Summary of Technology

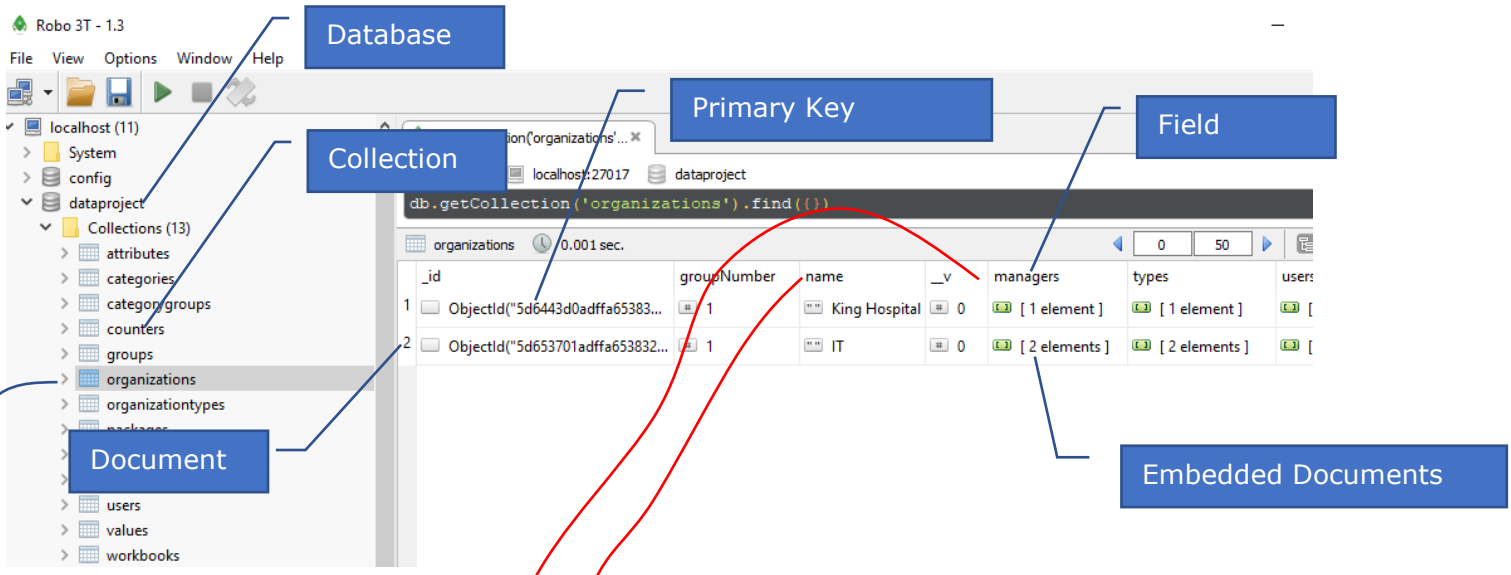
1 Technology Overview

Name		Description
Database		
	MongDB	document-based database
	MongGoose	an Object Data Modeling (ODM) library for MongoDB and Node.js
Back-end		
	Node.JS	uses JavaScript on the web server
	Express	a Node.js web application.
Front-end		
	React	A JavaScript library for building user interfaces
	Material-UI	UI Components Framework
	Axios	Promise based HTTP client for the browser and node.js
Useful tools		
	Swagger	A professional toolset to help developers design, build, document, and test RESTful Web services
	mocha	JavaScript test framework
	chai	a BDD / TDD assertion library
	marge	Marge (**m**ocha**w**esome-**r**eport-**g**enerator) generates a report that helps visualize your test suites.

