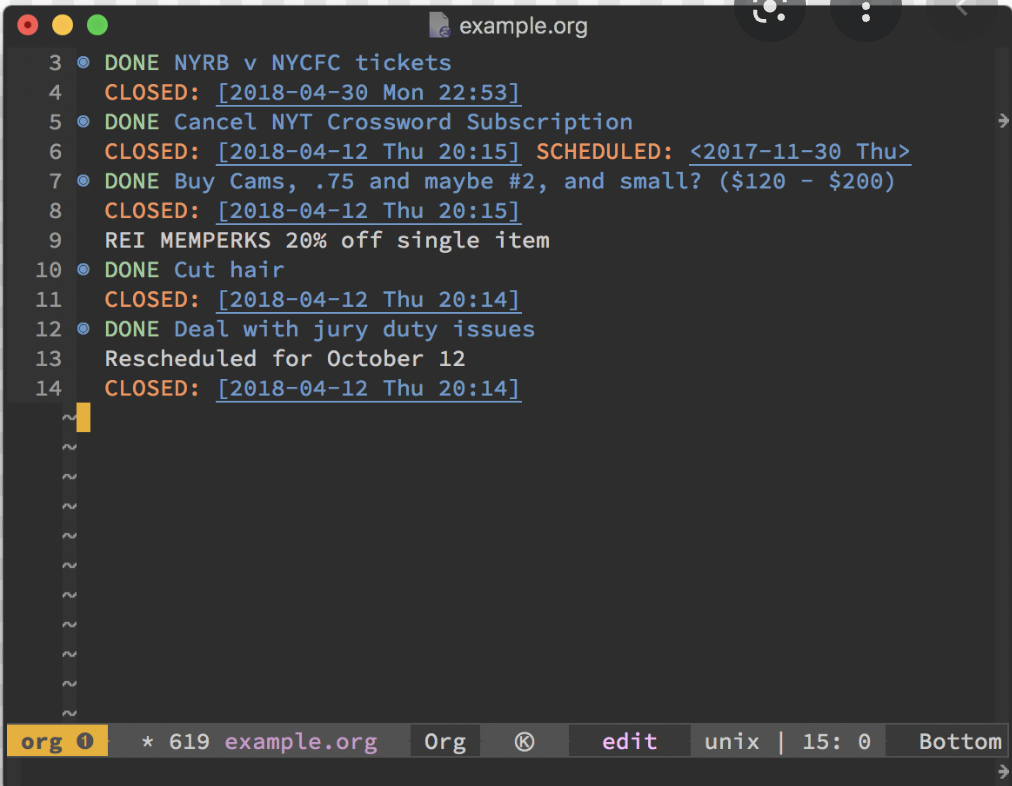
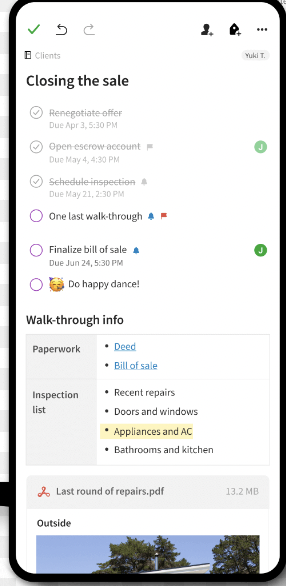
# **Note Keeping App**

## The state of the current world

* Note keeping apps have existed since computers have first been invented and are integral parts of the “user essentials” toolkit of any consumer using a smart device.



