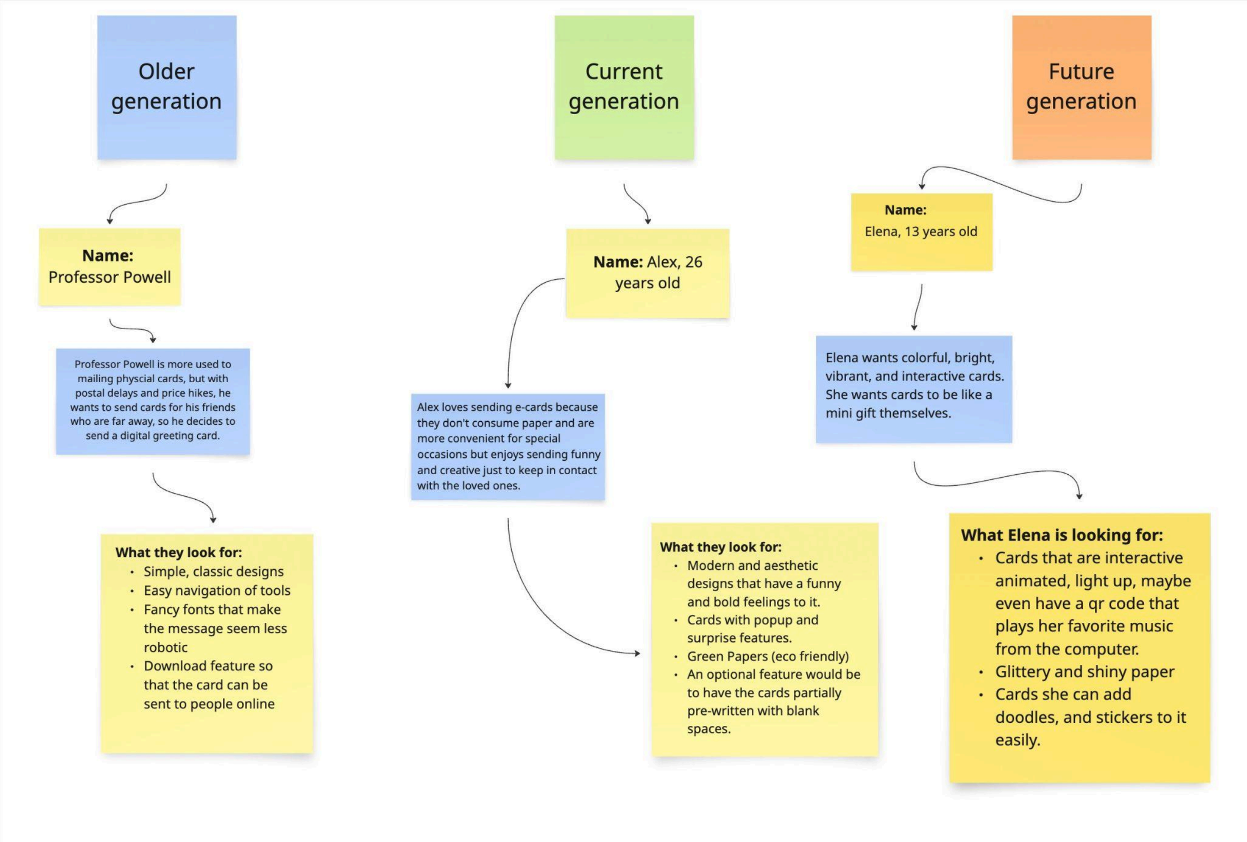
Card Project Proposal

Concept: Building custom greeting cards!
Title (subject to change): Card Create
Purpose: Letting friends and family transmit care through time and space with easily accessible, personal customization of digital greeting cards!
Minimum Viable Product (MVP):
Users can create and decorate their own 5” x 7” dimension (standard dimension) greeting card, uploading a photo or using plain text to decorate the cover, and typing a personalized message on the inside.

