

CS 422 Workbook - Group 13

GR 1:

Group Members:

- Shantanu Agara Dwarakanath
- Jon-Michael Hoang
- Ryan Nishimoto
- Andrew Wentzel
- Edward Nava

Links:

- Github: <https://github.com/tehwentzel/DocDoc>

Problem Statement:

Briefly state the problem(s) that your project will seek to solve. Take the user's point of view.

Consider what the user's goals are, and what obstacles lie in the way. Do not talk about any solutions here.)

Keeping up with doctors appointments can be difficult for parents with kids. If you're a new parent, or have many kids, it could be especially difficult finding out where to start.

It can be hard finding a pediatrician for your children. If you need an appointment quickly, you want to be able to see pediatricians that are around your area. You want to be able to know what areas the pediatrician specializes in, what form of insurance they take, and when they will be next available for an appointment. You want to be able to quickly give current medical information necessary without having to fill out a lengthy form for each visit. You want to be able to see what other parents thought of a particular pediatrician or appointment experience to help you decide on who is best for you.

Another potential obstacle is keeping up with all the medical requirements for kids. Children have specific medical needs, such as yearly checkups and vaccinations, and medical concerns for children are different than for adults. While most medical apps are designed for personal use, there is little to help parents manage multiple dependents. Potential problems in this area could include scheduling timely visits for multiple kids, accessing records and special medical instructions, and quickly communicating with a pediatrician or poison control for consultation with minor medical issues.

Some obstacles may include not knowing what kind of specific help your child needs or what particular ailment is present; you just have a list of observable symptoms. Being able to keep track of any medications, allergies, or vaccinations may be necessary when scheduling an appointment, and finding a particular treatment. In certain cases, when a dependent is older (usually above 10), medical confidentiality could be a concern.

GR1 Analysis:

- ***Observations and Interviews:***

Interview 1: The user is a working single mother of 3 older children, including 2 children in college, and one who was currently working. She is currently also supporting her single aging father. To get a better understanding of their workflow, she was asked to describe typical tasks and step through their typical processes. The user identified medical and other requirement document management to be one of the biggest issues with her current system(Goal 1), as well

as the ability to share information with other family members when responsibilities were shared (Goal 4).

Most of the regular checkups for the user's kids were done according to the requirements for the school, which dictated that she send in a number of forms at different periods. She would then call their pediatrician and schedule a visit (Goal 5). The remaining steps were done primarily in-person at the doctor's office: the client would obtain a list of the needed forms by looking up the school's website for the requirements, and bring them to the office, or look them up on her phone. She would then ask the doctor specifically for those forms, and get physical copies, or have forms mailed or emailed to her, depending on the doctor used (currently, she could get electronic copies). Scheduling would be done before leaving the office.

Certain complications tended to appear when switching doctors or schools, as she has moved multiple times (Goal 3). The user previously had issues getting proof of previous vaccinations from before she moved, and at one point had to call multiple walgreens in order to get a last minute polio booster shot before school started.

When asked to see if they could get access to the forms needed currently, the user googled the local medical requirements for the local school system, and found that they were different than when her kids were in high school. She then tried to identify the previous pediatricians that she had visited but wasn't able to figure out the contact information for the correct one, as several had very similar names (Goal 2).

Often information on major allergies, medication, or conditions that could be important for treatment or surgery. When interacting with medical staff on behalf of her father and kids, she was often asked if they had allergies. Her grandfather was allergic to penicillin, and a child was allergic to ibuprofen. In these scenarios, she had to confirm this from memory, which left a lot of potential for her to make a very high-cost mistake (Goal 4).

Interview 2: User is a mother of two children, both male, four and half years apart in age. She is a nurse who works at a hospital. Working in the medical field, she is organized in all medical documents for the kids. When scheduling a regular checkup, she would call the doctor and schedule a time that would work best. If it was more of a serious concern, she would call the doctor to see if there is a more immediate time available, or if there were treatments that could be done at home until a checkup could be scheduled (Goal 5).

Immunization records often required for school would always be on-hand and dated; After an immunization or appointment, she would often ask for a paper copy of the records for her own to keep at home (Goal 2). If an immunization was out of date or it was time for another shot, she would have that on record and call for an appointment. If a checkup resulted in results being given a time after the initial appointment, it would usually be over the phone; if a record of the results was wanted, a trip to the doctor would usually be needed (Goal 3). The user mentioned how some doctors do have recently started sending records through the computer and email, but some doctors still stick to paper records and may not utilize the technology for various reasons.

Interview 3: The user is a father of two sons, both no less than two years apart, with one currently in elementary school, and another in preschool. He is a married man, who currently

works as a truck driver making deliveries from one end of the country to another. When he wanted to schedule an appointment with a doctor for his kids, he usually makes a call to the doctor's office, and stated that he would sometimes be put on hold by for quite some time by the receptionist, as they were busy (goal 3). This usually wasn't a major problem, but something to take note of, he said. Other than that, he says the standard procedure for scheduling an appointment is: call the office, wait if necessary, schedule a time that fits best, and be there on time on the day of the visit.

Because his children are young and are enrolled in school, they are required by the school district to always have immunization records up to date and have physical copies of such records for the schools and their nurses (Goal 1). And with that in mind, the school follows the vaccination schedule for Infants and Children given by the Centers for Disease Control and Prevention (CDC)¹. Because of all this, the user does find it stressful to manage paper records and copies, as his job as a truck driver already involves considerable amounts of paperwork. As his sons' doctors and their schools utilize a mixture of computer and paper records, the user mentions as to how he hopes that there is a way to apply technology the could alleviate a considerable amount of his workload.

Interview 4: The user is a father of a 22-month old toddler. Both parents work full-time and the baby is taken care of by a daycare during their work hours. The parents always go to the same pediatrician for all medical needs of the baby. The hospital gives hand-outs of some sort every time they visit, which are often misplaced. However, all the important data related to the baby is digitally stored by the hospital, and thus the parents need not carry any physical records of the baby. The user wishes to switch the doctor and go to a hospital which is much closer to home, but he is unsure how the medical records will be transferred to the new doctor. The user noted that there are significant hassles when it comes to maintaining medical bills and receipts (Goal 3). Although the baby has insurance, sometimes the parents are charged and the user is often unable to keep track of the expenses.

The daycare that the baby goes to makes a note of what the baby eats and other activities throughout the day on a paper form. The user felt that it would be more convenient if they could send it on his or his wife's phone. Also, the baby has a few food allergies, which the daycare staff needs to be aware of. He was unsure about how the daycare maintains this record.

Because the baby is still very young, he frequently catches a cold or fever. Over the counter medicines like cough syrups and fever medications always need to be handy, but it becomes very difficult to keep track of the expiration dates. The user also noted that the parents face difficulty in keeping track of the availability of essential items for the baby like diapers, shampoo, baby wipes, etc.

Interview 5: The user is a mother of 4 children, 2 in college and the other 2 in high school. She works full time and has difficulty keeping up with her kids' health and medical records. When attending a medical checkup with her children, she typically receives some paperwork and would like to be able to save as many relevant documents as possible in an orderly fashion (not

¹<https://www.cdc.gov/vaccines/schedules/easy-to-read/child-easyread.html>

by paper). She also knows that schools, both higher education and K through 12, require vaccination and certain medical records to be presented when enrolling, and would like to be able to keep all her children's records in a single place to make this process easier. She says keeping all the paperwork can get cluttering and that the older paperwork has started to degrade, and would like to be able to easily save the images before they become even more degraded (Goal 2). Since her children can check their own medical records at this point, she would also like a way to give them access to their own medical records in the event they may need it and finds it difficult to do so with the amount of papers she has to go through to find a specific record (Goal 4). She has also discovered that her children tend to get ill around the same time every year and wishes that it would be easier to track those health trends in some manner that can predict when they are likely to get sick again. She says being able to do this would help her adjust her children's diets at home and give her time to prevent her children from spreading their illness easily.

- **User Classes:**

- Primary caretakers (parents) with school-aged children
 - Parenting age (usually 24-40 years old)
 - High-school level education, usually low medical knowledge
 - Normally computer literate
 - Little application experience, may use system a few times a year
 - Typically busy, motivated by ease of use

- **Goals:**

1. Schedule appointments easily.
2. Track school medical/legal records and requirements.
3. Save or update physical or electronic forms in the app
4. Minimize paperwork and unnecessary trips to the doctor..
5. View and send saved forms easily.

GR 2 - Design:

Scenarios:

Scenario 1:

Henry Henderson is a parent of two children, Jannette and Bob. Bob is just starting his first year at a new school, since Henry just moved to Chicago from Idaho, and the school wants him a long list of proof of vaccination forms to send in. Knowing that he needs to find a local pediatrician to take Bob to, he finds pediatricians near him using *docdoc*. After scheduling an appointment for Monday, he logs the date into *docdoc*'s handy calendar feature (Goal 1).

Monday comes, and Bob's checkup goes well. However, at the end, Henry has forgotten what forms he needs from the doctor! Undaunted, he pulls out *docdoc* and quickly looks up what forms Bob needs this year, which he had previously saved (Goal 2). The doctor comes back and hands him a nice stack of papers, which, obviously, annoys Henry greatly. Thankfully, with technology, he quickly scans the images to a pdf and sends it to *docdoc* for safekeeping (Goal 3).

At home, Henry goes to send the forms to the school. *docdoc* has conveniently kept track of what forms he has submitted, but Henry panics when he notices that he has forgotten to send a scan of Bob's polio vaccination! Suddenly, he realizes that Bob got that vaccination last year, and it's still conveniently stored in the app (Goal 4), so Henry sends it on its way (Goal 5).

Scenario 2 (Stretch Goals):

Henry is on a work trip when his daughter, Jannette, gets sick. In his place, Hilda Henderson, his grandmother who is watching Jannette, takes her to a physician. The physician finds out that Jannette has an ear infection, and prescribes her an antibiotic. Before deciding which, the physician asks Hilda if Janette has any allergies. Hilda isn't sure. She doesn't think so, but opens up *docdoc*, and looks at the information for Hilda there. Sure enough, Henry left a note that Janette is allergic to penicillin! Hilda informs the doctor, and a more appropriate antibiotic is given instead (Goal 6).

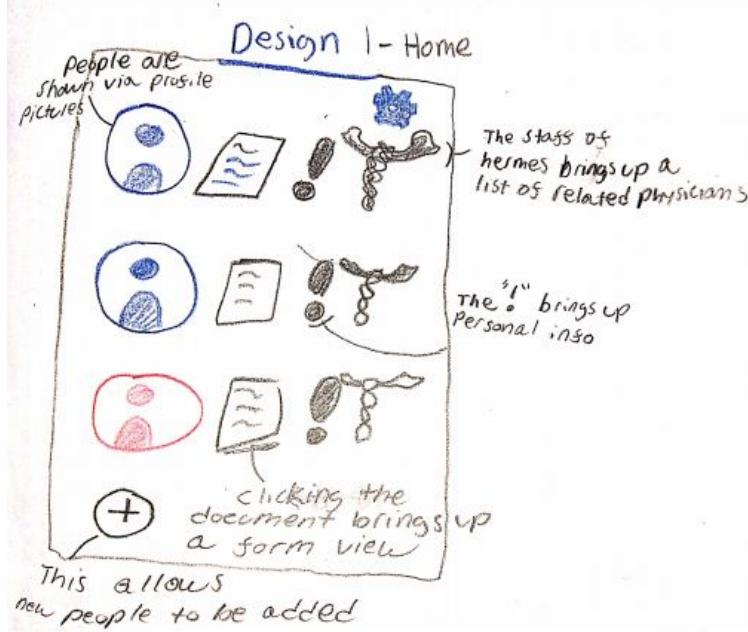
Later, Hilda remembers to log Jannette's ear infection in the health-tracker component of *docdoc*, for future reference. Looking at Jannette's history, she realizes that they all happen in the summer after family vacations. Perhaps, she thinks, it is due to water getting in Janette's ears. She isn't sure if that makes sense, so she quickly messages her pediatrician to ask if that's a possibility (Goal 7).

Individual Design Sketches:

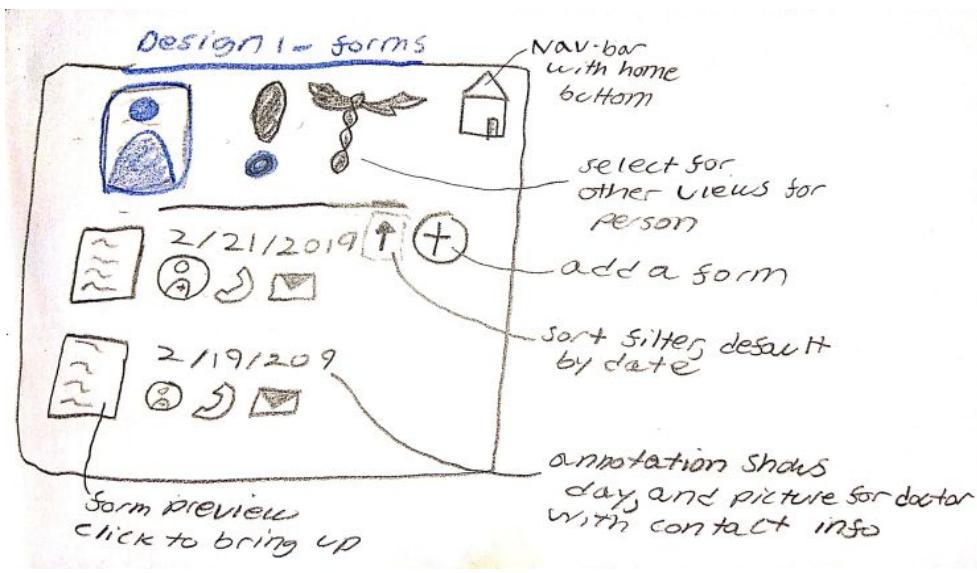
Andrew's Designs:

Design 1:

This design is intended to be usable for illiterate or non-english speaking users. The interface uses no words, with the exception of certain parts that can be shown to medical professionals for them to use, such as allergies or names.