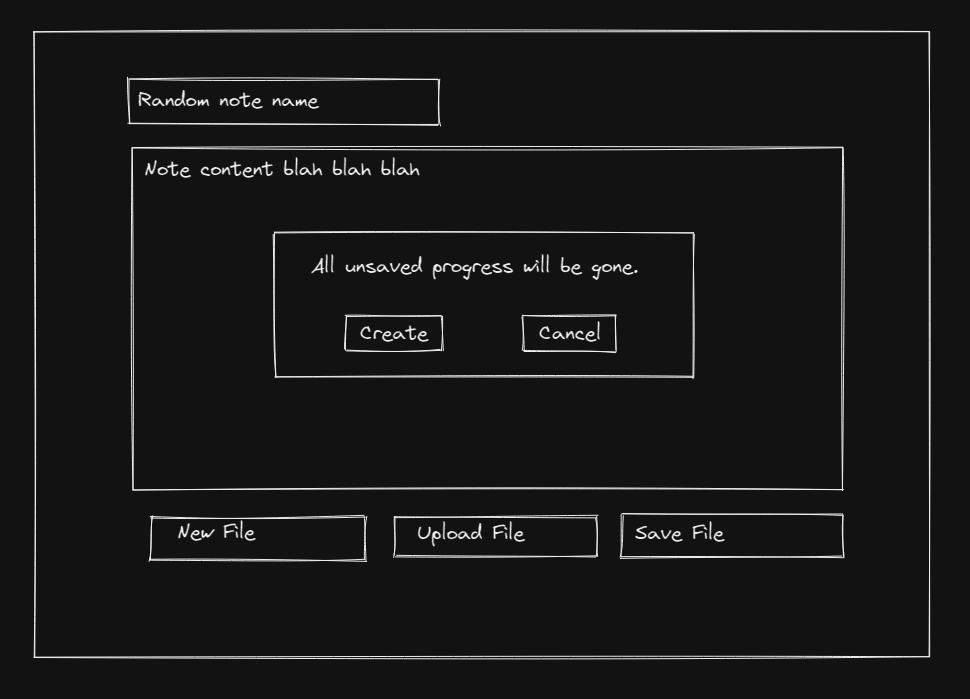
* They have existed in more arcane forms like Emacs’ org mode as well as more modern forms like Evernote and Google Keep which are hyper reliant on the cloud 
* However there are concerns for the average user in both cases. While Emacs and other local note keeping apps provide data ownership they are too complicated for the user and support on mobile devices and non Unix based systems is limited.
* On the other hand, cloud based note keeping apps have more multi device capabilities at the cost of sacrificing the user’s privacy and data ownership.

