

DoTo

A Task Management Application Created While Conducting an Exploratory Analysis With SQL

A PROJECT REPORT

Submitted By

HIMANSHU BERIWAL (Reg No: RA1811028010007)

RUSHIL RAI (Reg No: RA1811028010022)



S.R.M. Nagar, Kattankulathur, Kancheepuram District

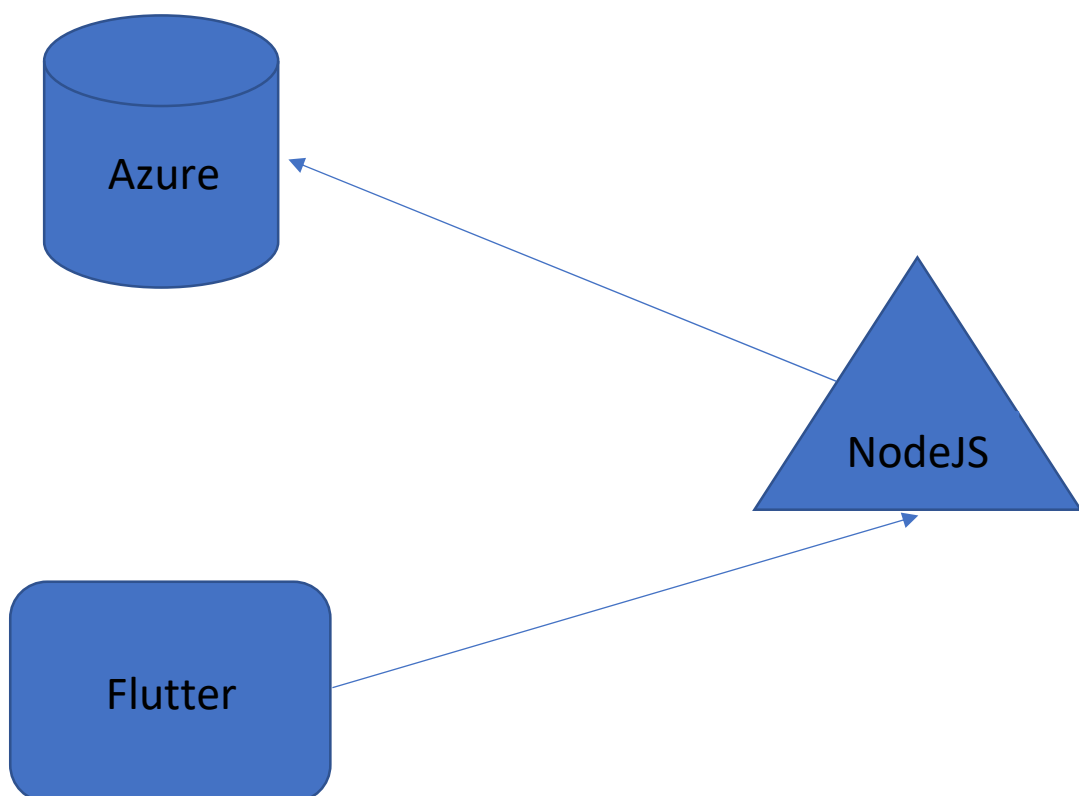
May 2021

Abstract

Task & Project Management Application. Allows Users to Easily Manage Tasks by following the concept of Separation of Concerns. Users can have separate Lists and Each List will have its own List of Tasks. Allows faster and more efficient workflow while still having the satisfaction of marking off tasks from a To-Do list. Provides an Online Database to allow sync across platforms, which ensures the user needs not worry about losing info or worrying about the device they use.