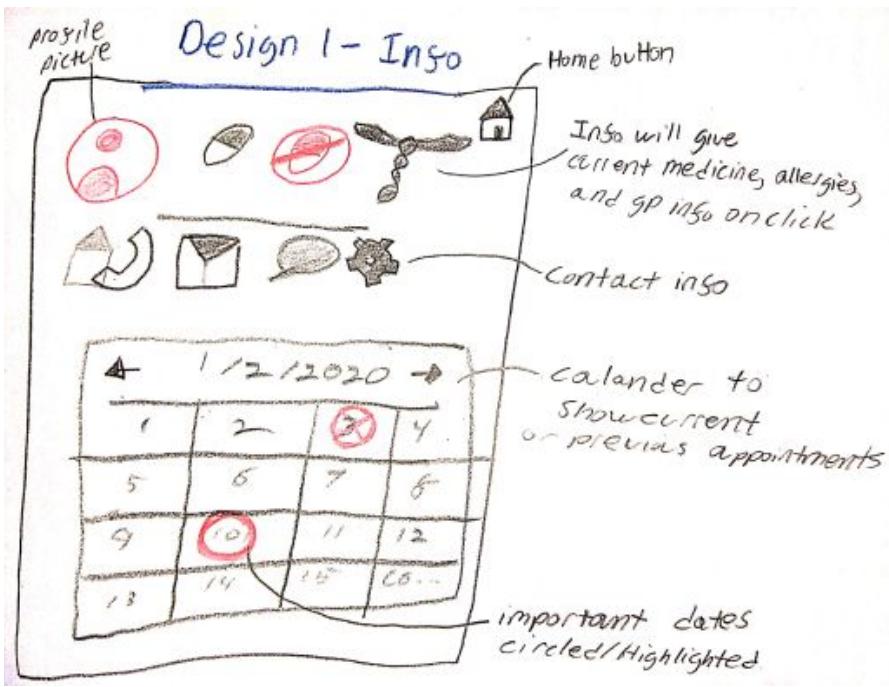


The home page focuses on a 'profile' for each user. People are designated with a profile picture representing them. Three buttons are aligned with profile pictures representing information, forms, and doctors, which brings up patient-related views. A plus button allows for the user to add another patient.



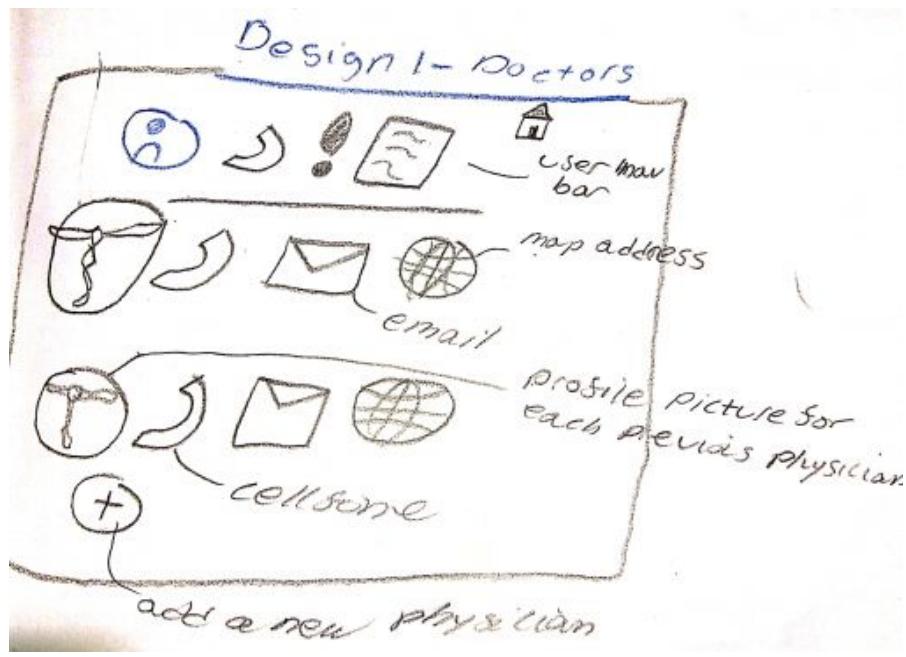
The 'form' view for a patient. A navbar that is similar to other views at the top allows you to switch to other views, or return to the home page. Each form is presented with a preview image, a date, and buttons that show a picture for the relevant doctor, and links to contact that doctor.

An arrow filter allows you to switch between ascending and descending sorting, chronologically. A similar 'add' button allows you to add a new form.



Person-level 'info' view. Buttons on the top show the nav bar. Buttons below allow you to easily access contact information for the person. The cog allows you to change the status of the person or edit info. The pill symbols will bring up a list of medications and allergies to show to doctors.

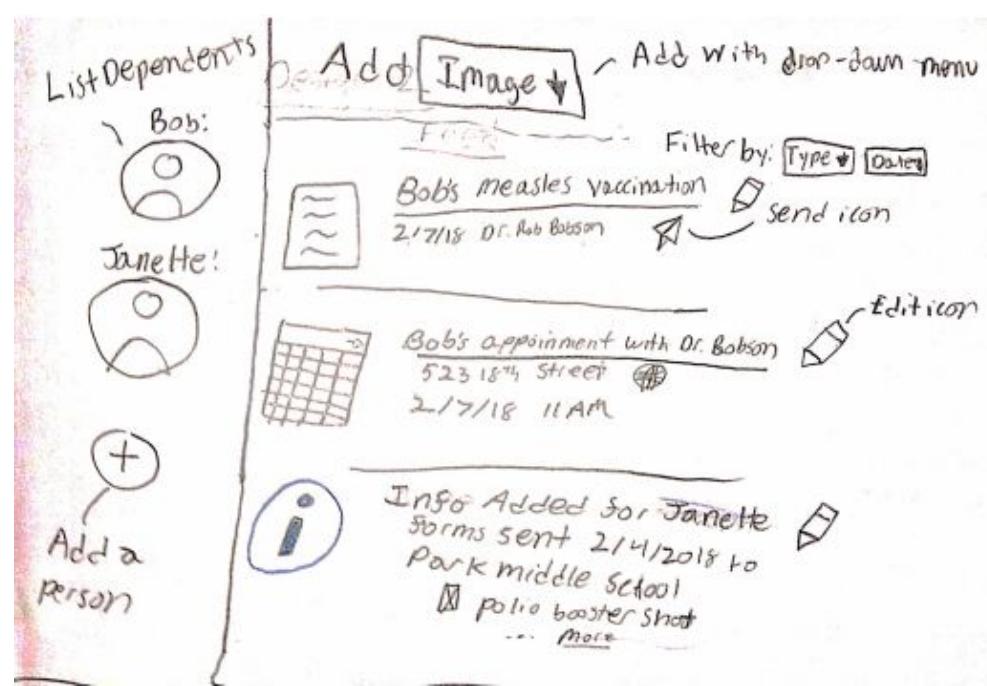
The calendar at the bottom allows you to add and see important dates.



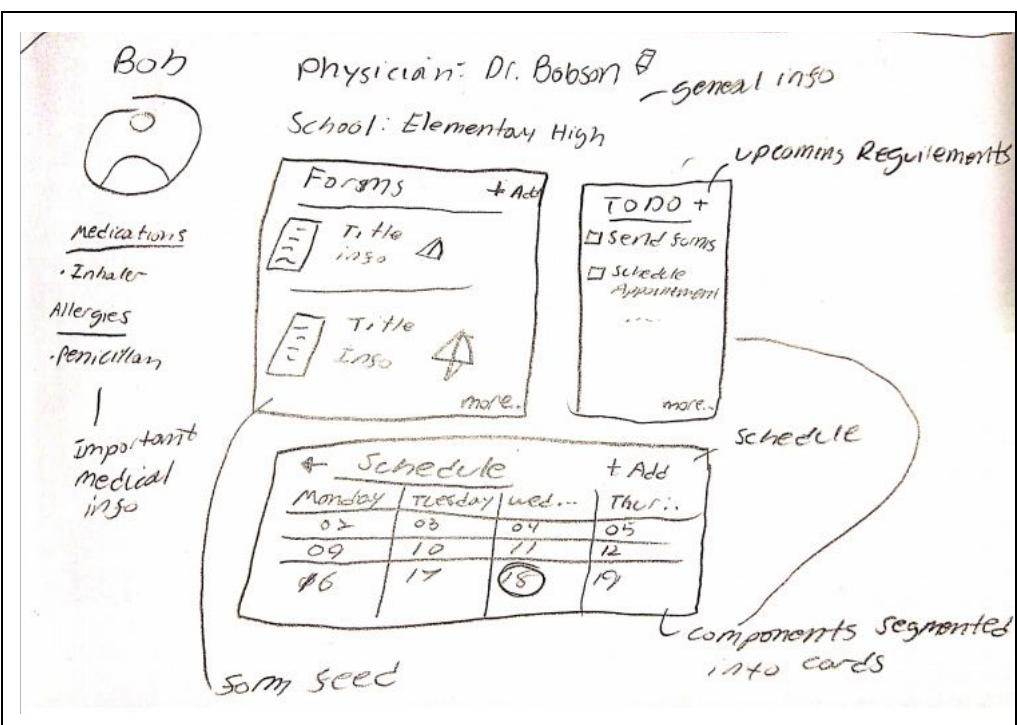
Doctor-level view. Essentially shows a list of positions for each physician that the person is associated with, with buttons to contact each doctors, bring up a google-maps (or similar) page showing their address, and add more physicians.

Design 2:

This design takes an attempt at a more condensed view. Every piece of information is treated as an event, with a recent/upcoming feed on the home page, and profiles for each person listed on the side. It is designed for high-frequency users.



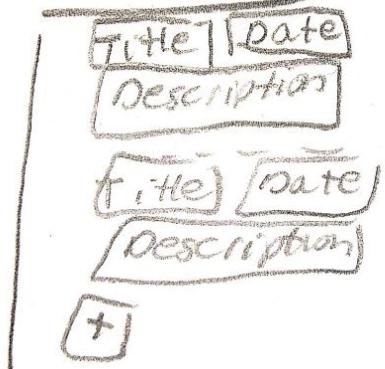
Home page. The feed lists events - visits, form uploads, deadlines, and notes - in chronological order. A filter option lets you sort for specific items, such as only forms. A button lets you add an event to the front. Profiles for dependents are shown on the side.



Dependent “profile” page. Lists a picture, relevant emergency information (physician, school, medication, and allergies). Events are divided into forms, notes, and actual events, such as appointments. These are given their own sections and can be viewed/added separately. The most recent events are shown, with buttons to show more, which will bring up a filtered feed similar to the first view.

View for adding an event. Designed to work for all cases, so metadata and file uploads (for forms) are shown.

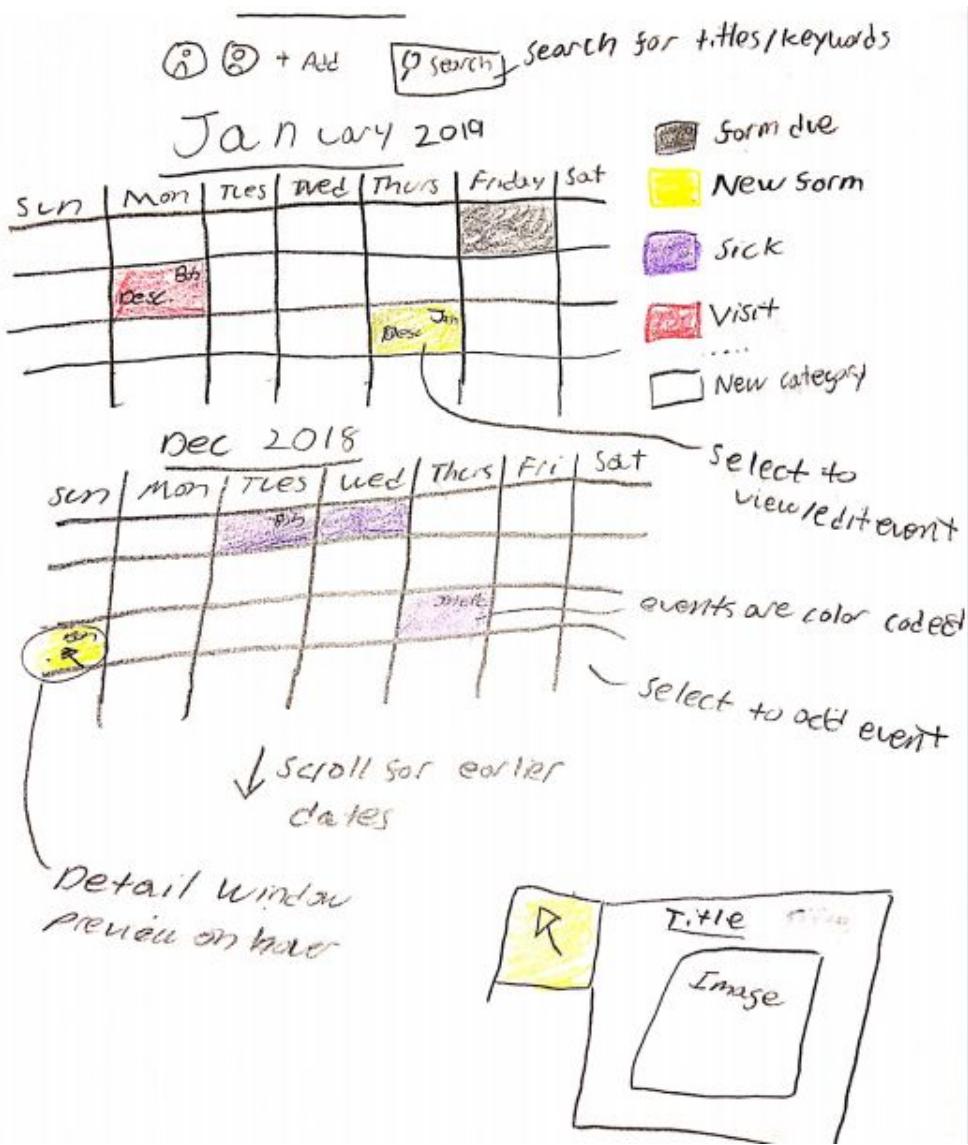
New Note



View for adding a new note on a person's profile page.