- The created card will be saved to local storage until deleted by user
- When they are done, they can choose to download the card

User Personas



Older generation

Name: Professor Powell

Professor Powell is more used to mailing physcial cards, but with postal delays and price hikes, he wants to send cards for his friends who are far away, so he decides to send a digital greeting card.

What they look for:
- Simple, classic designs
- Easy navigation of tools
- Fancy fonts that make the message seem less robotic
- Download feature so that the card can be sent to people online

Current generation

Name: Alex, 26 years old

Alex loves sending e-cards because they don't consume paper and are more convenient for special occasions but enjoys sending funny and creative just to keep in contact with the loved ones.

What they look for:
- Modern and aesthetic designs that have a funny and bold feelings to it.
- Cards with popup and surprise features.
- Green Papers (eco friendly)
- An optional feature would be to have the cards partially pre-written with blank spaces.

Future generation

Name: Elena, 13 years old

Elena wants colorful, bright, vibrant, and interactive cards. She wants cards to be like a mini gift themselves.

What Elena is looking for:
- Cards that are interactive animated, light up, maybe even have a qr code that plays her favorite music from the computer.
- Glittery and shiny paper
- Cards she can add doodles, and stickers to it easily.

Potential Feature List (organized by priority):

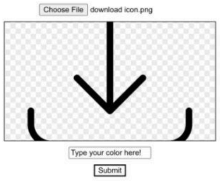
Highlighted items are required for MVP

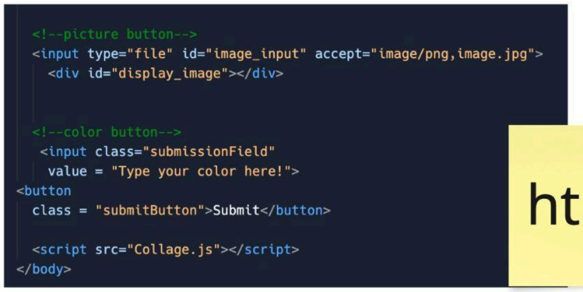
1. Buttons to create and delete a greeting card

2. Add text anywhere on a set space on the page
    a. Can we move text around?
3. Upload a photo into a set space on the page
4. Download finished card
5. Gallery display of cards built by the user for easy access
6. Option to customize fonts
7. Option to change the orientation of the card (sandwich or hot dog style)
8. Option to add stickers to card
9. Pre-built templates for users to edit/ start their card designs from

Feature Testing

upload image functionality



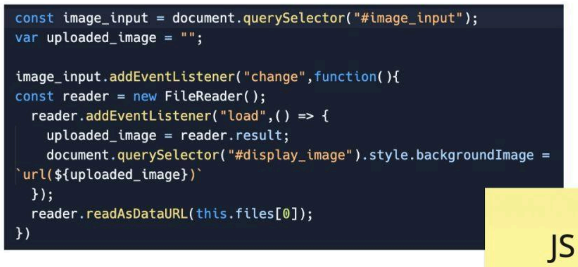


```html
  <!--picture button-->
  <input type="file" id="image_input" accept="image/png,image.jpg">
    <div id="display_image"></div>


  <!--color button-->
   <input class="submissionField"
    value = "Type your color here!">
<button
  class = "submitButton">Submit</button>

  <script src="Collage.js"></script>
</body>
```

