**Deployment Requirements**

**1.0 Client Side Features -** The client side features that are carried out are the verification of data into the forms. This allows data to be entered correctly to the database in the correct format and of the correct type. The client side sends xmlHTTP requests or “GET” requests for information from the middleware which then organises the input data to interact with the database on the back end effectively “translating” the inputs from the web application.

The advantages of this methodology are that the front end is a browser application that is platform agnostic and is very lightweight requiring few resources to implement. The application server mutates the data between the front and back-end doing the majority of the processing which allows a dedicated server to do most of the number crunching for the application. The database server or back end is used primarily for storage and executing SQL statements as dictated by middleware requests. Thus the majority of the search functions and sql processing are carried out on the back end.

The advantages of using the middleware is that it enables the front and back end to communicate even though they are using different platforms and protocols. With little processing on the front end it enables enterprise level scaling for the intensive use of the application by many users. It also allows for the management of traffic both in and outbound by the middleware (limited in this particular application but the scope for it remains). The use of the middleware allows us to control access and permissions to the database by controlling the traffic to and from the database, thus implementing an effective crud matrix of permissions. Thus middleware is key in maintaining the integrity of the database. The design also allows for the flow of real time information between the users.

**2.0 Database Interactions -** The design choices made around the use of the postgres database were to limit the amount of information pulled from the database and processed in the middleware. To this effect the searching and analysis of notes was carried out using SQL in the database itself and sending only the results to the application, rather than sending the contents of the database to the middleware for regex partial searches and analysis functions. This enables the system to retain enterprise level scalability by reducing the need for large data communications between the database and application server. The use of SQL searches allows for relatively low complexity (O(1)). So the methods used allows for minimisation of data transfers, lower complexity of software design (in using SQL partial searches to meet the analysis feature requirements). This should allow for better scalability of the project for enterprise needs.

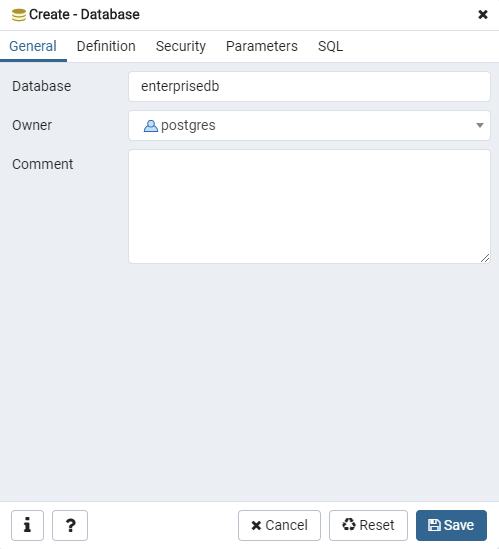
**3.0 - Front End -** The front end of the application is written in HTML5, CSS3 and Javascript using a bootstrap framework allowing for its implementation by users being only dependant on their platform being able to run a web browser, therefore the front end of the application can run on platforms as diverse as mobile phones, P.D.A’s desktop computers to a web capable fridge. The database is hosted on a Postgresql server and the middleware is a compiled go application. The use of golang allows the application server to run it’s binary on many different operating systems including android, windows, linux, macOS, netBSD and openBSD.

The use of the Postgresql database allows has many advantages. Firstly as an open source enterprise level database management system it has reduced costs while not sacrificing either performance or scalability when compared to commercial solutions. It also allows for full text search and can use multiple different indexing techniques for data structures if the data requirements change at a later date. Being able to run on Linux postgres also has lower OS system costs in running the server which may be an ongoing concern. It also has no vendor lock in or contractual obligations and licencing with a third party software provider.

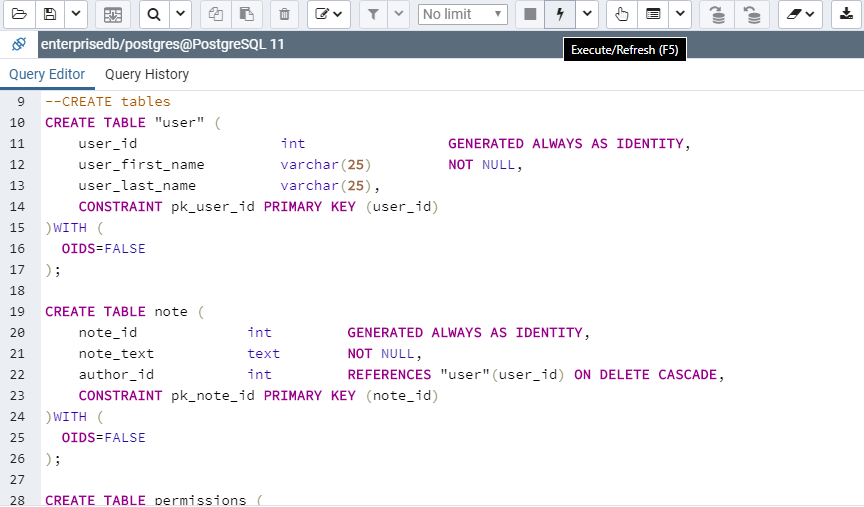
**4.0 - Installation Instructions**

**Enterprise Note Quick Start Guide**

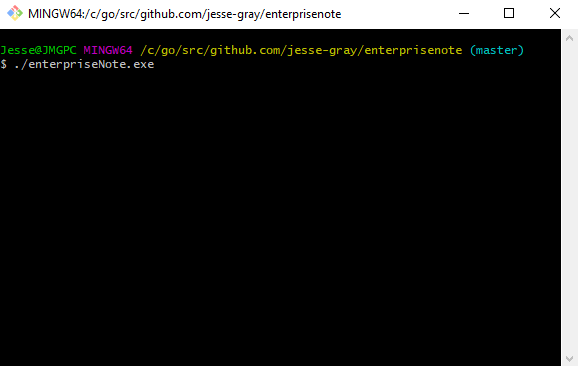
## **1. A PostgreSQL database must be created called *enterprisedb***

******

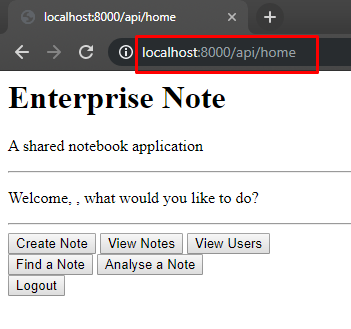
## **2. The script within CREATE.sql needs to be run within the *enterprisedb* database**

****

## **3. *enterprisenote.exe* needs to be running to start the server**

****

## **4. Open *localhost:8000/api/home* to open the web app**

****