Design 3:

This design is intended to be very learnable and minimalist. The whole app is 1-2 pages.



The home page is a large calendar feed color-coded to specific events that you can define. You can scroll to see past or future events (if there are any). Clicking on a square will bring up any event, or allow you to add one. Color coding can be edited on the legend on the right. Hovering over a data brings up a tooltip-like window that lets you see the title and a picture (if there is one) and a preview of the description. People can be associated with each event. The top allows you to click a person to show only their event. The search bar is also there for searching events keywords

The diagram shows a form for adding a new event:

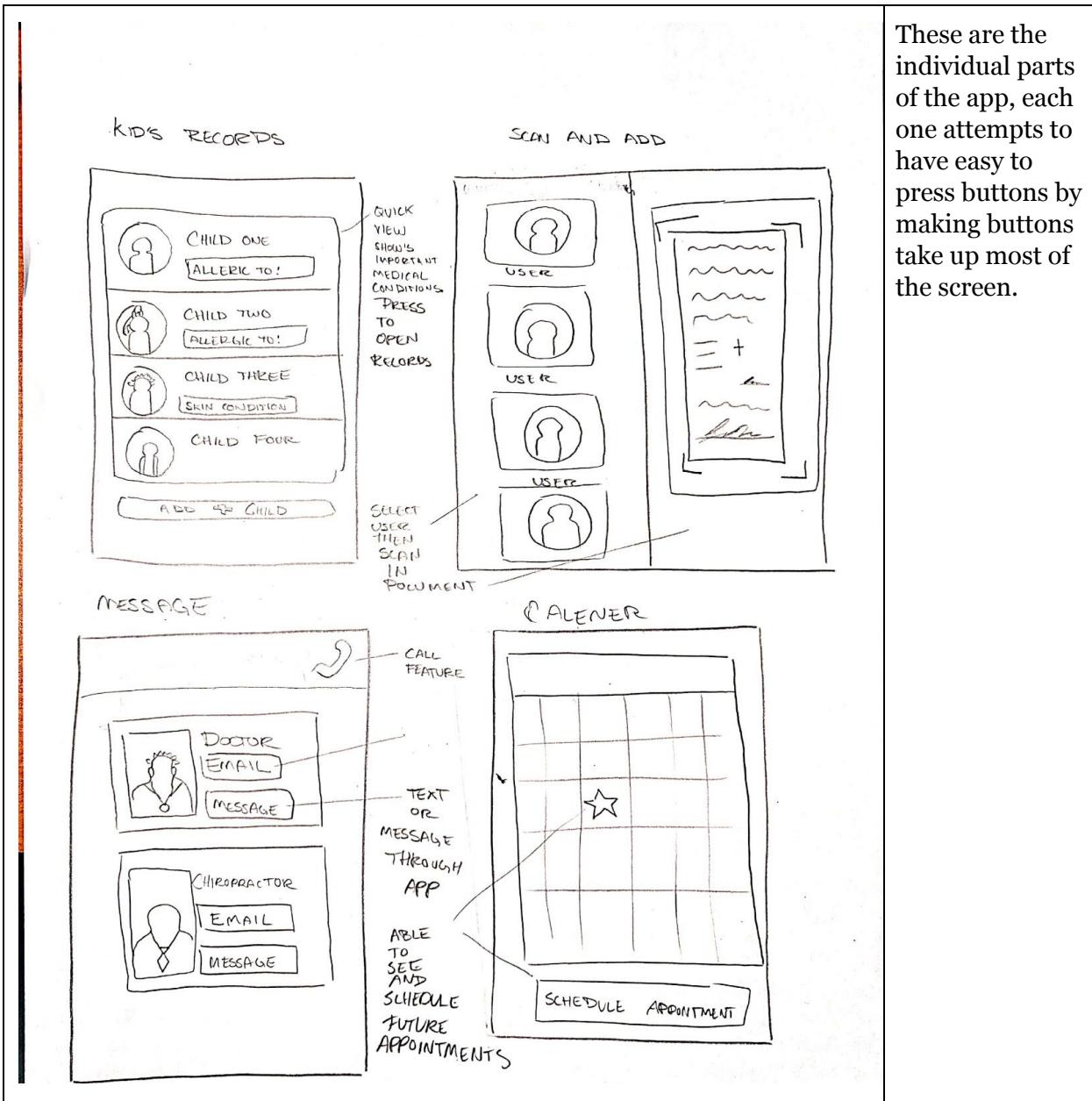
- Form Fields:** "Title", "Description", "Upload file", and "Type" (with a dropdown menu showing "New form", "Sick", and "Visit").
- Buttons:** "Send To:" and a small envelope icon.
- Annotations:** "New event" is written above the title field, and "description of event" is written above the description field.

Sample view for bringing up an adding a new event.

Edward's Designs:

Design 1

The home page allows the user to access all necessary scenes to view information they request. Here they can also view health suggestions and warnings their doctor has suggested.



Design 2

DocDoc Design 2

The design features two main screens:

- User Details Screen:** This screen displays user information such as USER NAME, ALLERGIES, and MEDICATION. It also includes sections for EMERGENCY CONTACT and PRIMARY DOCTOR. Below these are profiles for CHILD 1 and CHILD 2. At the bottom, there is an ICON DOCK containing icons for a heart, eye, envelope, and grid. An EDIT BUTTON is located near the child profiles.
- Scan and Add Documents Screen:** This screen shows a preview of a document with handwritten text and a camera icon at the bottom. It also has an ICON DOCK at the bottom.

① USER DETAILS

HERE THE USER IS PRESENTED WITH SELECTING WHICH CHILD'S PROFILE THEY'D LIKE TO VIEW AS WELL AS A QUICK OVERVIEW OF THEIR PERSONAL INFO

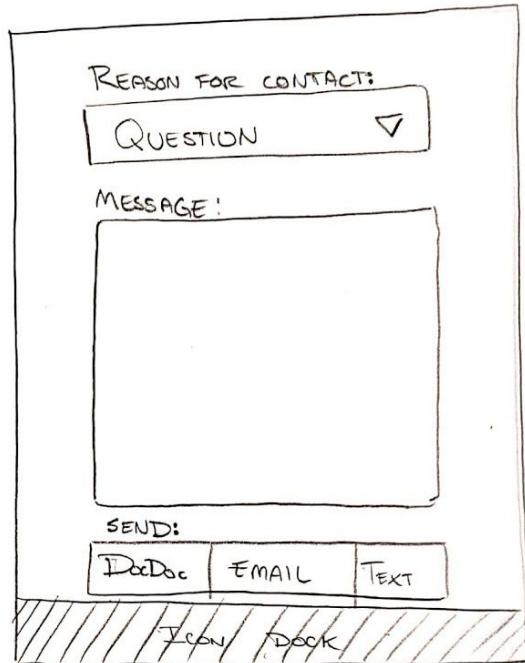
② SCAN AND ADD DOCUMENTS

THIS TAB ALLOW USERS TO QUICKLY TAKE A PICTURE AND THEN ADD IT TO A USER'S FILE REPOSITORY

SEARCH (FILE VIEW PREVIEW)

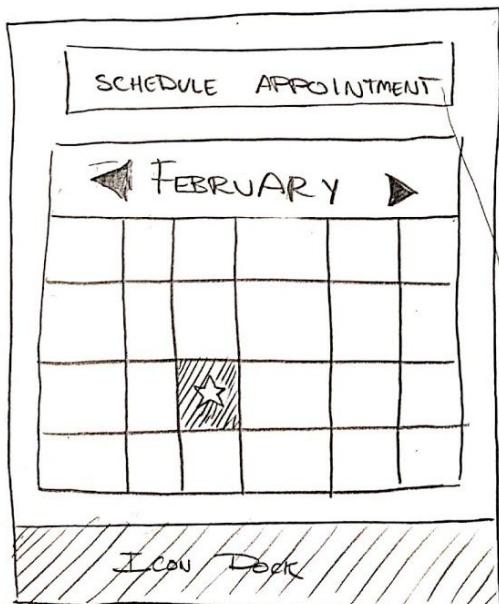
③ CONTACT DOCTOR

IN THIS TAB THE USER
IS ABLE TO WRITE
AN EMAIL/MESSAGE
TO THE DOCTOR.
THEY SET AS THEIR
PRIMARY Doctor



④ APPOINTMENT CALENDAR

HERE THE USER
CAN VIEW A CALENDAR
WITH SCHEDULED APPOINTMENTS
WHICH SEND PUSH NOTIFICATIONS
REMINDERS FOR UPCOMING
APPOINTMENTS.
ALSO USED TO TRACK 'ILLNESS SEASON'
THIS BUTTON SENDS THE USER
TO THE MESSAGE TAB
WITH 'SCHEDULE APPOINTMENT'
AS THEIR REASON FOR CONTACT.



Design 3

DocDoc DESIGN 3

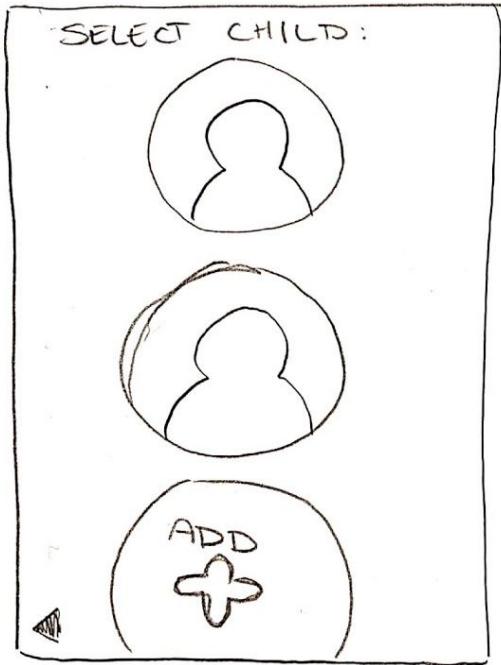
The wireframe illustrates a mobile application interface for 'DocDoc'. It features three main screens:

- USER INFO Screen:** Shows a placeholder profile picture and a 'QUICK INFO' box containing:
 - ALLERGIES: PEANUT BUTTER
 - CONDITIONS: LUPUS
- NAVIGATION Screen:** Shows a navigation bar with 'USER INFO' at the top, followed by icons for 'CONTACT DOCTOR' (5), 'ADD FILE' (4), 'CHILDREN' (1), and 'CALENDAR' (2).
- FILES Screen:** Shows a placeholder profile picture and a 'FILES' section with two preview boxes labeled 'PREVIEW'.

Annotations provide additional context:

- A callout for the first screen notes: "① THIS IS THE MAIN SCREEN THAT SHOW WHEN YOU OPEN UP THE APP. THE UI PRESENTS SCROLLING, BUBBLES WITH QUICK ACTION/VIEW FOR EVERY BUBBLE".
- A callout for the second screen notes: "EVERY USER HAS THIS INFO SCREEN WHEN THEIR PROFILE IS NAVIGATED TO. HERE THEY CAN VIEW FILES THEY ADD."

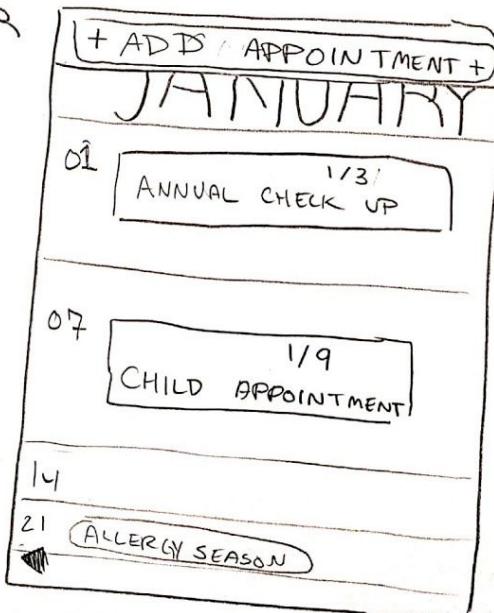
② CHILD'S PROFILE SELECTION



THIS IS A SIMPLE
SELECTION AND ADD/EDIT
SCREEN

③ CALENDAR

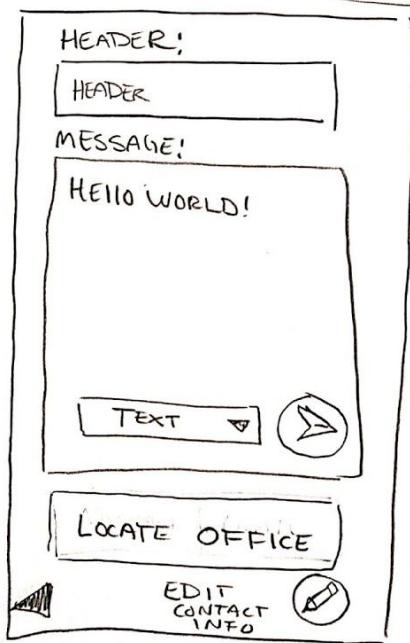
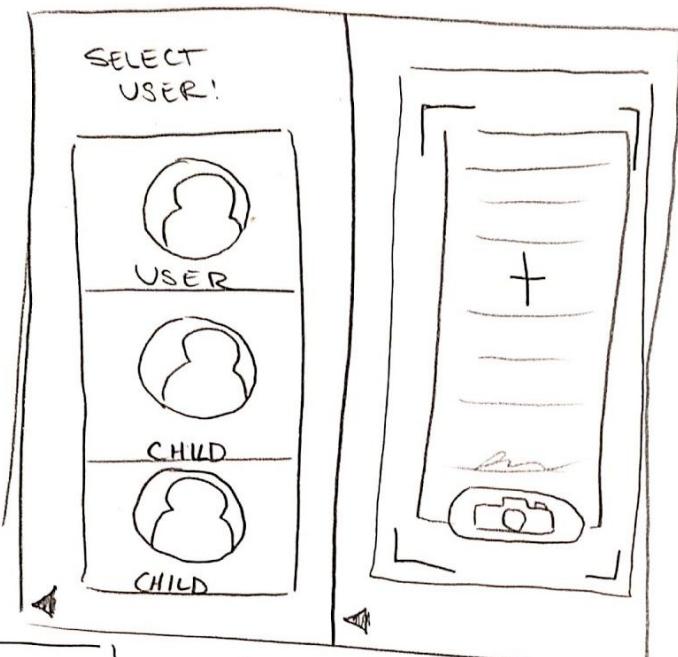
SEND PUSH
NOTIFICATIONS
WHEN AN
APPOINTMENT
IS THE
SAME
WEEK.