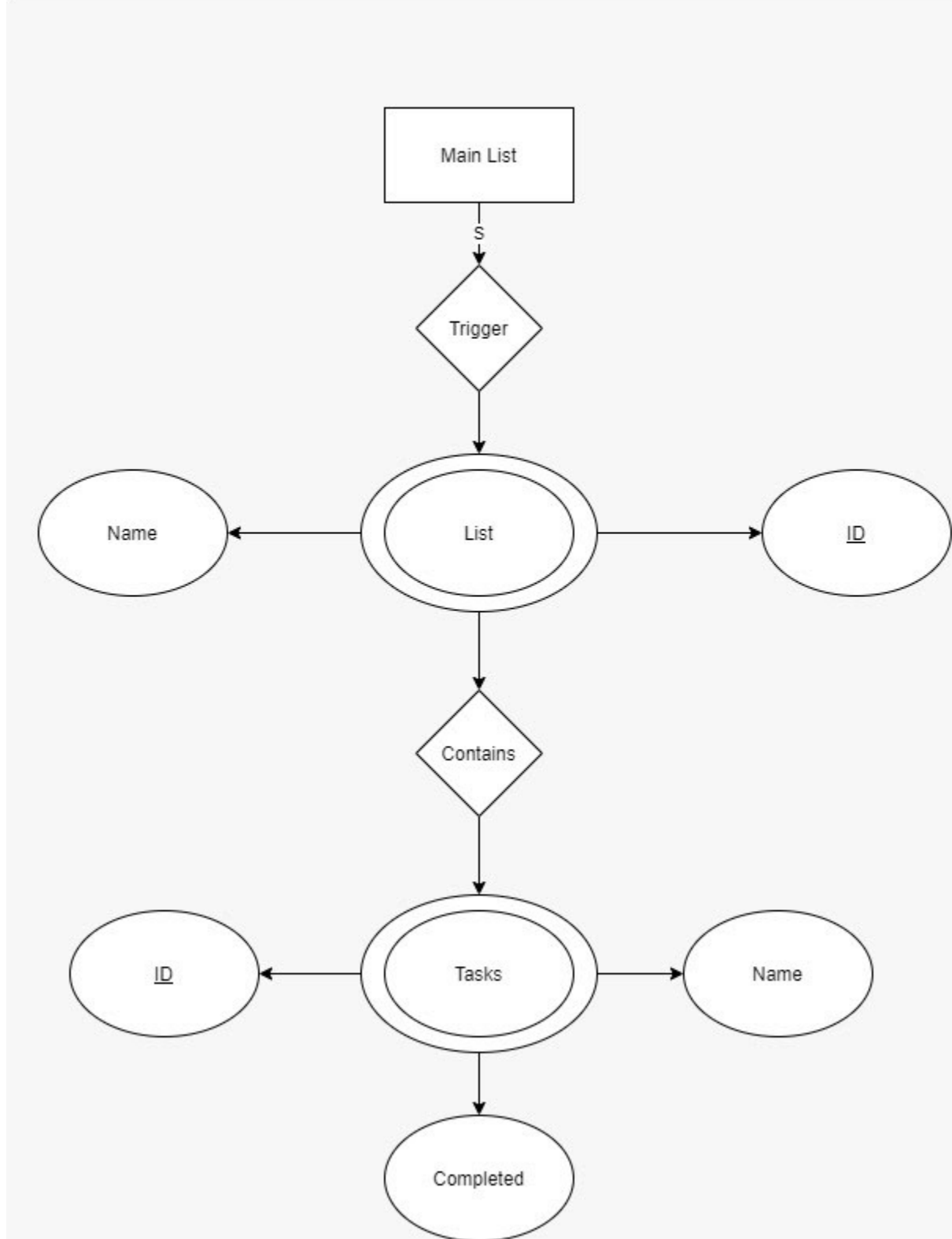
Technologies Used

- Database - SQL
- Database Hosting - Microsoft Azur
- NodeJS - APIs
- Flutter – Desktop App



