Requirements Analysis

1) User can create a box , or get a box
2) One box can have multiple elements. Elements can be of many types (text, code, image, document) .
3) Box can have a password, expiration time (burst time) , name , description.
4) Creator of the box will get a box-code , and access-token for update.
5) Creator can update the elements of a box using access-token.
6) User can get a box using box-code.

User flow

Create flow

->(optional) add metadata (password, expiry time, name desc)
-> click on “add element” button
-> then select the type (image, text, code,..)
-> fill the form according to element type
-> after adding all the elements
-> click on “save” button
-> gets the “box-code” and “access-token”
-> save the metadata including “box-code” and “access-token” to local storage
-> now creator can share the “box-code” to the concerned user

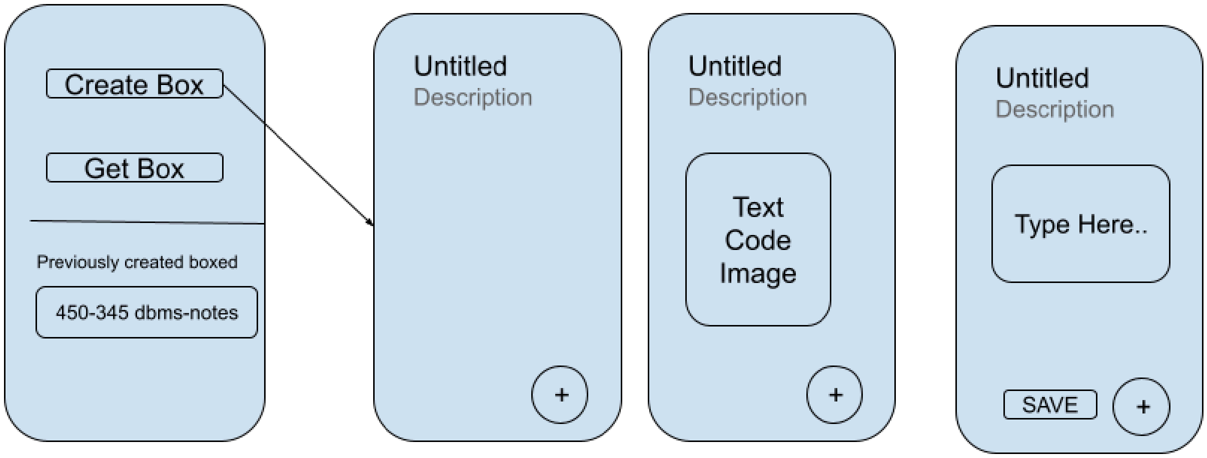
Update Flow

->load the previously created box from local storage
-> select the previously created box from list
-> update any element or metadata
-> click “save”
-> done

Get Flow

-> take the “box-code”
-> click on “get box”
-> put the code
-> (optional) enter password if applicable
-> done

Rough UI/UX design



DB Schema

BoxSchema

_id	ObjectID	Primary key
data	Array < Message >	data
password ?	undefined | string	If undefined that means no password
code	string (6 digit)	
expireAt	date	Time until box is accessable
createdAt	date	Timestamp at creation
updatedAt	date	Timestamp of last modification
name ?	string	Name of the box
desc ?	string	Description of box
owner ?	String	Name of owner / creator

MessageObject

type	TEXT | IMG | PDF | DOC	

MessageObject : type = “TEXT”

type	TEXT	
text	string	

MessageObject : type = “IMG”

type	IMG	
url	string	Image url
Text ?	string	caption

API Design

- create-box : create a box with given content
    - Request
        - data
        - expireAt
        - password ?
        - name ?
        - desc ?
        - owner ?
    - SUCCESS Response
        - status : “SUCCESS”
        - code
        - Token
    - ERROR Response
        - status : “ERROR”
        - error: “INVALID_DATA” | “MISSING_FIELD” | “INVALID_FIELD”
        - message

- ➔ update-box : update the content of given box-id (must be owner)
  - ⇒ Request
    - data
    - token
    - expireAt
    - name ?
    - desc ?
    - owner ?
  - ⇒ SUCCESS Response
    - status : “SUCCESS”
  - ⇒ ERROR Response
    - status : “ERROR”
    - error: “INVALID_DATA” | “MISSING_FIELD” | “INVALID_FIELD” | “INVALID_TOKEN”
    - message

- ➔ get-box: get the box with box-code
  - ⇒ Request
    - code
    - password ?
  - ⇒ SUCCESS Response
    - status : “SUCCESS”
    - data
    - name
    - desc
    - owner
    - expireAt
  - ⇒ ERROR Response
    - status : “ERROR”
    - error: “INVALID_CODE” | “PASSWORD_REQUIRED” | “INCORRECT_PASSWORD”
    - message