DISPLAYS
USERS
FUTURE
APPOINTMENTS.
ALLOWS
FOR THE
USER TO
SET APPOINTMENT
REMINDERS.

④ ADD FILE

IN THIS SCREEN THE USER CAN ADD FILES TO A USER'S SELECTED FILE BY OPENING A CAMERA



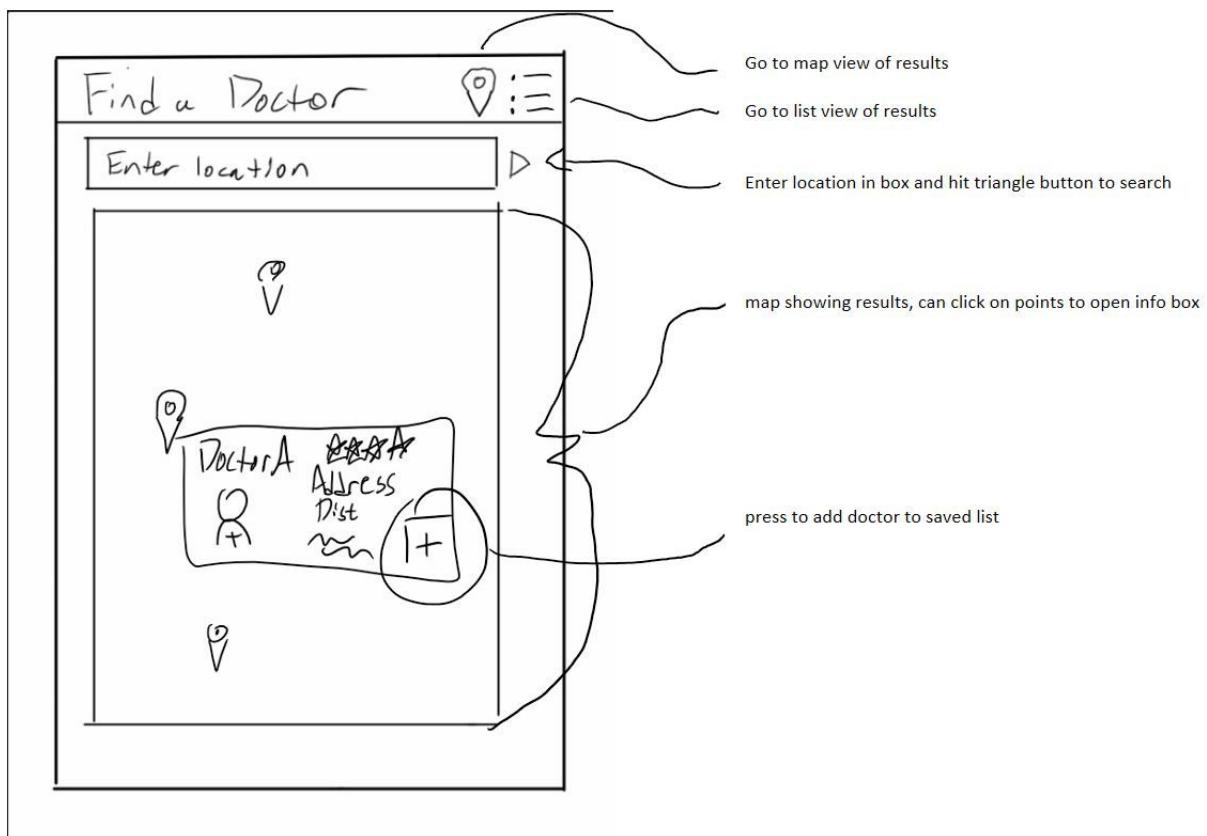
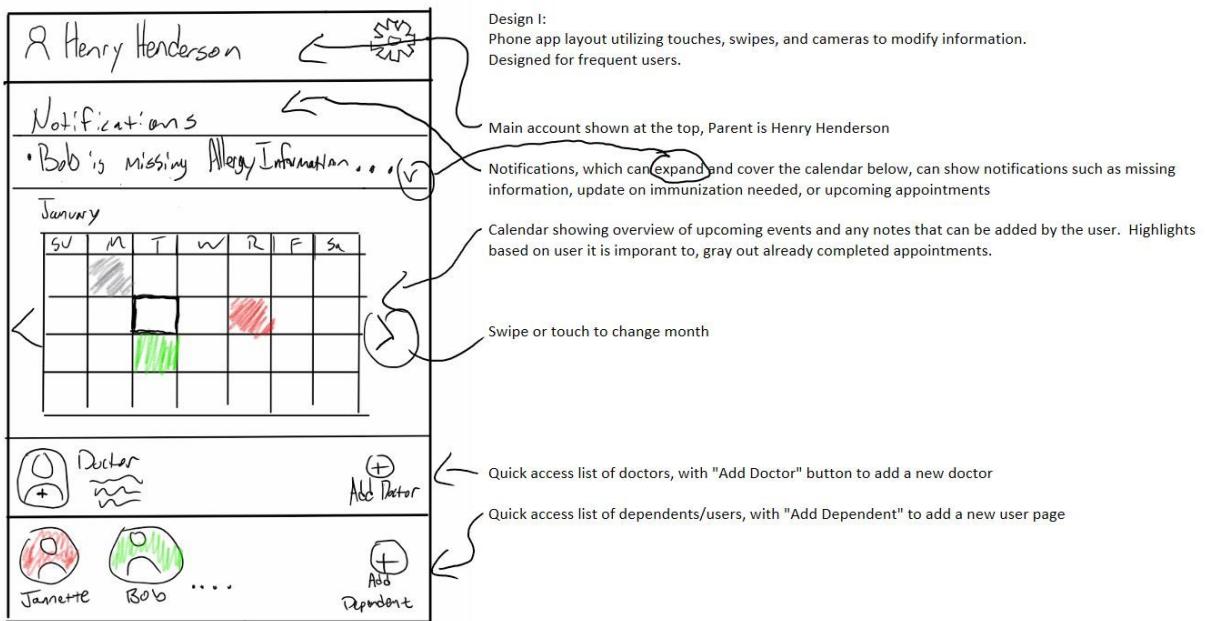
⑤ CONTACT DOCTOR

IN THIS SCREEN THE USER CAN CONTACT THE DOCTOR VIA TEXT MESSAGE OR EMAIL. ALSO ALLOWS THE USER TO NAVIGATE TO THE DOCTOR'S LOCATION

Ryan's Designs:

Design 1

Mobile app layout meant for quick access of dependents and doctors, meant for frequent users.



Find a Doctor		
Sort by name		
(<input type="radio"/>	Doctor A Address Dist	★★★ ~~~~~ +
(<input checked="" type="radio"/>	Doctor B Address Dist	★★ ~~~~~ +
(<input type="radio"/>	Doctor C Address Dist	★★★★★ ~~~~~ +
(<input type="radio"/>	Doctor D Address Dist	★★★★ ~~~~~ +
(<input type="radio"/>	Doctor E Address Dist	★★★★★ ~~~~~ +
:		

List view of same search page, showing information on doctors. Click on name to go to doctor page

Add doctor to saved list

Remove doctor from saved list if already in it

Doctor A

Info

(A) Address (B) Phone (C) Email

For: Bob

Appointment: Checkup / Schedule

	M	T	W	Th	F
1					
2					
3					
4					
5					
6					
7					

Hours: ① Hours

Button to add/remove doctor from saved list

Click on fields to open maps for address, add phone number to contacts, or open email

Auto-send patient information when scheduling appointment

Click to schedule appointment

Select reason for visit

swipe to switch week you are looking at

< (U) Bob >

General Info Edit

Name	Age
Address	D.O.B.
Insurance	
Allergies	

Immunizations

Add Form (+)

Measles, Mumps, Rubella	1/1/14
Flu.pdf	1/1/14
Tetanus.pdf	1/1/14
Polio.pdf	1/1/14

Quickly switch between patients using buttons/swiping

Click Edit Button to modify general information

Add a form from file system or by camera (pdf scanner)

Shows date file was uploaded or modified

Click on Immunization filename to open in pdf viewer

Design 2

Web design meant to utilize increased screen space to minimize pages. Also meant for frequent users that primarily use a computer to manage files.

Go to map to find doctor

Find a Doctor

January

SU	M	T	W	TH	F	SA
App for Bob						

Arrow keys to change calendar month
click on specific date to get more info

Checkup for Jannette 11:00 AM @ Doctor A's office

dashboard always visible for quick access to patients and doctors, click + button to add

access account information

edit basic info for patient

upload file through explorer or drag and drop to file area

clicking on "Bob" from left Dashboard opens up patient page

Click on files to open up in new tab/window pdf

Make and appointment with primary care physician currently selected

Basic Info

Name: Bob D.O.B.: _____
Address: _____
Insurance: _____
Primary Care Physician: _____
Allergies: _____

Up Load File

Files

- Monday Checkup results.pdf
- Monles papers.pdf
- Flue shot report.pdf

Quick-select Find doctor button

Clicking on doctor from dashboard opens doctor page

add doctor to dashboard

Select purpose of visit from dropdown menu
Select patient to pull data from/map appointment to on calendar from dropdown menu
View Doctor Availability using month and week buttons, using arrow buttons to select which week/month

click on available time box to select an appointment time

Hit schedule button to make an appointment and add it to calendar.

The Find a Doctor page

Standard search box and button, utilizing google maps

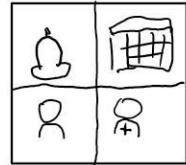
Select a result to show small popup window showing rating, address, and contact information of particular doctor

A list view of the results is next to the map, clicking on the box expands to show more information, with quick contact buttons on bottom for phone, email, and scheduling an appointment, which will take user to doctor's page for scheduling. "+" save button also here to add doctor to dashboard.

Design 3

Meant for extreme case of using a small screen such as on a smart watch, with a focus on gestures to navigate the many screens. Meant for users who might not have access to their phone at all times and want quick access to updates in small chunks of information. Paired with mobile or web app version to edit and add important files/settings, with the watch interface mostly being used for viewing and sending files already attached, but editing smaller things like events or notifications.

Notification bell - display text representation of upcoming appointments or outdated documents

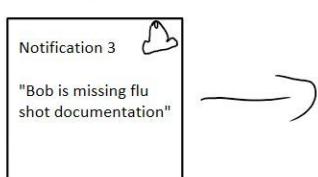


User Icon - go to list of patients/dependents

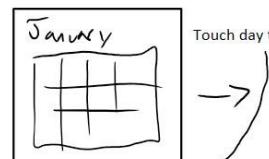
Doctor Icon - go to list of Doctors

notification gives "tweet" like updates such as:

"Bob is due for a checkup"
"Janette has an appointment in 2 weeks"
"Bob is missing the flu shot documentation"
"Doctor A. has to reschedule an appointment"
Swipe up and down to scroll on an update, left and right to go to a different notification
touch and hold to hide update



Touch day to open that day, swipe left and right to go between months



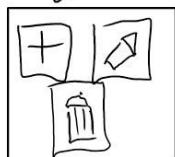
Swipe up to show month view

Scheduler shows a day, with scrolling text for information for that day.

