# Textbook

```
{
        id :  int,
        title: str,
        chapters_id: [id: int] -> array of chapters id
}
```

*api/textbook/{id}*

# Chapters

```
{
        id: int,
        title: str,
        pages_id: [id: int] -> array of pages id
}
```

*api/chapters/{id}*

# Page

```
{
        Id: int,
        content: {
                header: str,
                left_content: str,
                right_content: str,
                img_src: {
                        top_right: str,
                        bottom_right: str,
                        top_left: str,
                        bottom_right: str } },
        config: {
                left_width: Int,
                right_width: Int,
                mirrored_view: Bool,
                completed: Bool },
}
```

*api/pages/{id}*

# User

```
User
{
        id: Str,
        name: Str,
        email: Str,
        pwd: str, -> encrypted
        occupation: Str,
        authorised: Bool, -> to edit web-app
        Achievements: [] -> progress tracking?
}
```

*api/user/{user_id}*

# Action Log / User Event

```
Action logs
{
        events:
                [
                        {
                                id: Str,
                                type: Enum <Int>, -> type of event (0 = login, 1 = register..)
                                creationTime: Timestamp,
                                user: { id: Str, name: Str },
                                description: str -> describe the action
                        }
                ]
        }
}
```

*api/action_logs*

# Scenario

```
scenario
{
        {
                id: int,
                title: str,
                description: str,
                learning_objectives: str,
                content: { } -> objects of data needed for scenario, example below
                model_simulation: bool -> indicate would scenario utilise pulse model
}
```

```
content
{

}
```

## ECMO stats

Vitals

```
{
        arterialBp:iInt,
        systolicBp: int,
        SpO2: int,
        temperature: double,
        pulseRate: int,
        CO2: int,
        etCO2: int,
        awRR: int
}
```

Stats

```
{
        ecmoState: bool,
        ecmoRPM: int,
        sweepGasFlow: double,
        patient_id: int,
        scenario_id: int
}
```