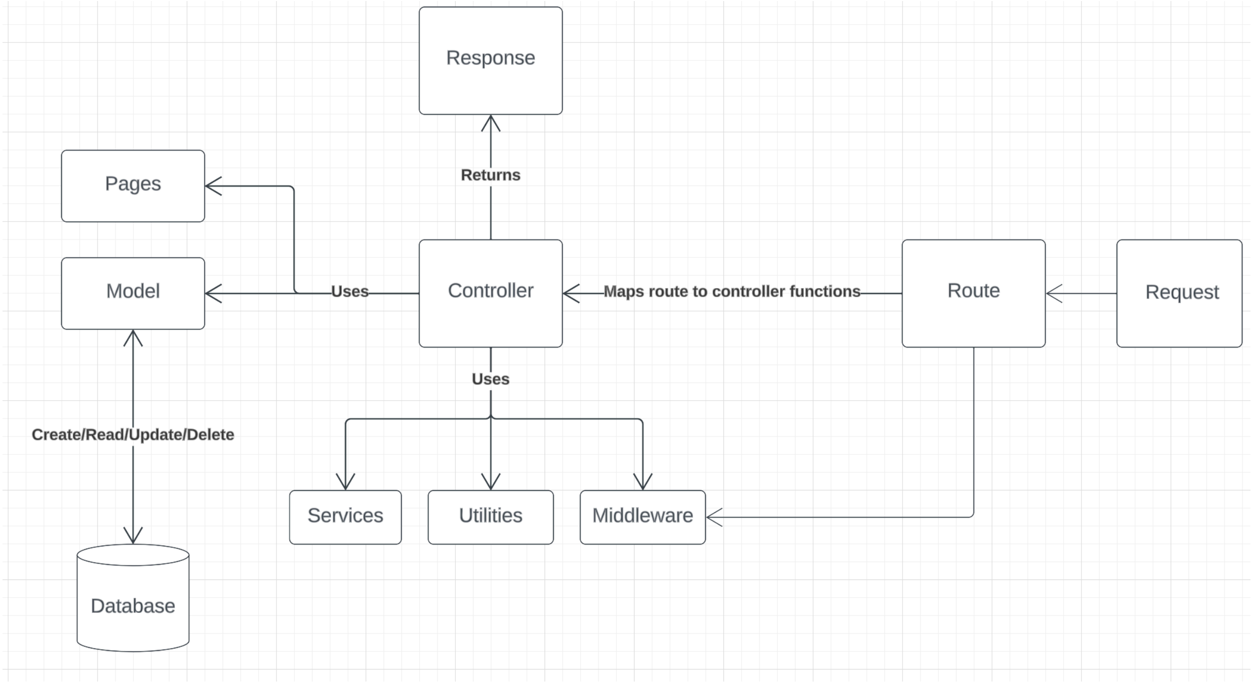
ISYS2160 LocalShop Design Documentation



Express server:

Controller:

- Controllers are responsible for handling the application's logic.
- They receive requests from the routes and interact with other components such as Model to access the database, to return responses back to the client.
- Controllers are associated with specific routes and HTTP methods.

Middleware:

- Middlewares are request handlers that have access to the application's request-response cycle, they access and modify the request or response data.
- Middleware functions are typically used to handle tasks like authorization and error handling in this application.

Routes:

- Routes define the URL paths the client can access and the corresponding endpoints. They map URL paths to Controller functions.
- Thus, Routes are responsible for directing requests to the right controller methods based on the HTTP method and URL.
- Routes can use Middleware to perform pre-processing on requests before directing them to the Controller.

Services:

- Services include business logic that is used by Controllers or other components of the application.

- Controllers can delegate the complex business logic to services, keeping them clean and focused.

Utilities:

- Utilities or "utils" contain generic and reusable functions.
- They are used for filtering products and calculating the total value of carts.

React:

Pages:

- Pages contain high level views that represent individual pages of the application.
- Each page corresponds to a specific URL path and can use Components to help build the layout.

Components:

- Holds reusable UI components. These components are small, self-containing parts of the user interface that can be reused across many pages.
- Components include things like dashboard, side bar, headers, and more. Using components promotes code reusability.

Assets:

- Contains static assets, which are often referenced to provide visual content.

Redux:

- Handles user authentication, shopping cart, orders, products, and admin functions by connecting to the backend via APIs and updating the frontend state accordingly.

Database:

Models:

- Models define the structure and schema of the application data.
- Models interact with the database to perform different operations on data (Create/Read/Update/Delete).
- Models are mostly used by Controllers to retrieve data from the database or to perform database modification.

Database:

- The Database holds application data, including user information, product details, orders and more.
- Uses MongoDB as the database management system.
- Database provides consistency, the data remains the same after server or application restart.