Swipe left or right to show next or previous day

M, Jan 4
Apt. w/ doc A
for Bob, 11AM

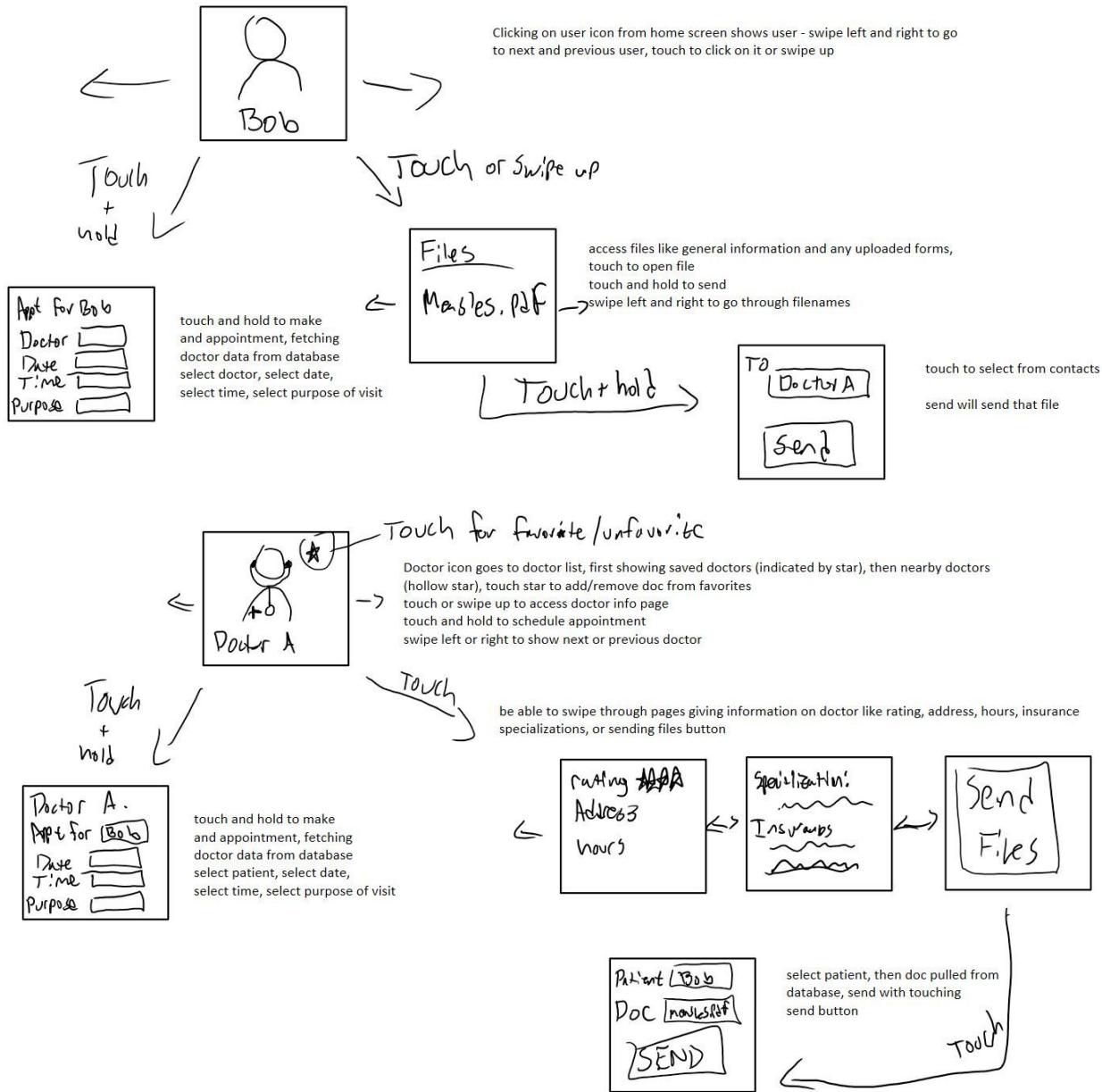
Touch and hold to add/edit/remove event with plus/pencil/garbage buttons



Add/Edit



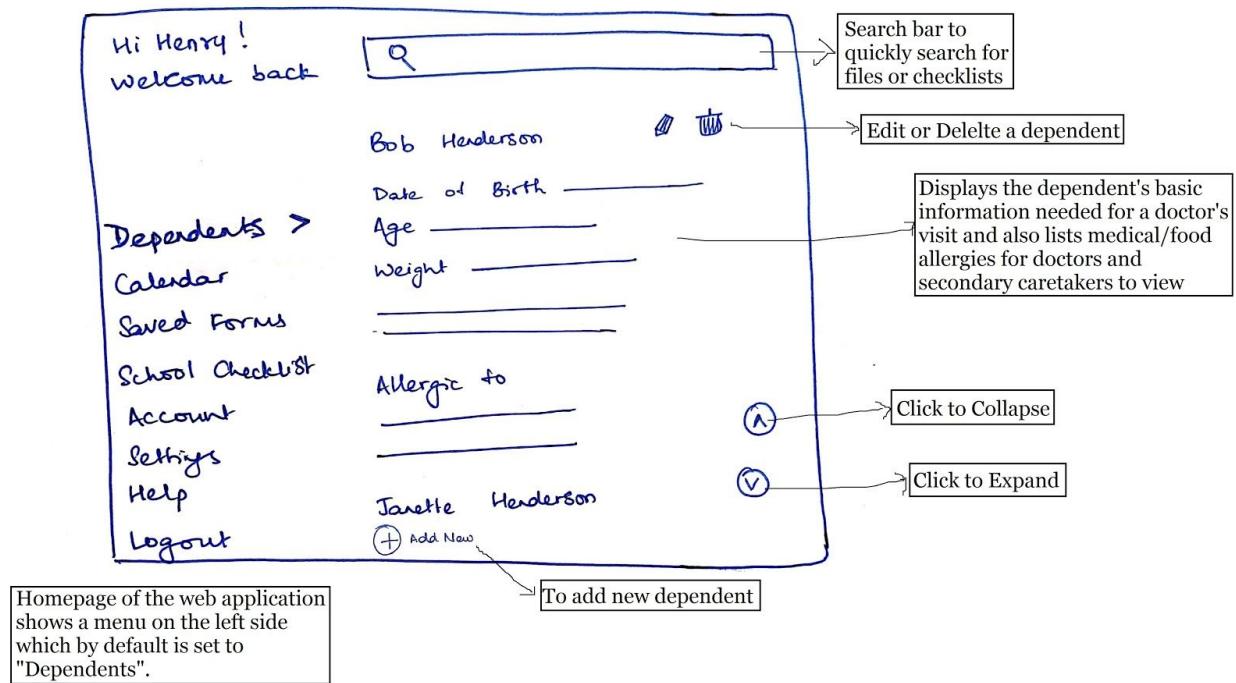
touch box to select values for time,
note (Generic - appointment, checkup, vaccination/etc.
Doctor (From saved or nearby doctors list)
Patient (From saved list of patients)



Shantanu's Designs:

Design 1

This design is for a web application with extensive menu items and links.



Hi Henry!
Welcome back

Dependents
Calendar >
Saved Forms
School Checklist
Account
Settings
Help
Logout

Upcoming Appointments
02/14/2019 Dr. Smith (800 S Morgan St.)
Patient: Bob Henderson

④ Add New Appointment
See past Appointments

Notes
02/01/2019 Bob's report due
02/01/2018 Janette's Ear Infection, Doctor gave Amoxicillin

④ Add New Note
See all Notes

A list of all past notes is stored in the application and can be viewed upon clicking this link.

The "Calendar" tab shows your upcoming appointments. For convenience, there are shortcuts to call or mail the doctor. You can edit or delete the appointments.

To add a new appointment the user will have to fill in basic details like the date of appointment and which of the dependent's will be seeing the doctor.

A list of all past appointments is stored in the application and can be viewed upon clicking this link.

The user can also add quick notes and optionally get email reminders.

To add a new note the user will have to input the date, the note itself and opt to get reminders on their email.

Hi Henry!
Welcome back

Dependents
Calendar
Saved Forms >
School Checklist
Account
Settings
Help
Logout

SAVED FORMS

BOB JANETTE Add Folder

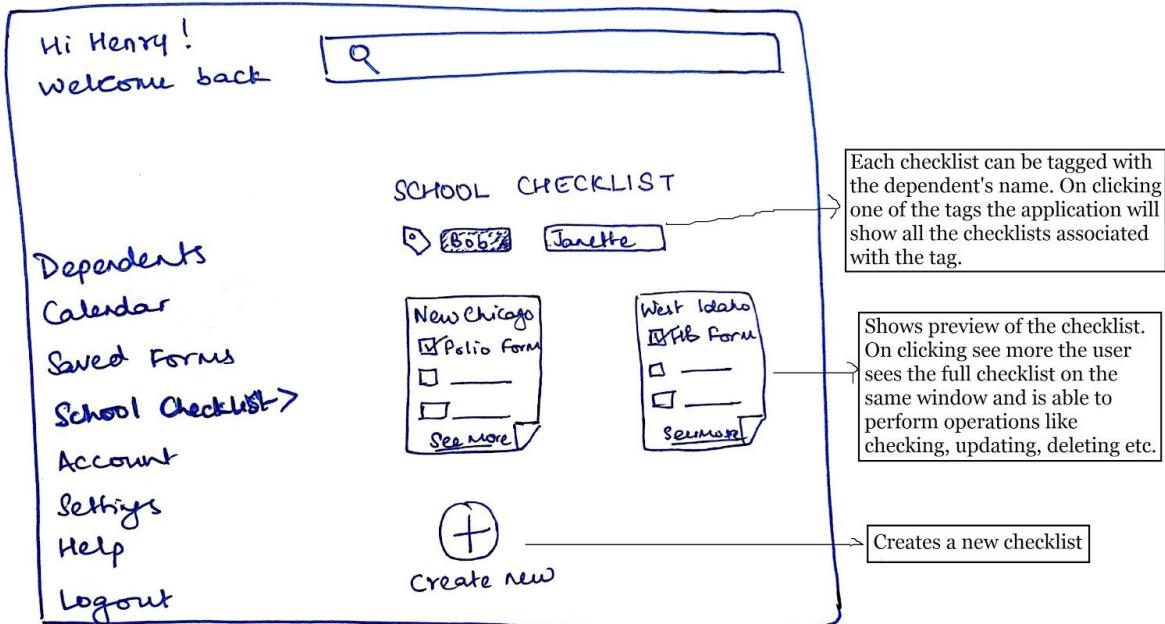
- Polio vacc. form
- _____
- _____

④ Add new form

Easy folder view to segregate dependents' files.

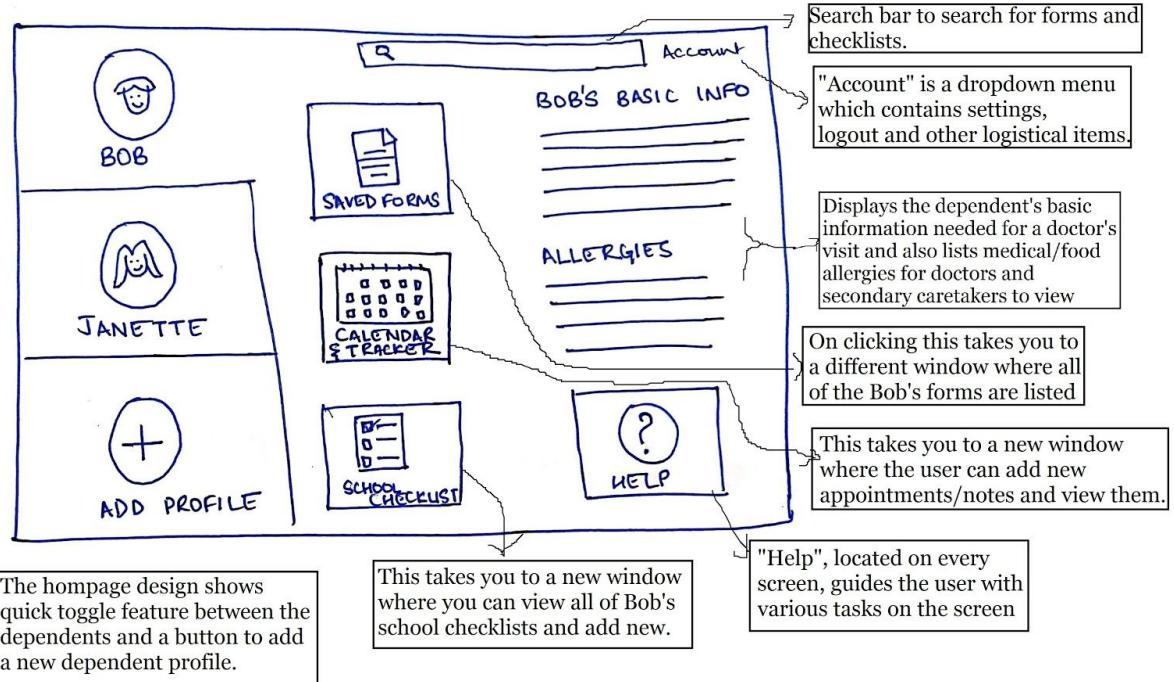
List of Bob's forms. Each form can be easily downloaded, edited or deleted.

