

Requirements:

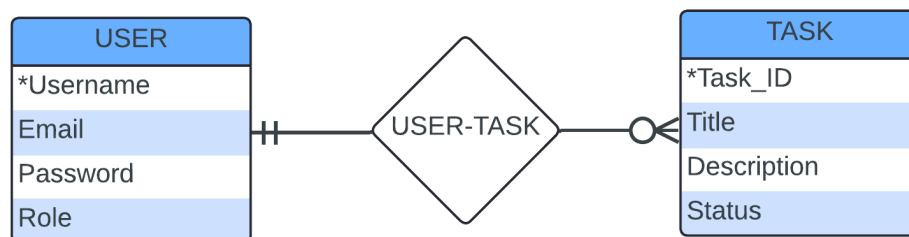
Functional Requirements:

- Register, and Login Users
- Get Users (All , By Username)
- CRUD operations on Tasks (Update Task Status)

Non-functional Requirements:

- Role Based Access Control (RBAC)
 1. Users can CRUD on their own Tasks
 2. Admin can Get Users, and CRUD on any task.
- Data Persistence using MySQL
- Robustness
- Use NodeJs, Express Framework

Data Model:



Architecture:

Express Server: Routes, Middleware

CRUD App: MVC

My Layered Architecture:

Client

|
| (HTTP Request)
|

Routing Layer

- routes/tasks.js
- routes/users.js
- routes/auth.js

|
| (Route matched, request forwarded)
|

Middleware Layer

- middleware/authMiddleware.js
- middleware/errorHandler.js
- middleware/roleMiddleware.js

|
| (Pre-processing: Auth(RBAC), Logging, ErrorHandling etc.)
|

Controller Layer

- controllers/taskController.js
- controllers/userController.js
- controllers/authController.js

|
| (Use Cases(Business Logic): CRUD, Login/Register Interacting with Models)
|

Model Layer

- models/Task.js
- models/User.js
- models/db.js

|
| (Data operations, Database interactions)
|

Database (MySQL)