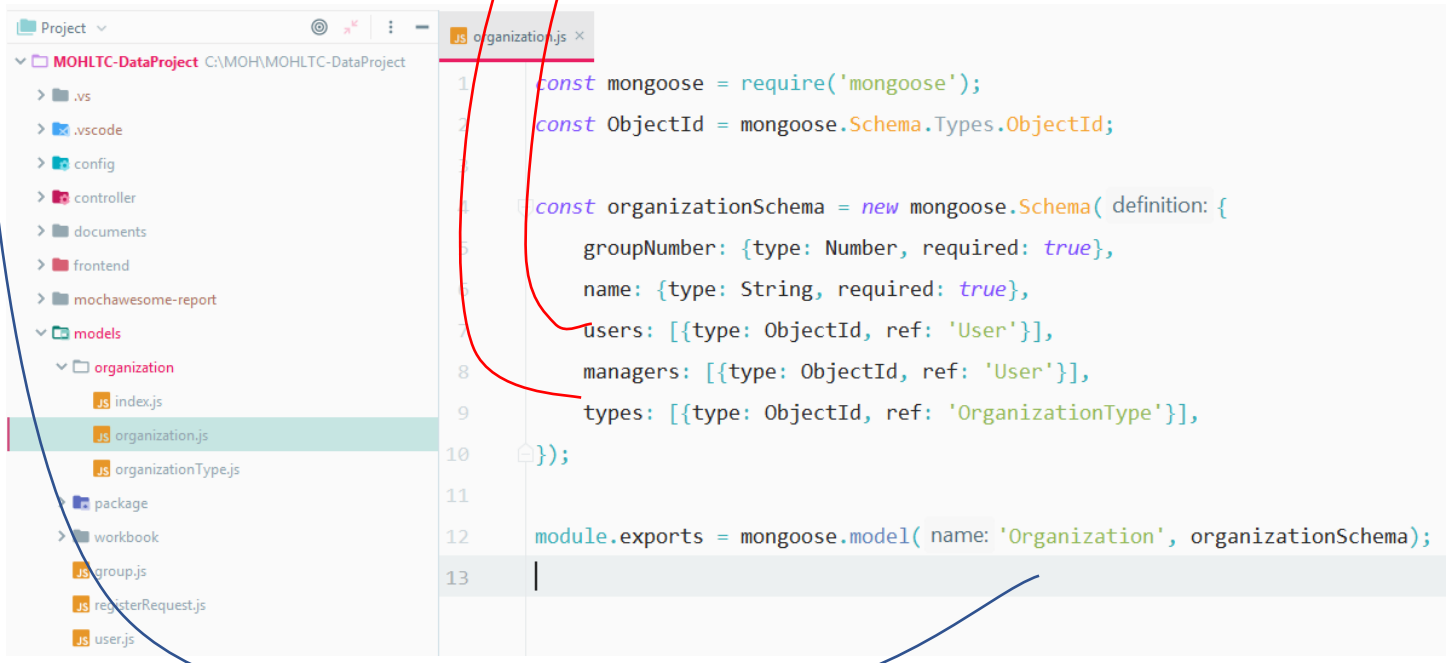
2 MongDB & MongGoose

MongoDB works on concept of collection and document. The following table shows the relationship of RDBMS terminology with MongoDB.

RDBMS	MongoDB
Database	Database
Table	Collection
Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by mongodb itself)



Mongoose is used to translate between objects in code and the representation of those objects in MongoDB.



3 Node.js & Express

A simple example to show how to retrieve all organization documents on the database and return them to the API caller.

```
const express = require('express');
let router = express.Router();      // Creater a new router object
const Organization = require('../models/organization/organization');
router.get( path: '/api/v2/organizations',  handlers: async (req, res, next) => {
    try {
        const organizations = await Organization.find();
        if (organizations[0]) {
            return res.status( code: 200).json( body: {organizations});
        }
        return res.status( code: 204).json();
    } catch (e) {
        next(e);
    }
```

Request URL

```
http://localhost:3000/api/v2/organizations
```

Server response

Code	Details
200	Response body


```
{
  "organizations": [
    {
      "users": [
        "5d5eafbcc3b85c1c5cc81dc8"
      ],
      "managers": [
        "5d5eafbcc3b85c1c5cc81dc8"
      ],
      "types": [
        "5d603977d7ac99d4a4f4d0ce"
      ],
      "_id": "5d6443d0adffa653832f057d",
      "groupNumber": 1,
      "name": "King Hospital",
      "__v": 0
    },
    {
      "users": [
        null,
        "5d5eafbbc3b85c1c5cc81dc7"
      ],
      "managers": [
```


Download

4 React

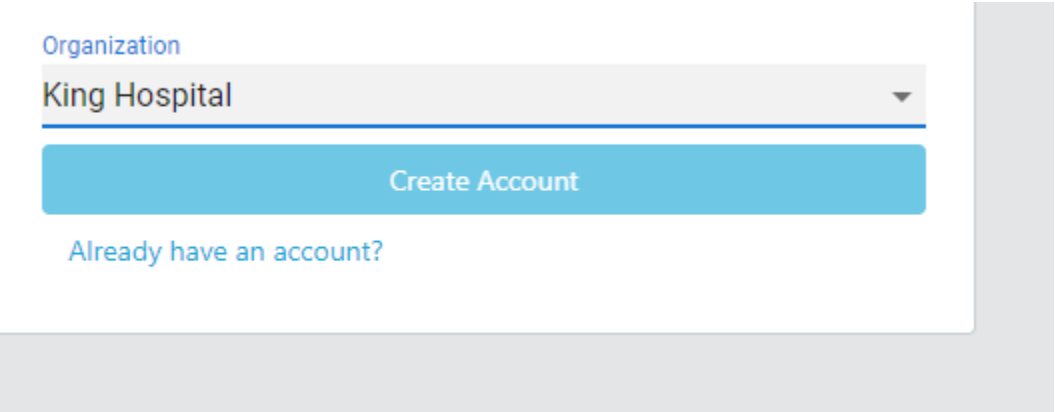
We use AXIOS to consume our GET Api to get all organizations data in the back of the front-end.

```
export async function getAllOrganizations(groupNumber) {
  let urlStr = config.server + '/api/v2/organizations';
  try {
    const result = await axios.get(urlStr);
    return result.data.organizations;
  } catch (e) {
    throw e;
  }
```

Then transfer them to the state of React component.

```
initialGroups = () => {
  getAllGroups().then((dbGroups) => {
    if (dbGroups[0]) {
      getAllOrganizations().then(dbOrganizations => {
        this.setState( state: {
          dbGroups,
          selectedGroupNumber: dbGroups[0].groupNumber,
          dbOrganizations,
          selectedOrganizations: dbOrganizations,
          selectedOrganization: dbOrganizations[0].name
        });
```

In the end, the browser Displays the organization data to the end-user by using the select menu in a textfield(Material-UI component).




```
<TextField
  select
  InputLabelProps={{shrink: true}}
  label="Organization"
  value={this.state.selectedOrganization}
  onChange={this.handleChange('organization')}
  margin="normal"
  fullWidth
>
  {this.state.selectedOrganizations.map(option => (
    <MenuItem key={option._id} value={option.name}>
      {option.name}
    </MenuItem>
  ))}
</TextField>
```