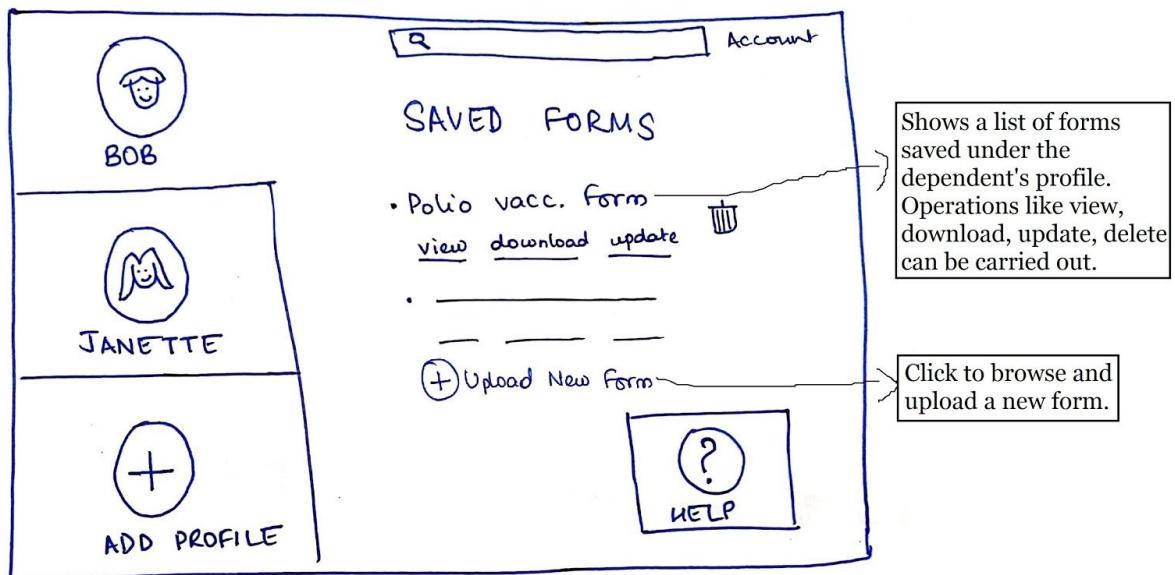
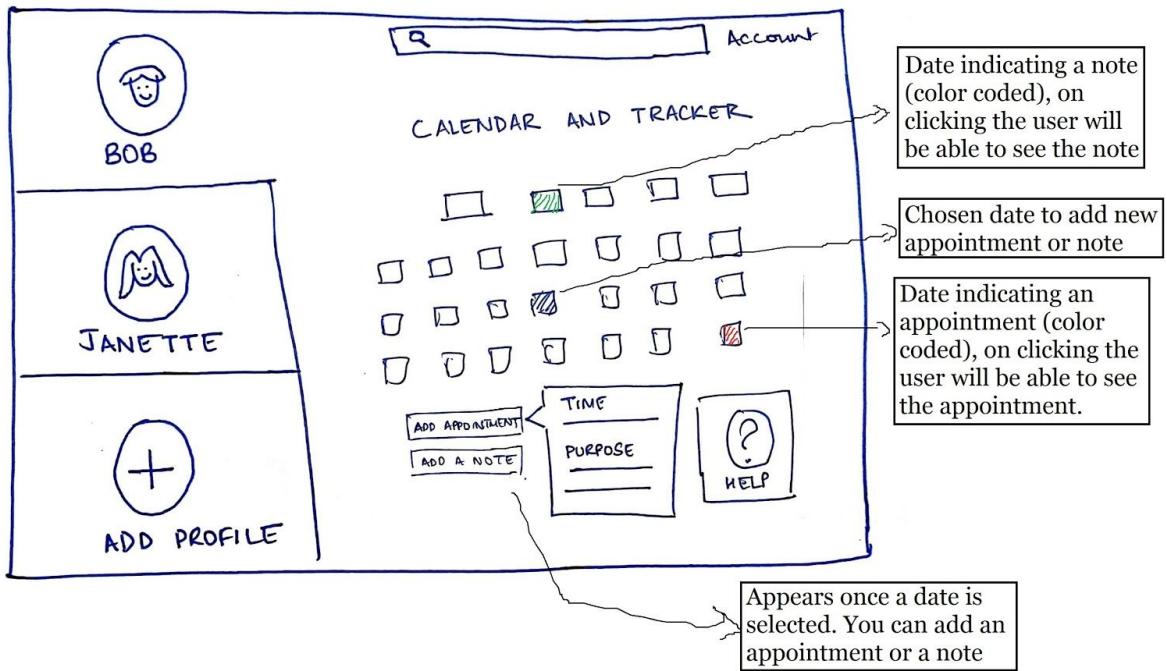
To upload a new form from your system.

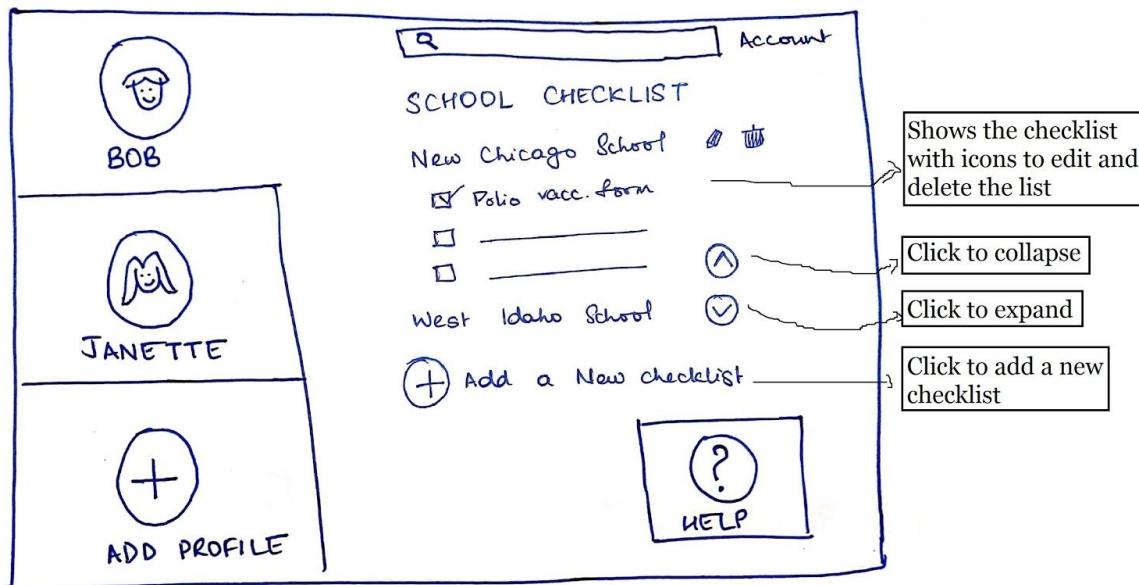


Design 2

This design focuses on easier navigation.



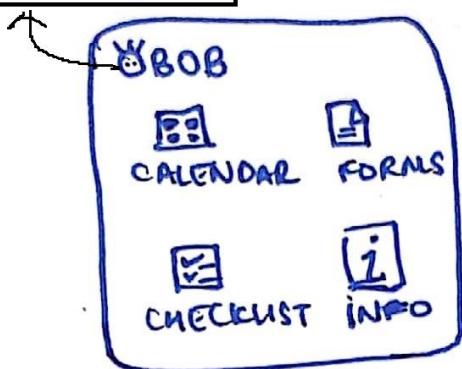




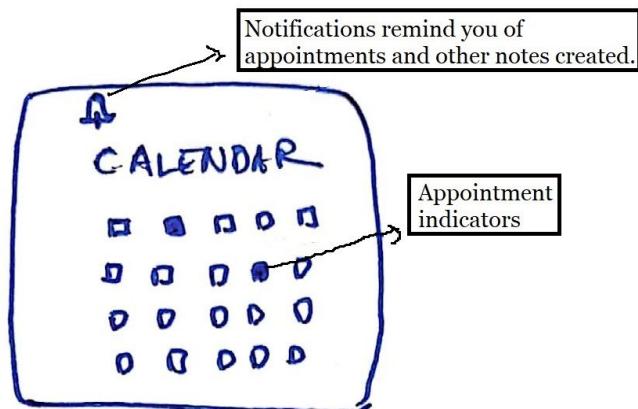
Design 3

This is an extreme design for a tiny-screen (smart watch). This design is not intended to be a stand alone application and needs to be paired up with the web application. This design is for users to have a quick look when they can not access their laptop/desktop computers.

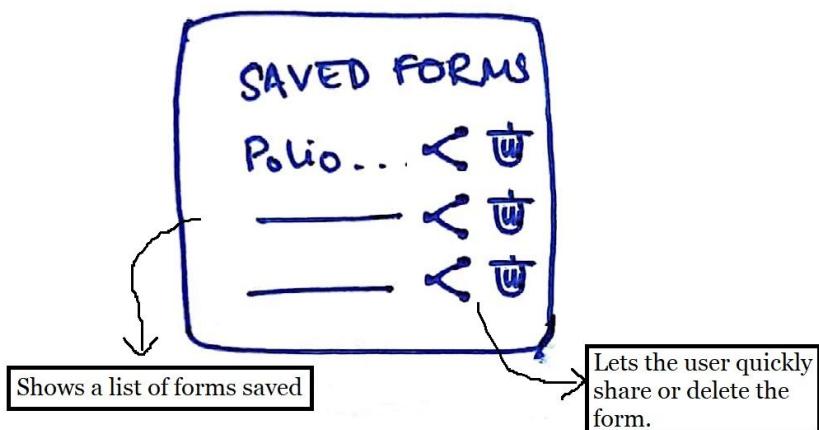
Profile of the dependent. On selecting this item you can switch between the profiles.

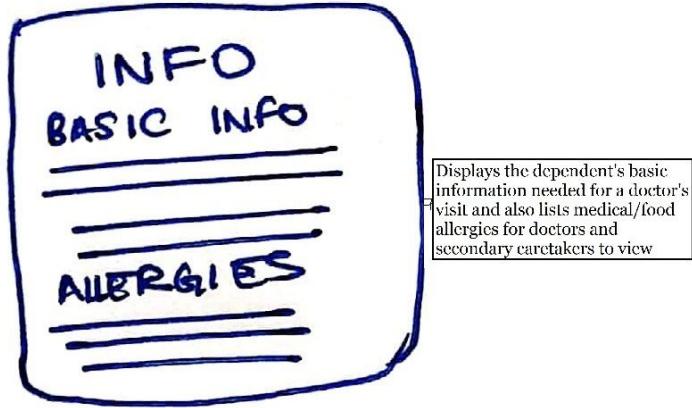


This design is for smart watches. The app homepage lets you toggle between four major menu items.

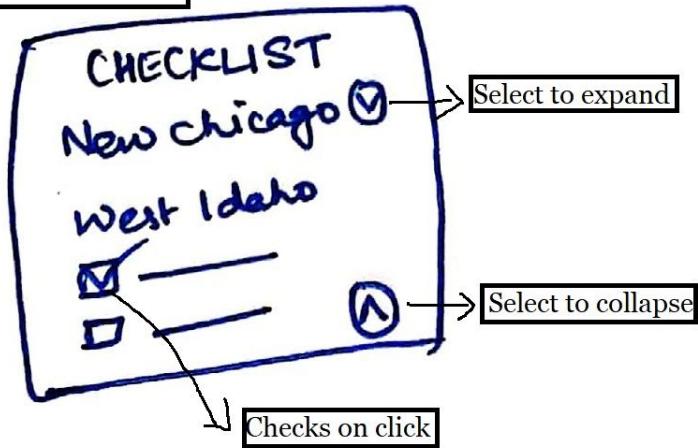


New appointments and notes can be created through voice commands or through the web application.





Displays various checklists stored under the dependent profile.



Jon-Michael's Designs

***NOTE:** Because the images are too large, they are too small on this document. On this link, I will provide you with larger versions of the image:

https://www.dropbox.com/home?preview=jhoang6_GR2_designImages.pdf

or

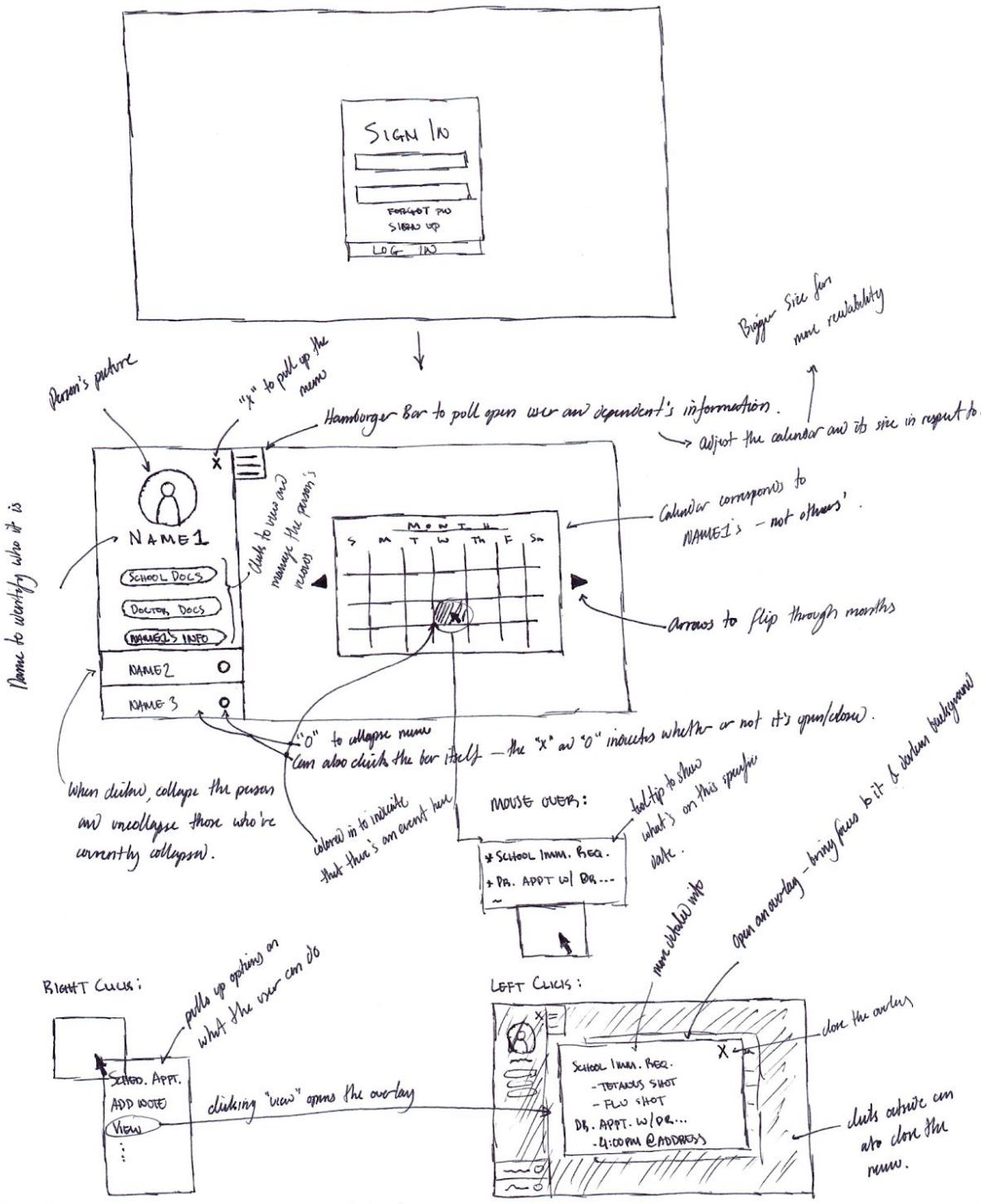
https://www.dropbox.com/s/vjcu8a1yfx54tbf/jhoang6_GR2_designImages.pdf?dl=0