html

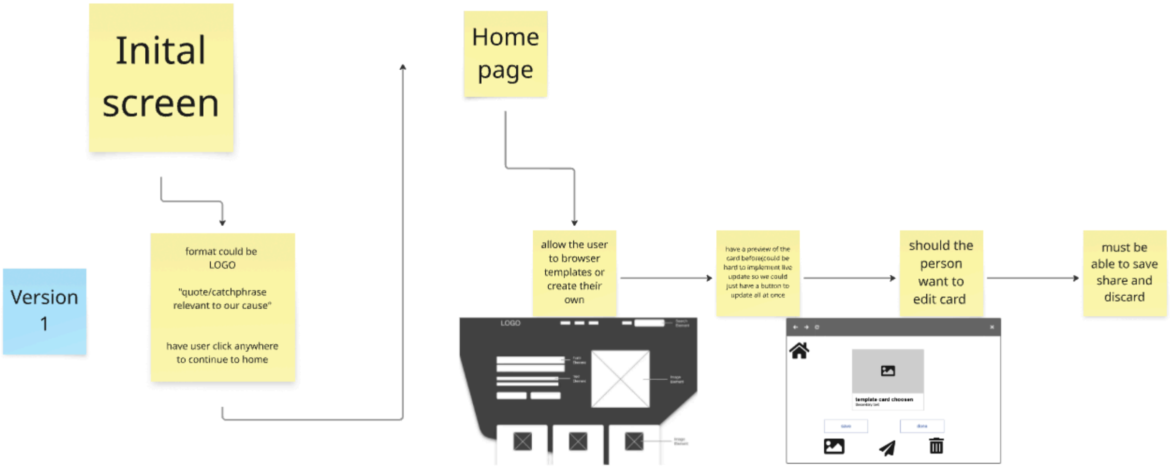


```javascript
const image_input = document.querySelector("#image_input");
var uploaded_image = "";

image_input.addEventListener("change",function(){
const reader = new FileReader();
  reader.addEventListener("load",() => {
    uploaded_image = reader.result;
    document.querySelector("#display_image").style.backgroundImage =
`url(${uploaded_image})`
  });
  reader.readAsDataURL(this.files[0]);
})
```

JS handling

Sketches



Inital screen

Version 1

format could be LOGO

"quote/catchphrase relevant to our cause"

have user click anywhere to continue to home

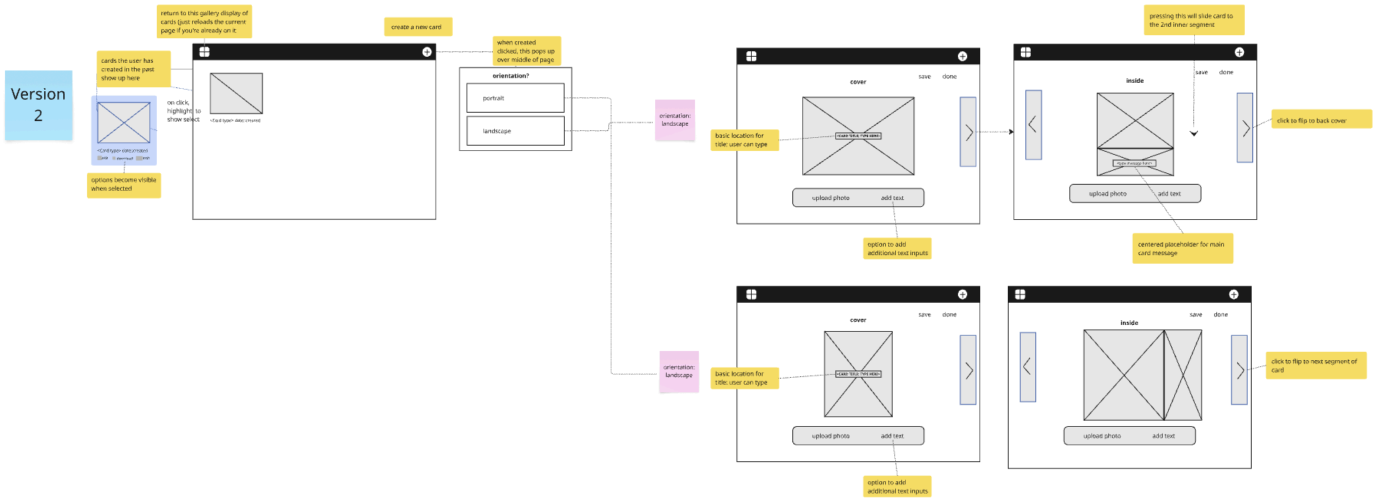
Home page

allow the user to browser templates or create their own

have a preview of the card before(could be hard to implement live update so we could just have a button to update all at once

should the person want to edit card

must be able to save share and discard



Version 2

return to this gallery display of cards (just reloads the current page if you're already on it