## Be the change you want to see

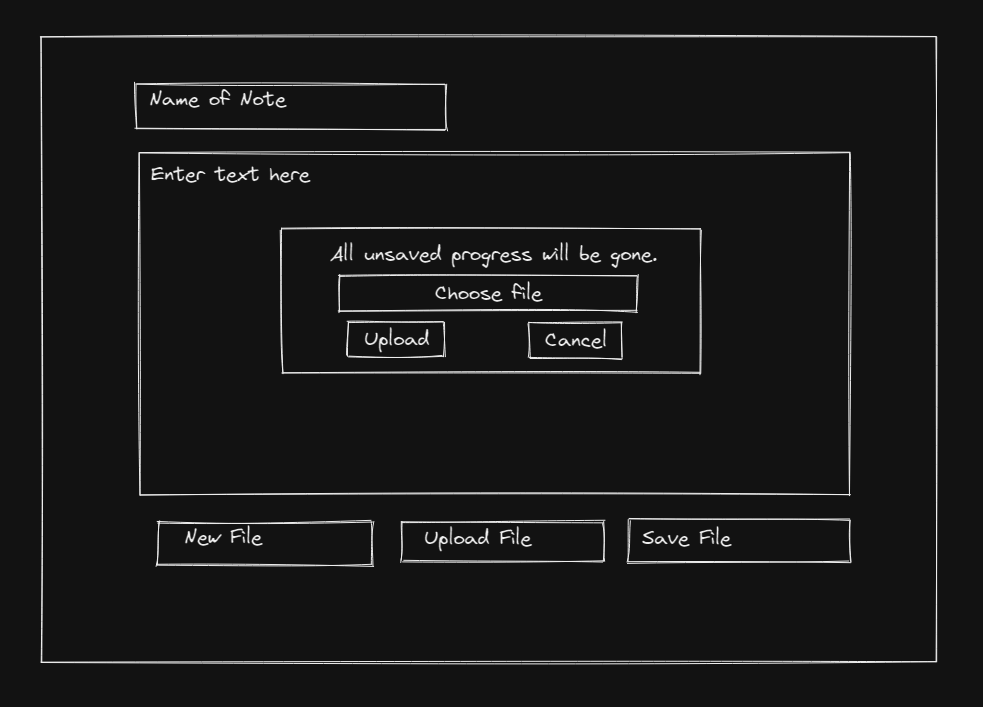


### User Experience

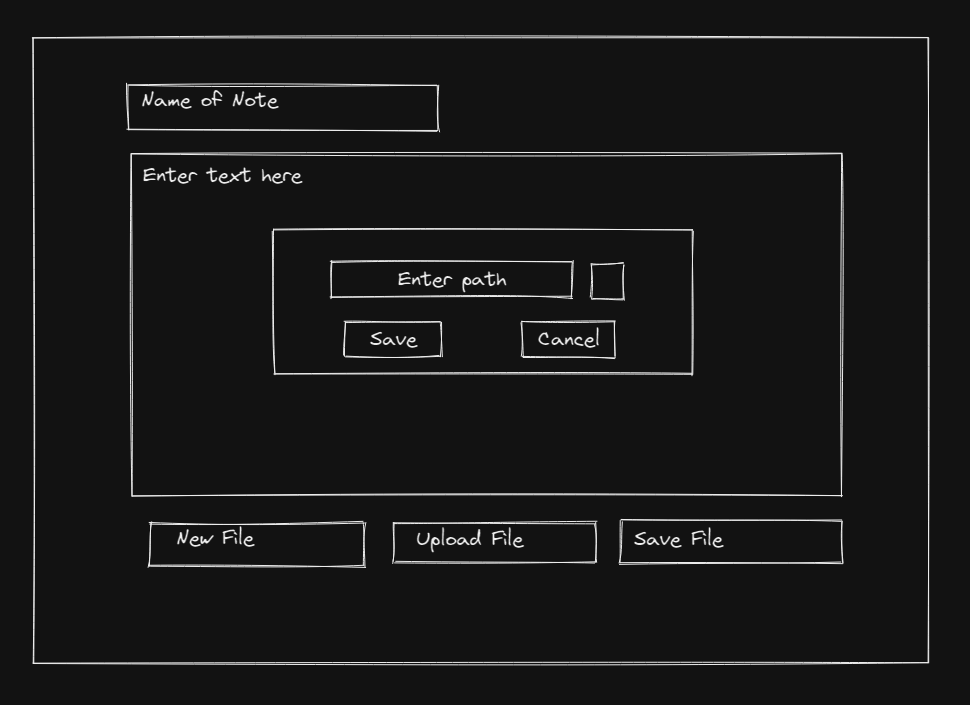
* **New File Button -** User should be able to create a new note with a name and be able to add text to the note. This would act like a reset button that resets the note name and note text. If there is existing data on the screen then it would ask if the user is sure they want to proceed.



* **Upload File Button -** This allows the user to upload a txt file. The name of the txt file will be parsed as the note name and the contents will be displayed in the text box. If there is existing data on the screen then it would ask if the user is sure they want to proceed.



* **Save File Button -** This allows the user to save the contents of the note as a txt file. The name of the note will be used as the file name.



### Developer Considerations

* Application should be local first and would be distributed as a Progressive Web App that the user can install locally once fetched from the remote server.
* Users should be able to edit even during internet blackouts, as long as the PWA has been downloaded from the server and not be server dependent.
* Utilize orchestration based testing tooling such as puppeteer in combination with traditional testing approaches.
* Current version ideally would not require many, if any, third party NPM packages though consideration should be given to use a markdown parser in the future to add support for .md files and avoid implementing a parser from scratch.

## Future Wishlist

* Support for simple cloud storage system optionally if the user wishes to do so such as AWS S3
* Support for more file formats such as markdown and a better editing experience for the user
* Share button to post the contents of the notes to the social media of the user’s choice
* Client side encryption for the user’s peace of mind as even a cloud provider or administrator would not be able to read the file even if they have access to the raw data.