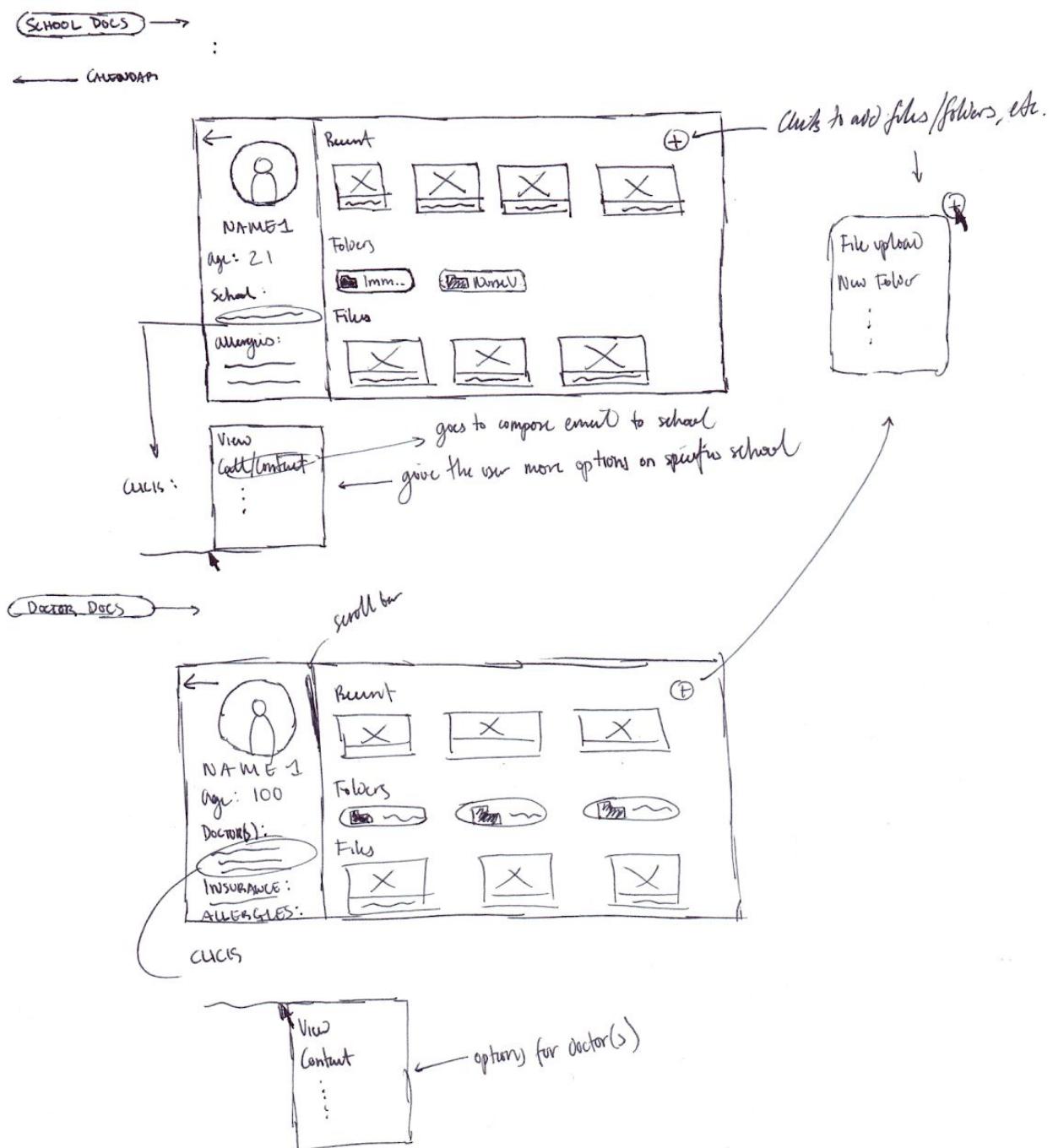
*The first three images correspond to Design 1, the next three for Design 2, and the last image for Design 3.

Design 1:

This design is for the web application which utilizes overlays to mitigate pages and to allow the user to easily navigate and keep track of their things.

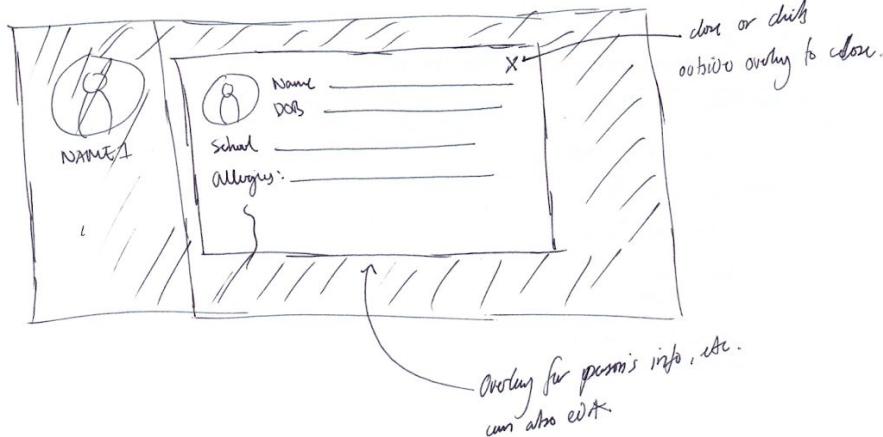
Here, is the main page which primarily consists of a calendar to the specific person selected on the left of the screen.





The file manager here allows the user to add and create files relevant to the person that they have had previously selected. The interface also has a recents feature which will list the document that the user has had used last time and folders to help organize the documents.

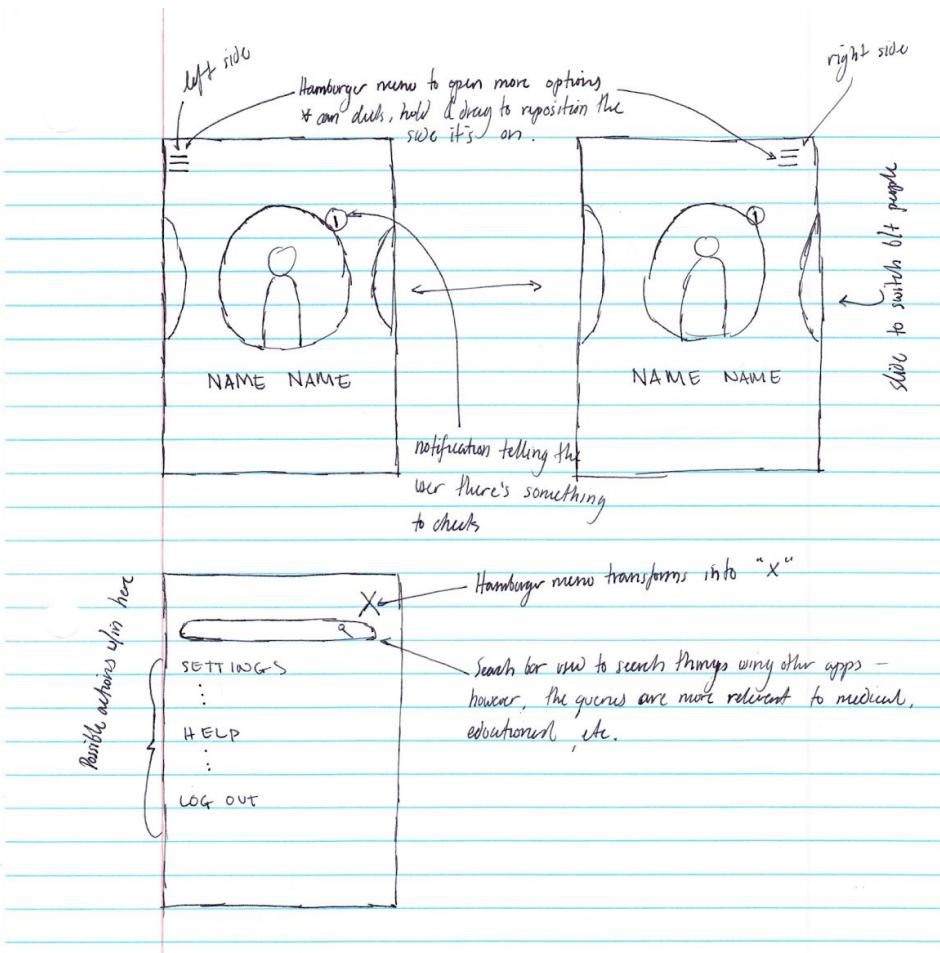
INFO →



An overlay to manage the user's information.

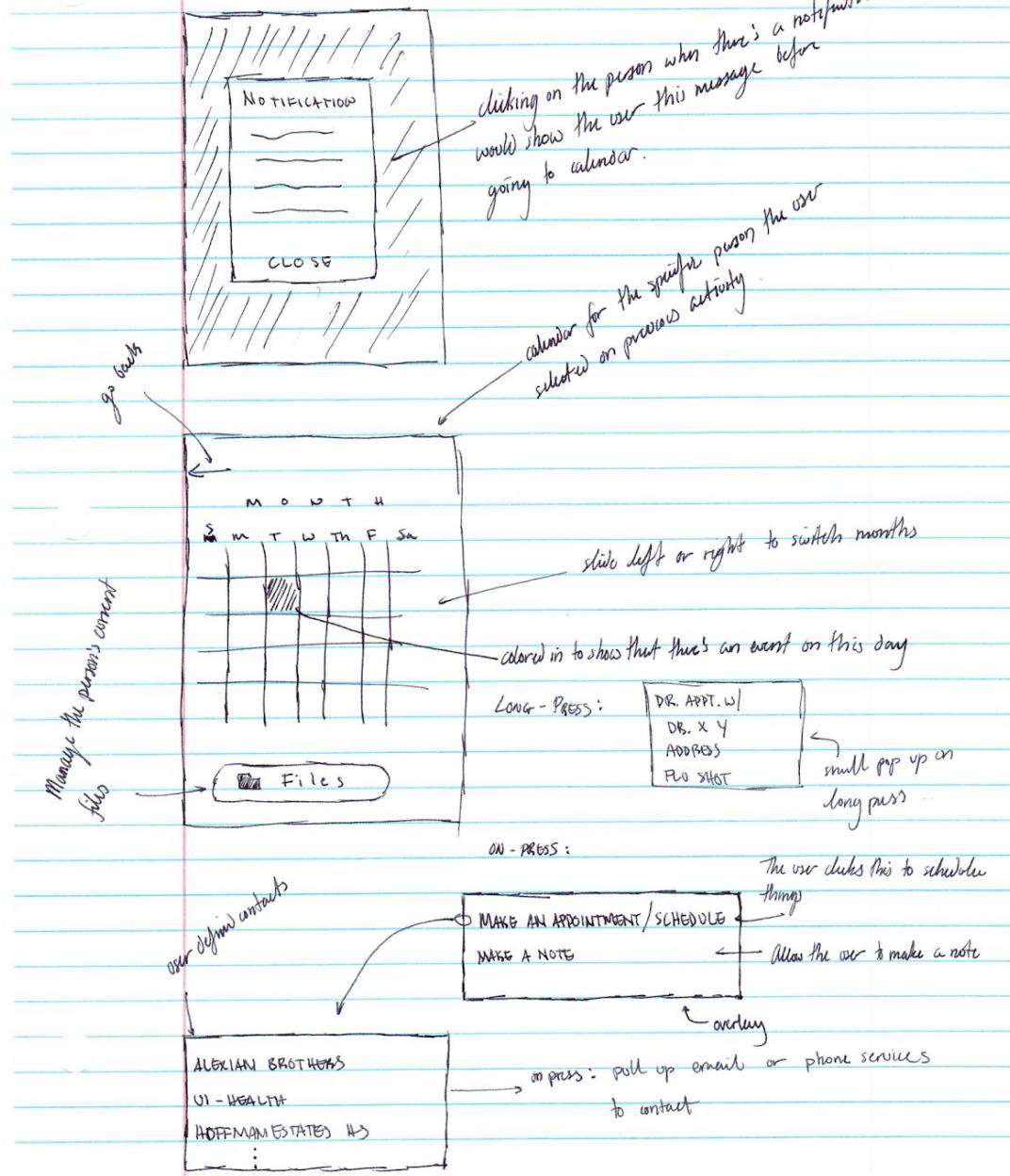
Design 2:

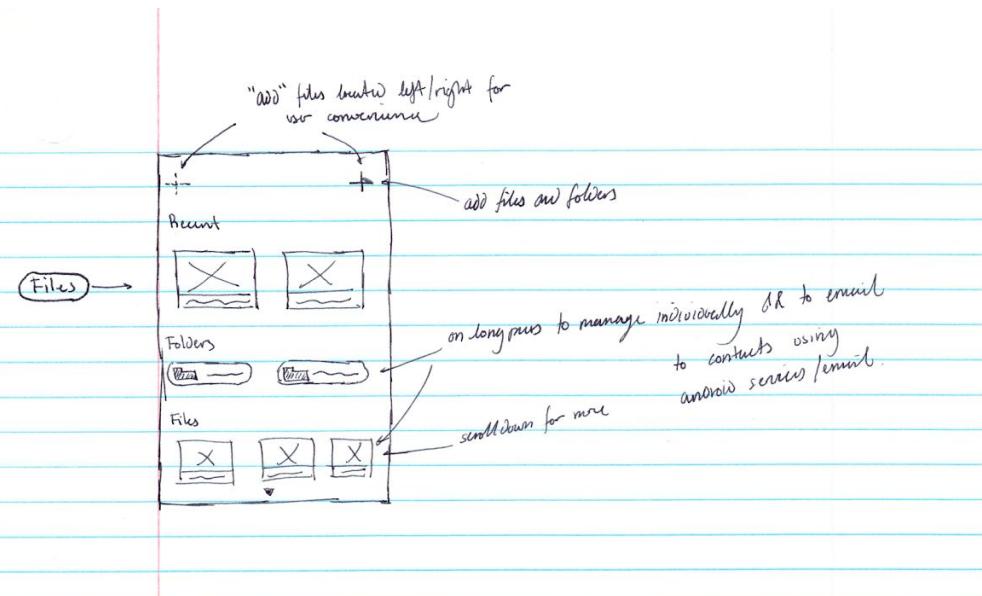
This design is for a mobile application - which allows the user to access and manage their information and be able to make contact with their doctors. The application minimizes the amount of pages the user has to go through by utilizing overlays to allow the user to view and edit things with the ability to easily go back and not have to keep up with managing pages as a result.



