierarchy

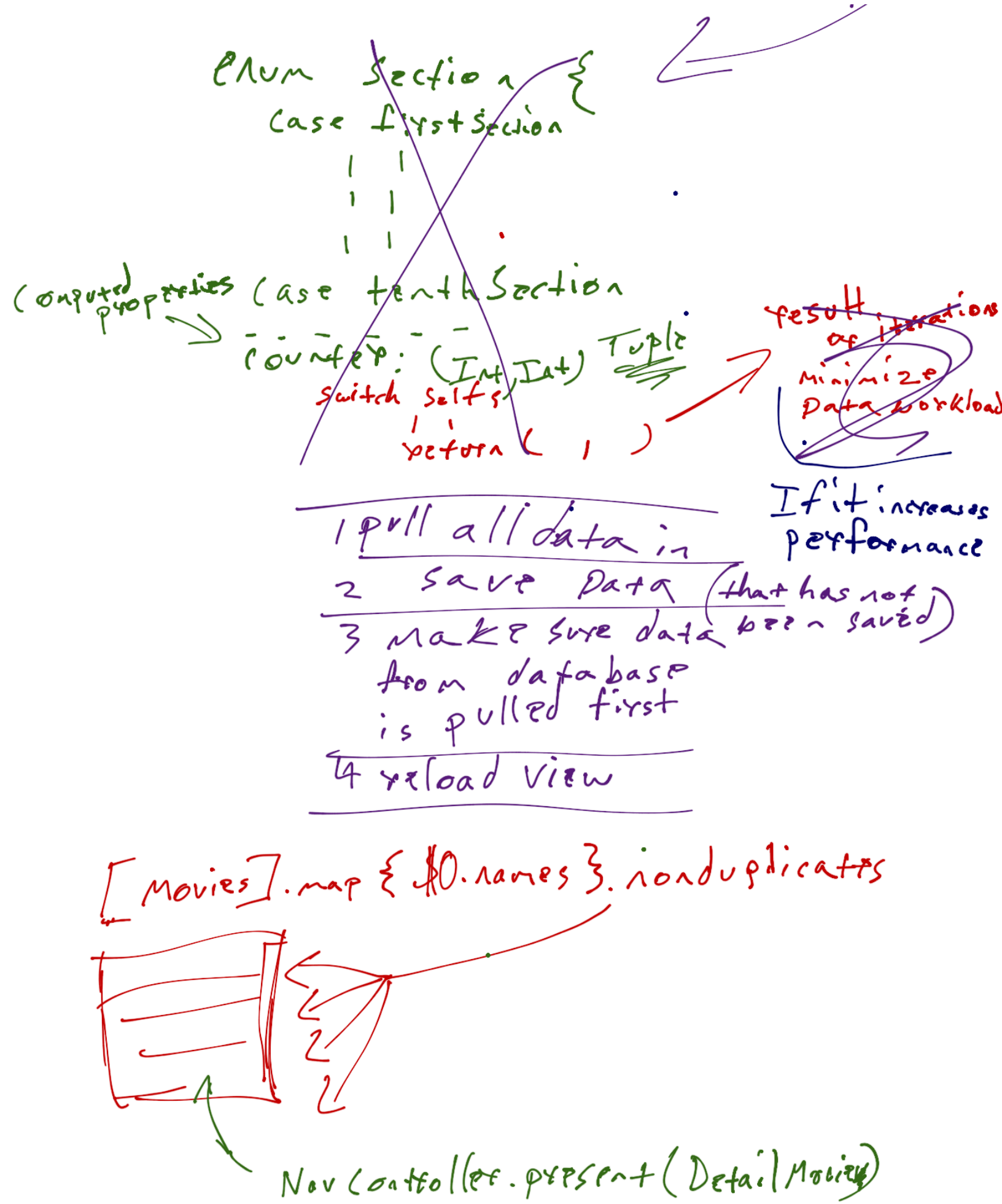
Models
- location
- Movie

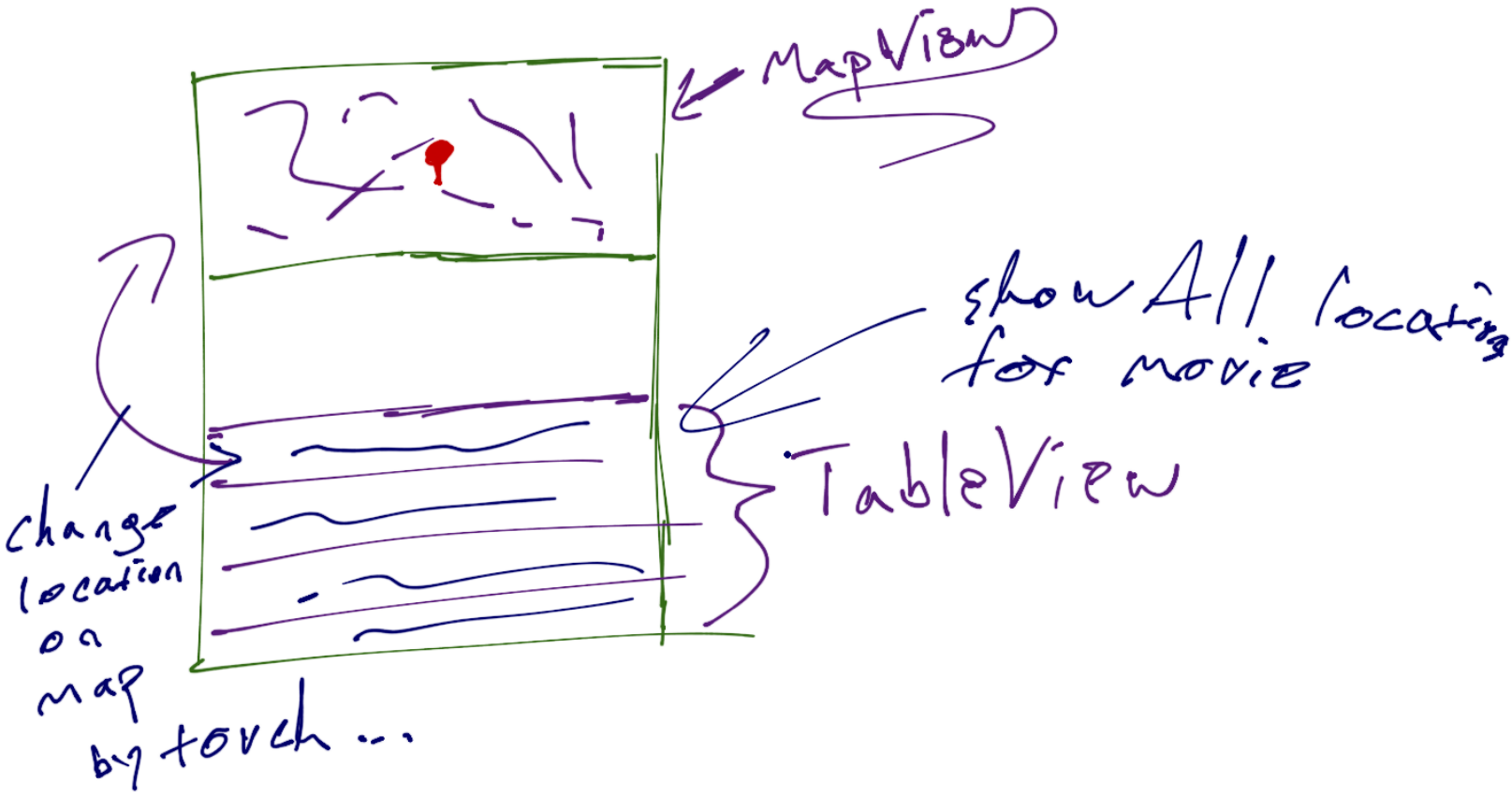
ViewControllers
- Location VC
- Top Movies VC
- Movie Detail VC

Model Controller
- Database Interface
- AppState

Views


[[ ]] - JSON Response
Large response
[300]
[20]
[40]
[60] ...
Load pagination
A
Movie Class {
class obj
movie
init( Array .[] ) {
response
self.id = response[0]
Struct Indecies {
Static id: Int = 0
" name: Int = 9
} id: String = Response[0]
MovieName: String = Respoase[9]
Date: String = Response[10]
Address: String = Response[11]
[15]
Array
[0, {, 9, 10, 11]
JSON
"Data": [ Array, Array
Breakdown
maype static Actor
Movie Obj
Before this
300/10 = 30


enum Section {
case firstSection
case tenthSection
(computed properties
counter: (Int, Int) Tuple
switch self {
return ( , )
result
of iterations
Minimize
Data workload
If it increases performance
1 pull all data in
2 save Data (that has not been saved)
3 make sure data from database is pulled first
4 reload view
[Movies].map { $0.names }.noduplicates
NavController.present(DetailMovie)




Make Movie Arr Unique

```
func filterMovies() {
    movieNames: [String] =    .map {$0.comp}
      var newArr: [Movie]
    Loop
        for mov in moviesArr
      - 1st index
      - append to new
      - remove from
}
```