First of all I used MongoDB as database for saving the data of user so I build a user schema.

Second to have users I have to register users first so I build a page where can register user and send the data to database to save it and sends the data “-password” as respond also a token that includes the user id to verify the logged in user and also I created middleware to protect the sign up and login “discuss later” so if a user logged in he can’t access sign up and login pages and will be redirected to home.
Another thing that the mail and phone number must be unique.

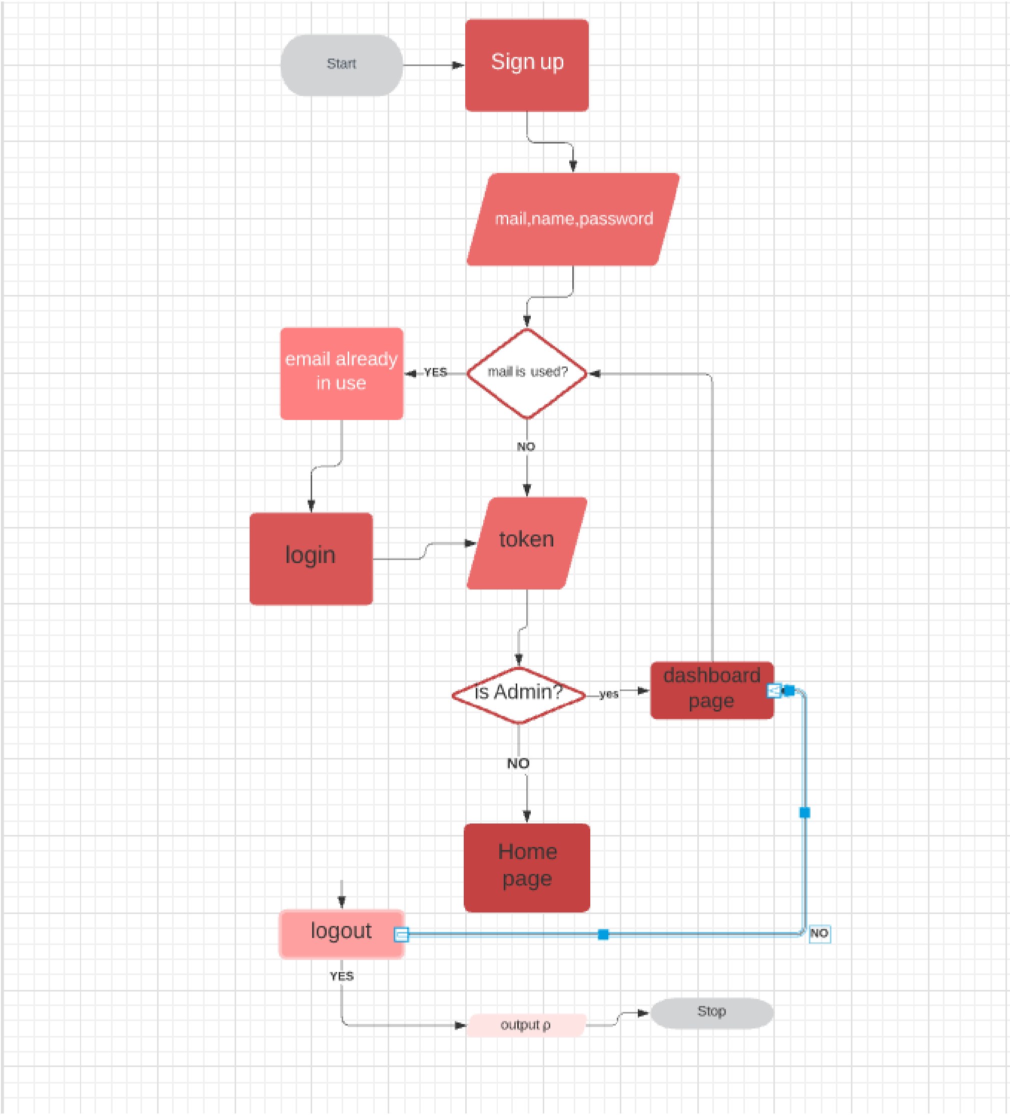
Then for authentication I build the log in page which will match the mail and password with the data saved on database if the mail and password matches then will send the name and password also a token to be saved in the browser to verify the user.

If the user is admin the isAdmin will be true and using the generated token he will have access to the dashboard page if not, he will be redirected to home page.

