* **System Design**

📦System Design

┣ 📂config app configuration files.

┃ ┗ 📜config.env

┣ 📂controllers controllers layer which handle requests.

┃ ┗ 📜userController.js

┣ 📂data-access database access layer (contains queries).

┃ ┗ 📜repository.js

┣ 📂error-handling error handling layer.

┃ ┗ 📜appError.js

┣ 📂models database models schemas.

┃ ┗ 📜userModel.js

┣ 📂posts posts filesystem.

┃ ┣ 📂postid1

┃ ┃ ┗ 📜post-id-time.png

┃ ┗ 📂postid2

┣ 📂routes routers layer according to restful api standards.

┃ ┗ 📜userRoutes.js

┣ 📂scripts misc. scripts and helper methods.

┃ ┗ 📜catchAsync.js

┣ 📂service service layer which contain business logic.

┃ ┣ 📜service.js

┃ ┗ 📜userService.js

┣ 📂subreddits subreddits filesystem.

┃ ┣ 📂subredditname1

┃ ┃ ┗ 📜subreddit-id-time-icon.png

┃ ┗ 📂subredditname2

┣ 📂unittest unit testing layer .

┣ 📂users users filesystem.

┃ ┣ 📂username1

┃ ┃ ┗ 📜user-id-time-profile.png

┃ ┗ 📂username2

┣ 📜app.js application Creation.

┣ 📜package-lock.json

┣ 📜package.json

┣ 📜README.md

┗ 📜server.js server from where to start app.

* Software Architecture (Layered approach)
  1. Description:

Application is consists of layers,  each layer performing a specific role within the application.

* 1. Layers:
     1. Entity Layer (Domain)

Contains database models and schema which describe database collection and connect to it.

(ex: User – Post - Subreddit).

* + 1. Database Access Layer (Repository)

Contains database queries using above models

**Should be generic layer** .

* + 1. Service Layer

Contains All Business logic of API.

* + 1. Controller Layer

Handles requests and responses for all routes

* + 1. Routes Layer

Mapping endpoints to its handler in Controller Layer

* + 1. Error handling Layer

Handle all kind of errors occurs in previous layers

* Software Design Patterns
  1. MVC
  2. Dependency Injection
  3. Singleton
  4. Follow SOLID principle.

Useful Links:

<https://scoutapm.com/blog/nodejs-architecture-and-12-best-practices-for-nodejs-development>

<https://dev.to/pacheco/designing-a-better-architecture-for-a-node-js-api-24d>