create a new card

cards the user has created in the past show up here

on click, highlight to show select

options become visible when selected

when created clicked, this pops up over middle of page

orientation: landscape

basic location for title: user can type

option to add additional text inputs

pressing this will slide card to the 2nd inner segment

click to flip to back cover

centered placeholder for main card message

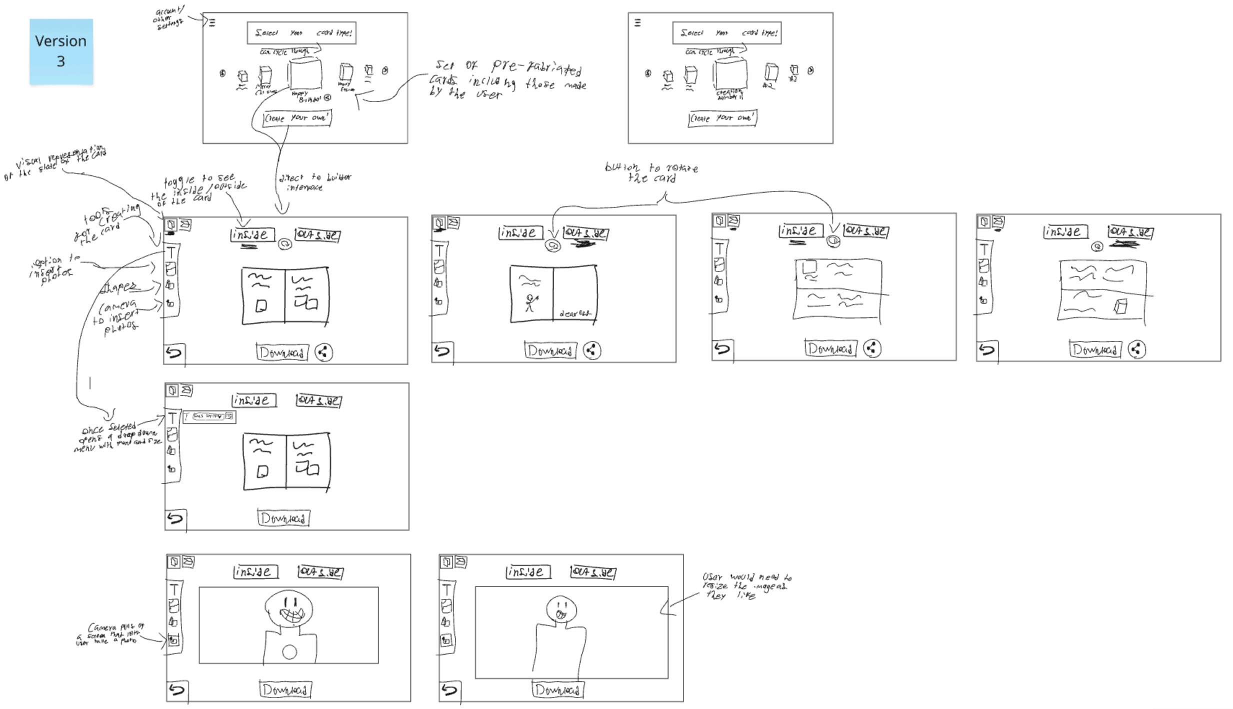
orientation: landscape

basic location for title: user can type

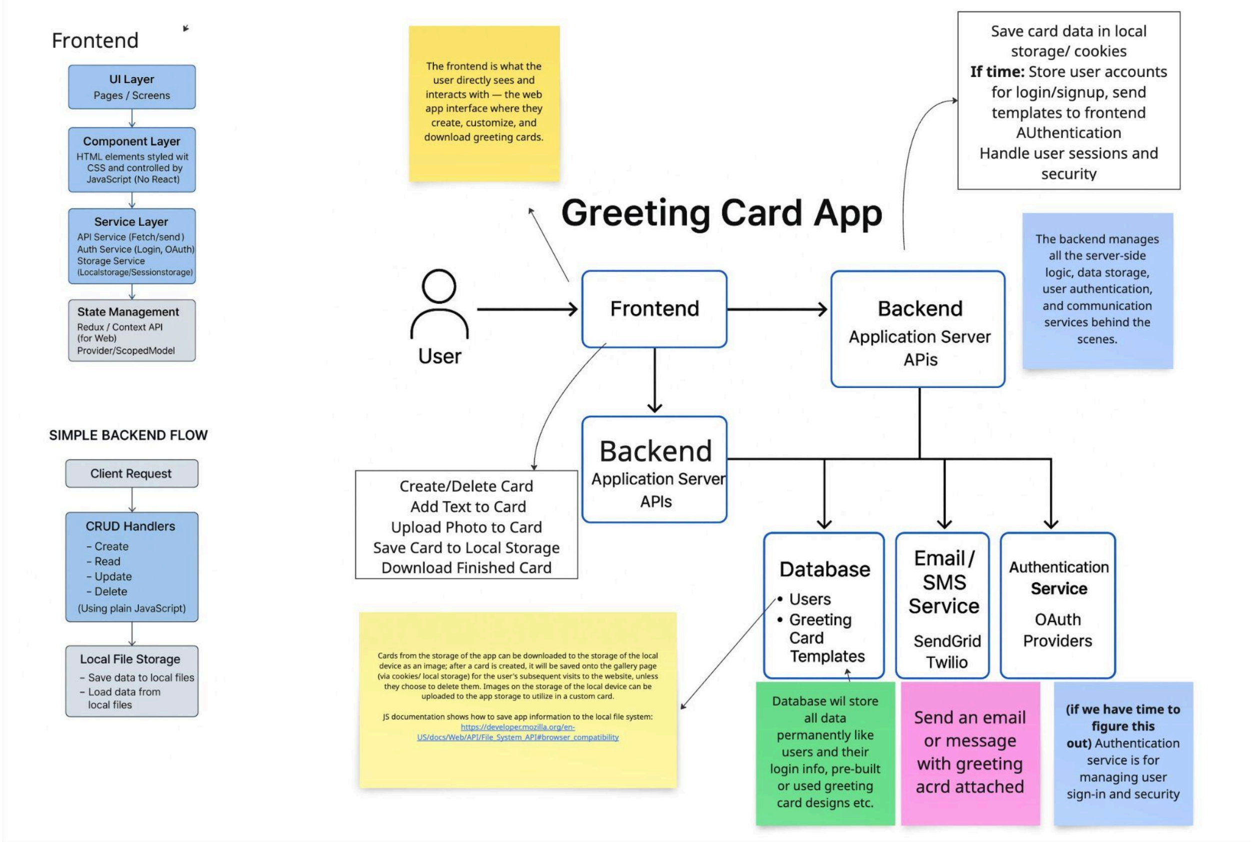
option to add additional text inputs

click to flip to next segment of card

Version 3


account/ other settings
Select your card type!
can cycle through
Merry Christmas
Happy Birthday!
Happy Easter
creation number 1
#2
#3
Create your own'
Set of pre-fabriated cards including those made by the user
visual representation of the state of the card
toggle to see the inside / outside of the card
direct to builder interface
tools creating for the card
option to insert photos
shapes
camera to insert photos
once selected opens a drop down menu with font and size
button to rotate the card
inside
outside
dearest
Download
Camera pulls up a screen that lets user take a photo
User would need to resize the image as they like

System Design



Frontend

UI Layer
Pages / Screens

Component Layer
HTML elements styled wit CSS and controlled by JavaScript (No React)

Service Layer
API Service (Fetch/send)
Auth Service (Login, OAuth)
Storage Service (Localstorage/Sessionstorage)

State Management
Redux / Context API (for Web)
Provider/ScopedModel

SIMPLE BACKEND FLOW

Client Request

CRUD Handlers
- Create
- Read
- Update
- Delete

(Using plain JavaScript)

Local File Storage
- Save data to local files
- Load data from local files

The frontend is what the user directly sees and interacts with — the web app interface where they create, customize, and download greeting cards.