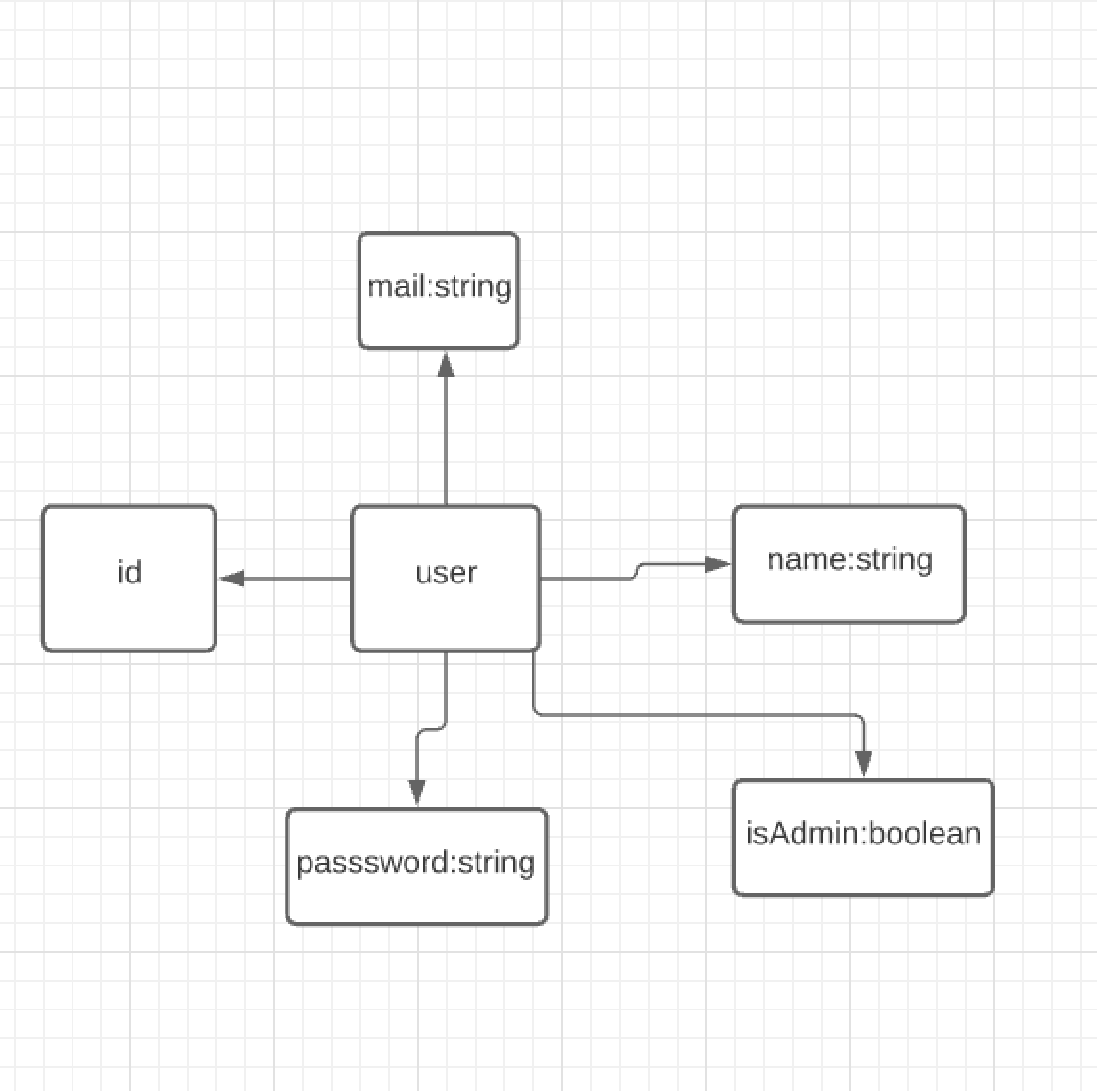
To achieve that I used middleware with the token and check if this user is admin or not.

Finally for logging the user out I can’t destroy the token so I will send invalid token that will expires fast.

Flow chart of the server



ERD diagram for user



Packages used:

1. **express: to handle the requests and the routing.**
2. **bcryptjs: to encypt tha password.**
3. **colors: to colorise the logs.**
4. **dotenv:to use variables from .env file.**
5. **jsonwebtoken: to generate the token.**
6. **jwt-check-expiration: to check if the token expired or not.**
7. **mongoose:: to deal with mongoDB.**
8. **nodemon:to restart the server with every edit.**