E-R Diagram

DO-TO : ENTITY RELATIONSHIP DIAGRAM



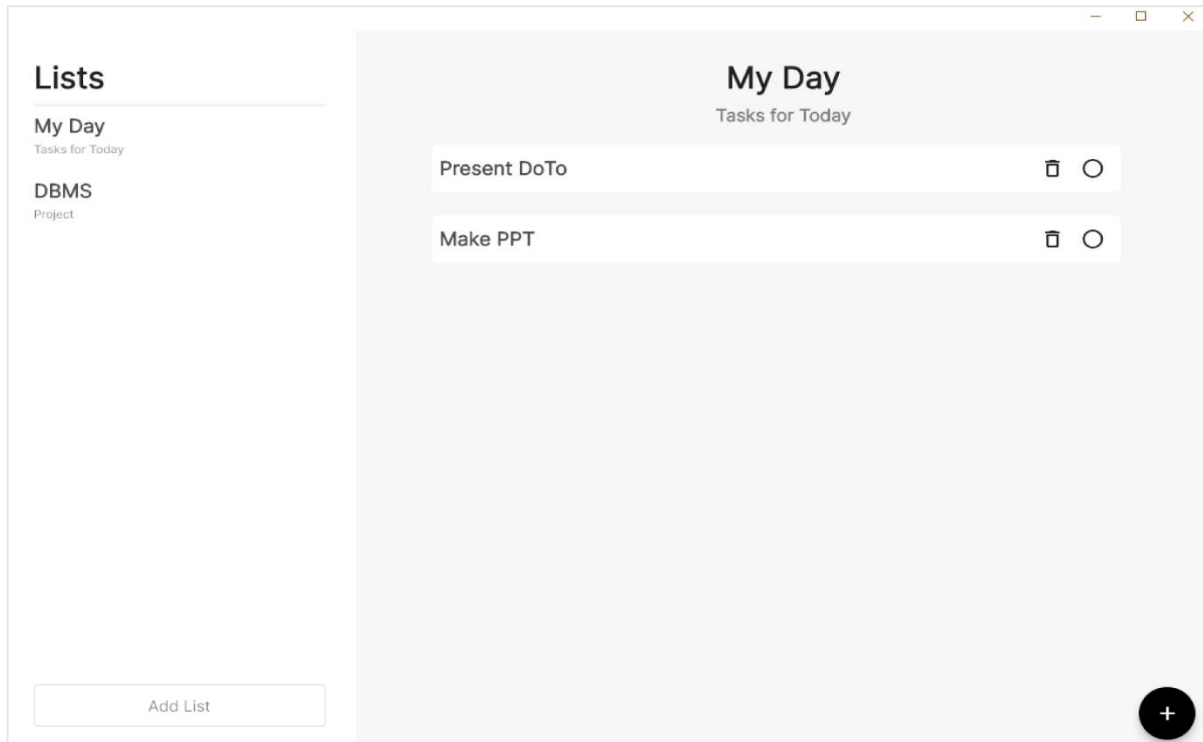
Connection

```
1 var express = require("express");
2 var sql = require("mssql");
3 var cors = require('cors');
4 var app = express();
5
6 app.use(express.json());
7 app.use(cors());
8
9 var server = app.listen(process.env.PORT || 8080, function () {
10   var port = server.address().port;
11   console.log("App now running on port", port);
12 });
13
14 var dbConfig = {
15   server: "",
16   database: "",
17   user: "",
18   password: "",
19   options: {
20     encrypt: true
21   }
22 };
```

Sample API

```
1 var express = require("express");
2 var sql = require("mssql");
3 var cors = require('cors');
4 var app = express();
5
6 app.use(express.json());
7 app.use(cors());
8
9 var server = app.listen(process.env.PORT || 8080, function () {
10   var port = server.address().port;
11   console.log("App now running on port", port);
12 });
13
14 var dbConfig = {
15   server: "",
16   database: "",
17   user: "",
18   password: "",
19   options: {
20     encrypt: true
21   }
22 };
```

Screenshot



Database

doto-db (doto-server-admin)

Showing limited object explorer here. For full capability please open SSDT.

Tables

- > dbo.list16 ...
- > dbo.list25 ...
- > **dbo.lists** ...

Views

Stored Procedures

Query 2

Run ☐ Cancel query Save query Export data as Show only Editor

```
1 SELECT TOP (1000) * FROM [dbo].[lists]
```

Results Messages

Search to filter items...

listID	listName	listDesc
16	My Day	Tasks for Today
25	DBMS	Fav Subject

Procedure

Each User is assigned with their own Database.

- Each Database contains a Primary Table of Lists.
- Each List added to the primary as a column gets its own Table using Triggers.
- Database was created on Microsoft Azure and the using its connection parameters, connected to an API platform built upon NodeJS.
- For all requests, the Application calls upon an API from the Node App, which then communicates with the Database on Azure, ensuring users cannot meddle with Data.
- The User facing Application was built using Flutter, and is able to run on all Desktop Platforms.
- The Flutter App uses HTTP requests to communicate with the API whenever the User performs any actions.

Conclusion

Our project implements the concepts of DBMS successfully to ease Task & Project Management for working Individuals.

Scope of Improvement:

Caching to be implemented using another Database self-contained locally with the User facing Application which shall eliminate all User Facing Delays in Data Synchronization.

GitHub Link: <https://github.com/rushilrai/DoTo>