Greeting Card App

User

Frontend

Backend
Application Server
APIs

Save card data in local storage/ cookies
If time: Store user accounts for login/signup, send templates to frontend
AUthentication
Handle user sessions and security

The backend manages all the server-side logic, data storage, user authentication, and communication services behind the scenes.

Backend
Application Server
APIs

Create/Delete Card
Add Text to Card
Upload Photo to Card
Save Card to Local Storage
Download Finished Card

Database
- Users
- Greeting Card Templates

Email / SMS Service
SendGrid
Twilio

Authentication **Service**
OAuth Providers

Cards from the storage of the app can be downloaded to the storage of the local device as an image; after a card is created, it will be saved onto the gallery page (via cookies/ local storage) for the user's subsequent visits to the website, unless they choose to delete them. Images on the storage of the local device can be uploaded to the app storage to utilize in a custom card.

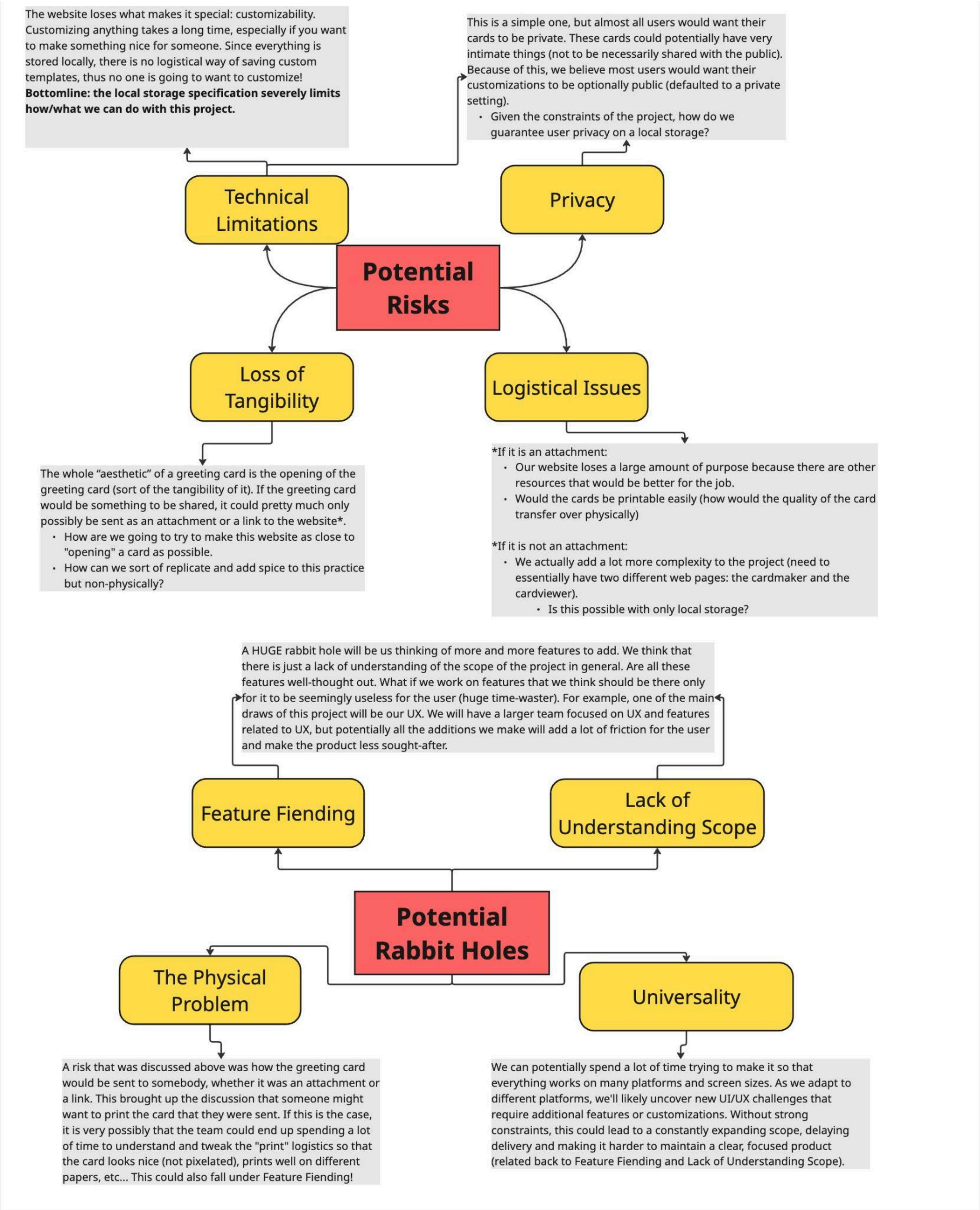
JS documentation shows how to save app information to the local file system: https://developer.mozilla.org/en-US/docs/Web/API/File_System_API#browser_compatibility

