

Enterprise Note

By James Hessel-Bowman and Joe van der Zwet

Team member contributions

Name	Preparation	Participation	Reliability	Comment
James Hessel-Bowman	10	10	10	
Joe van der Zwet	10	10	10	

Github Links:

[Frontend](#)

[Backend](#)

Missing Specs

Actual Requirements Backend

Requirement	Status	Comments
User can create account	[x]	done
Users can create note	[x]	done
User who creates note is owner of note	[x]	done
All users have access to the list of users	[x]	done on the front end
A note can be shared with other user	[x]	done
Note share read privilege	[x]	done
Note share edit privileges	[x]	done
Notes contain just text and meta data	[x]	ours also contains a subtitle and the ability to do markdown
Users can edit or read notes depending on privileges	[x]	done

Requirement	Status	Comments
The owner of a note can add more people to note.viewers and note.editors	[x]	done
The owner of the note is also the only one who can add new people to the list of editors and viewers	[x]	done
Third table that contains userid, user.viewers array, user.editors array for ease of sharing	[x]	done
Ability to apply said common settings to a note	[x]	done

Actual Requirements Frontend

Feature	Status	Comments
Login Page for selecting current user	[x]	done
Register Page for making new users	[x]	done
Home Page that shows all notes a user has access to	[x]	Pretty sure this is done
Note access page/component for displaying note information	[x]	It looks pretty cool tbh
Note edit button that allows changes with a submit button	[x]	done
Note access change button that opens access screen	[x]	done
Access screen has a list of users that can read and edit this note	[x]	done
Access screen has a dropdown of all the users and a radio button for no access, viewer, editor	[x]	done
Search bar for filtering and counting	[x]	done
User List Page	[x]	This is technically required and we have it twice. It could be removed.
Ability to filter notes you can access	[x]	done

Feature	Status	Comments
Users can count occurrences of the following among notes they have access to	[x]	done
A sentence with a given prefix or suffix	[x]	done
A phone number with a give area code and/or consecutive number pattern	[x]	done
An email address on a domain that is only partially provided	[x]	done
Text that contains at least three of the following case sensitive words. [meeting, minutes, agenda, action, attendees, apologies]	[x]	done
A 3+ letter word thats all caps	[x]	done

Usage Instructions

Initial Setup

Postgres

- Run Postgres.
- In PGADMIN go to servers on the left
- Right click, Create Server, Name: postgres
- Connection tab, hostname: localhost
- Save

Enterprise Note

- Extract the files to a folder
- Edit the .env file to point to your postgres
- Run the exe

Maintenance

None should be required. The program is very set and forget other than keeping on top of any bugs that might pop up that we haven't found yet.

Design Choices

Initially our plan was to use flutter for the front end but with some playing around we found it a bit clunky for a web interface. So we switched to React and implemented a full front end with that. We use Material Design as our UI framework because it looks amazing and is easy to use with react. Coding everything in typescript has been relatively easy to work out and making components is quite interesting and fun.

Features

Enterprise Note is a fully functional note taking application that could be used in every day life with the functionality it has now. So far included is:

- Full user registration
- Ability to create, read, update and edit notes
- Ability to share notes with other users including letting them edit your notes
- Ability to search the notes you have access for a list of predefined criteria
- Partial markdown support for notes.

Deployment requirements

- Access to a server where you can run an exe and use two ports.

Features that could have been client or server side

- Searching I think was our only feature that could have been both but when we implemented base64 encoding on the notes it made it impossible to do the searching via regex and postgres queries.

Explain database design choices

- Our database has 3 tables all linked though they probably don't need to be.
- We have a User, Note and UserSettings Table.
- The user is all the information the user requires to exist.
- The note is all the information in a note
- UserSettings is for a users commonly shared settings.
- The tables are technically linked with foreign keys I think but we don't actually join any of the tables because we didn't need to for our purposes.

Discuss how we deal with multiple browsers and OS's

The beauty of react and material design is that it works the same on all browsers and has built in support for media queries and break points so it should look great on all devices.

Go as a backend cross compiles to most operating systems so it should run on any kind of server or desktop as required.

Quick start guide for installing on a new server

Postgres

- Run Postgres.
- In PGADMIN go to servers on the left
- Right click, Create Server, Name: postgres
- Connection tab, hostname: localhost
- Save

Enterprise Note

- Extract the files to a folder
- Edit the .env file to point to your postgres
- Run the exe

Things that we not in the initial spec but had to be added to make it work.

- A decent front end. There is no way we were going to use standard html/css
- Markdown support for actual quality notes.
- Better searching because the standard requirements were gross.

Screenshot of us doing unit tests on the front end

```
PASS src/pages/Login.test.tsx
PASS src/pages/Register.test.tsx

Test Suites: 2 passed, 2 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        5.425 s, estimated 6 s
Ran all test suites related to changed files.
```

We opted not to do unit tests on the backend because we couldn't get the unit tests for mux to work and that was all we had to unit test for the backend. We have no other functions.