Database wil store all data permanently like users and their login info, pre-built or used greeting card designs etc.

Send an email or message with greeting acrd attached

(if we have time to figure this out) Authentication service is for managing user sign-in and security

Risks & Rabbit Holes



The website loses what makes it special: customizability. Customizing anything takes a long time, especially if you want to make something nice for someone. Since everything is stored locally, there is no logistical way of saving custom templates, thus no one is going to want to customize! **Bottomline: the local storage specification severely limits how/what we can do with this project.**

This is a simple one, but almost all users would want their cards to be private. These cards could potentially have very intimate things (not to be necessarily shared with the public). Because of this, we believe most users would want their customizations to be optionally public (defaulted to a private setting).

- Given the constraints of the project, how do we guarantee user privacy on a local storage?

The whole "aesthetic" of a greeting card is the opening of the greeting card (sort of the tangibility of it). If the greeting card would be something to be shared, it could pretty much only possibly be sent as an attachment or a link to the website*.

- How are we going to try to make this website as close to "opening" a card as possible.
- How can we sort of replicate and add spice to this practice but non-physically?

*If it is an attachment:

- Our website loses a large amount of purpose because there are other resources that would be better for the job.
- Would the cards be printable easily (how would the quality of the card transfer over physically)

*If it is not an attachment:

- We actually add a lot more complexity to the project (need to essentially have two different web pages: the cardmaker and the cardviewer).
  - Is this possible with only local storage?

A HUGE rabbit hole will be us thinking of more and more features to add. We think that there is just a lack of understanding of the scope of the project in general. Are all these features well-thought out. What if we work on features that we think should be there only for it to be seemingly useless for the user (huge time-waster). For example, one of the main draws of this project will be our UX. We will have a larger team focused on UX and features related to UX, but potentially all the additions we make will add a lot of friction for the user and make the product less sought-after.

A risk that was discussed above was how the greeting card would be sent to somebody, whether it was an attachment or a link. This brought up the discussion that someone might want to print the card that they were sent. If this is the case, it is very possibly that the team could end up spending a lot of time to understand and tweak the "print" logistics so that the card looks nice (not pixelated), prints well on different papers, etc... This could also fall under Feature Fiending!

We can potentially spend a lot of time trying to make it so that everything works on many platforms and screen sizes. As we adapt to different platforms, we'll likely uncover new UI/UX challenges that require additional features or customizations. Without strong constraints, this could lead to a constantly expanding scope, delaying delivery and making it harder to maintain a clear, focused product (related back to Feature Fiending and Lack of Understanding Scope).

Possible Expansions

- After the MVP is complete and tested with users, we can make our way down the features list
- Improve app security and retention of designs

(Rough) Workflow => 5 weeks to MVP (or beyond) completion**:**
**this is subject to change based on assignment structure laid out by teaching team*
-> weekly meetings + drop in work hours during Wednesday lab sessions
-> full team check ins Tuesdays and Thursdays: sprint updates to leads on Sundays + more as needed
Weeks 1 & 2: interface design, user research and feature feasibility testing
Week 3: implementation
Week 4-5: testing and refinement