Here, one can see that there are user profiles which allows the user to choose who's on the list and to choose a person to schedule appointments with. Not only that, the hamburger helper expands to allow the user to search easily for doctors and institutions as the pulled results are not only based on the query but related things in education and doctoral, etc.

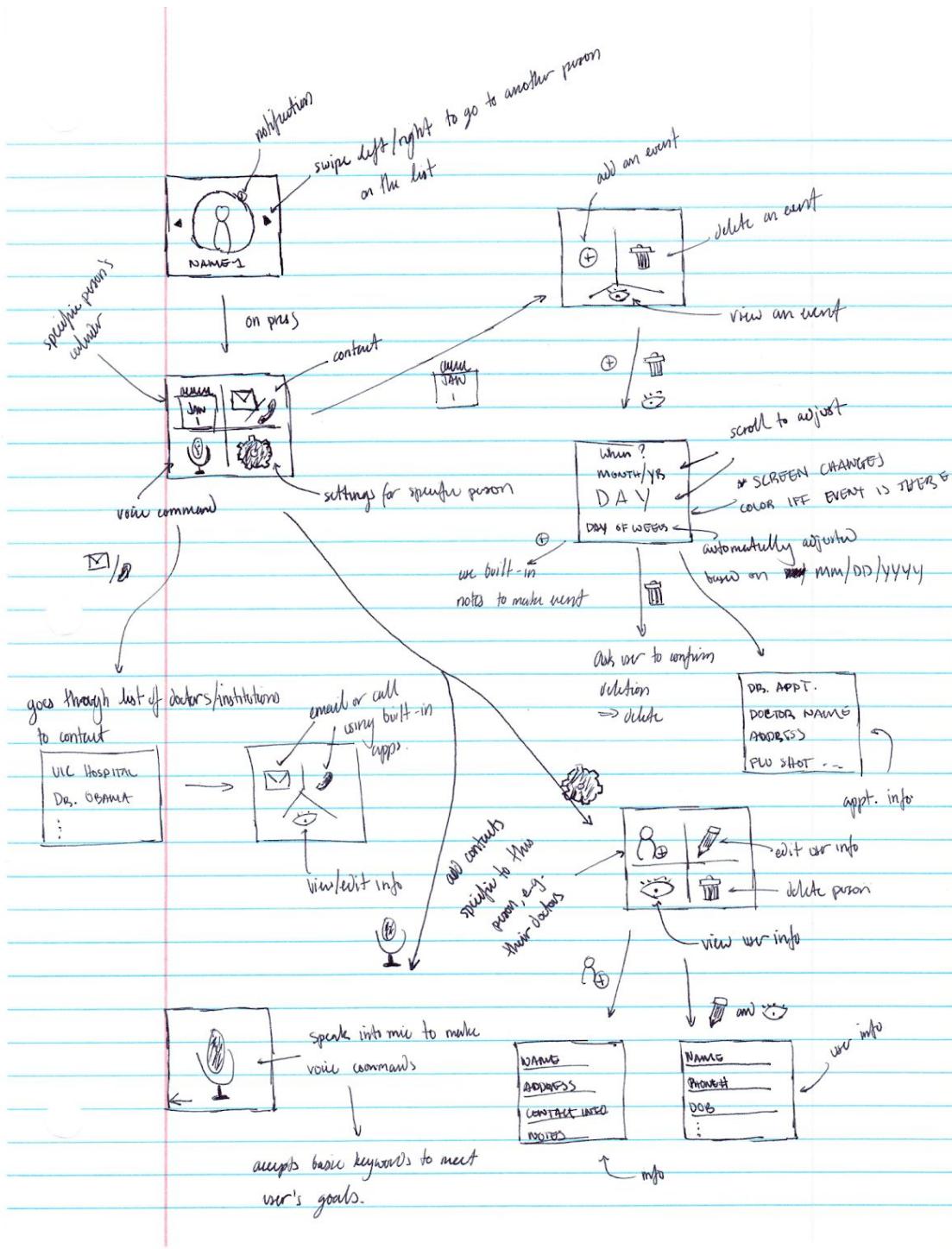
Here, use overlays to allow faster actions and less page management. There also exists a calendar to keep track of a specific person's schedule and a file button which takes the user to the file manager.





The file manager here allows the user to add and create files relevant to the person that they have had previously selected. The interface also has a recents feature which will list the document that the user has had used last time and folders to help organize the documents.

Design 3:



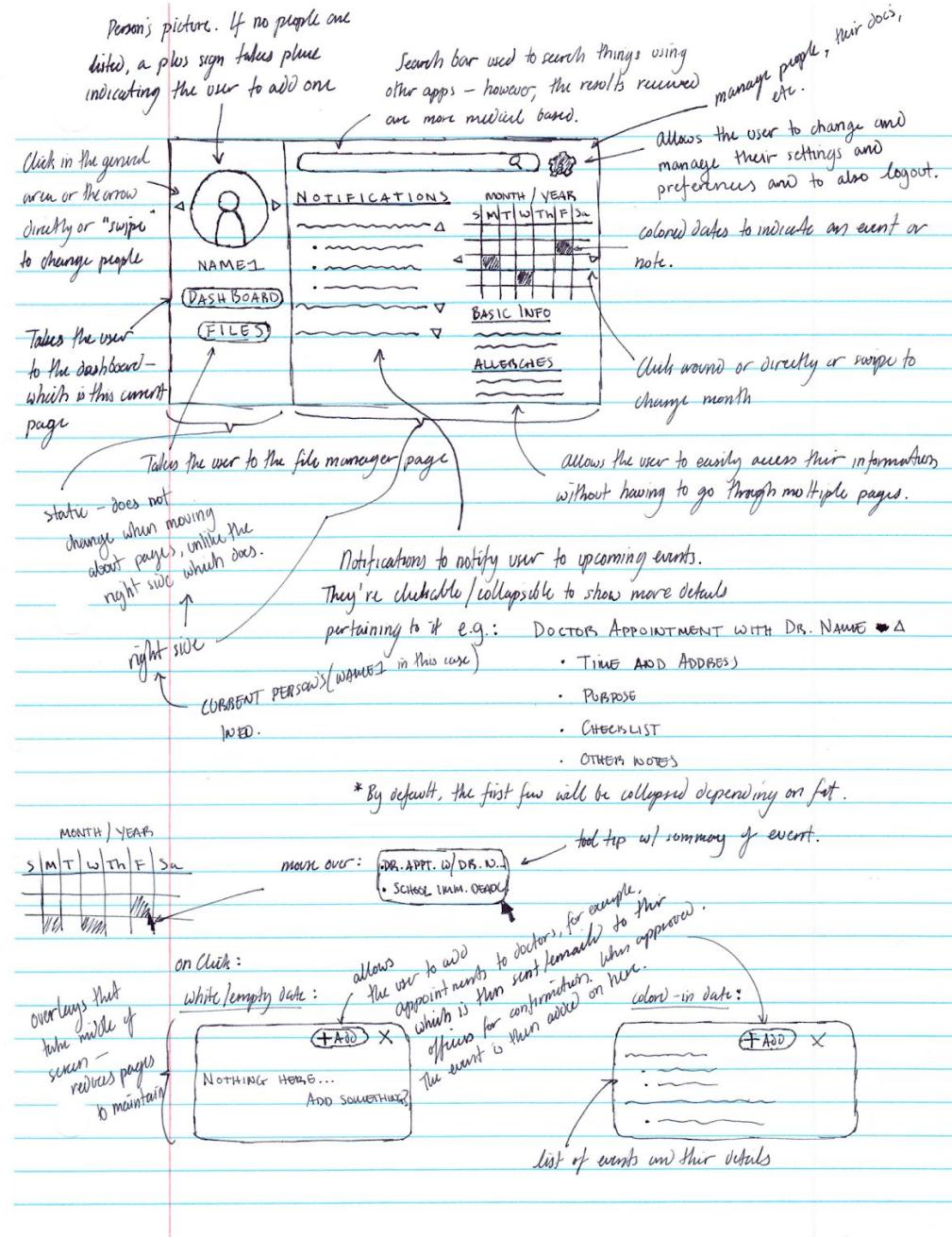
This is a sketch for the smartwatch design of the project, which utilizes both hand/finger gestures and voice commands to navigate the application. This is meant for those who do not have consistent access to their phones but do have access to their watch to make calls, edit contacts and information, etc.

One can note the symbols used to allow the reader to correspond them to everyday actions to meet their goals and instead not having to read such small text on a small screen.

Group Designs:

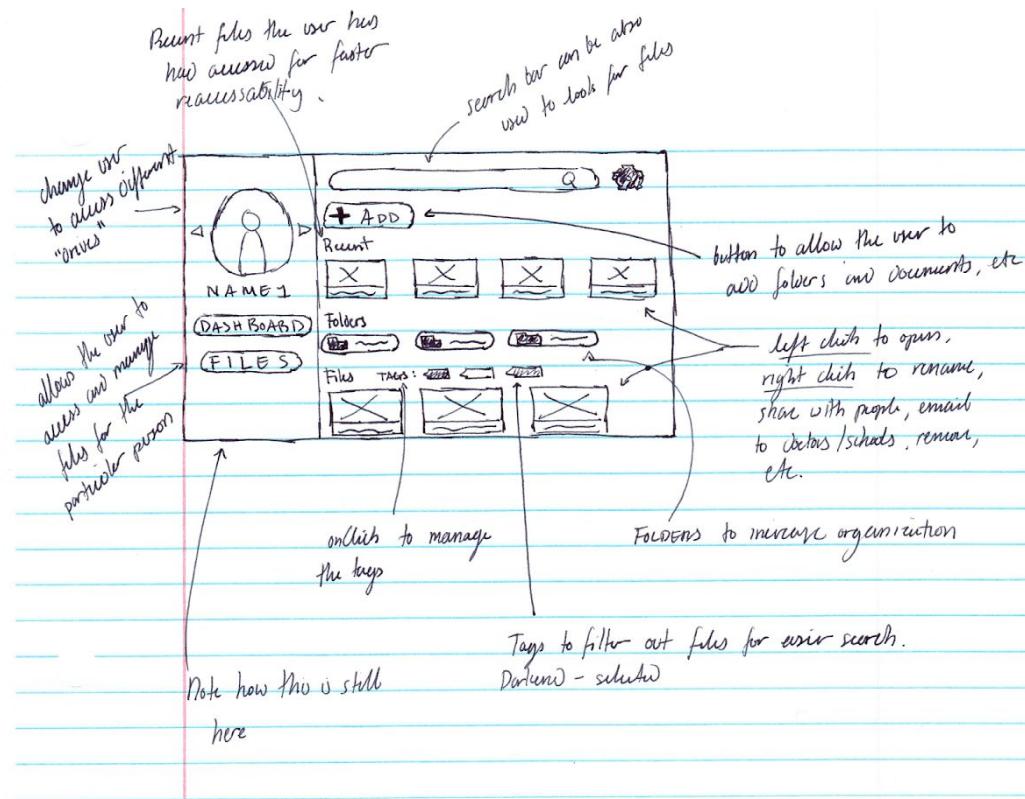
- 1) https://www.dropbox.com/s/2gdv5cun3uaaf15/GR2_GD1.pdf?dl=0

Design 1



Henry has set up an account on *docdoc* by providing all the necessary information of his kids, Bob and Janette. Now, Henry wants to add the details of the appointment he just scheduled with Dr. Brown, he left clicks on the specific date on the calendar widget on *docdoc*'s dashboard. Henry adds all the details and saves the appointment (goal 1).

On the day of the appointment, Henry has forgotten what forms he needs from the doctor! He quickly checks *docdoc*'s dashboard and expands the notification which has a checklist of all the forms Bob's school had requested (goal 2).



When Dr. Brown gives Henry a big stack of forms to carry, Henry quickly makes electronic copies of all the forms. He clicks on the “Files” button on Bob’s profile and then clicks on “Add” to upload the new files (goal 3). He selects all the files he need to send and right clicks to share it with Bob’s school via email (goal 5).

Henry notices he missed getting polio vaccination form from the Doctor and panics. Then, he realizes that last year’s vaccination form is still stored in docdoc! He quickly shares the file with the school staff (goal 4).

Design 1 Analysis:

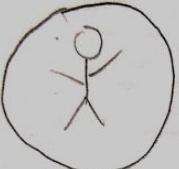
This design focuses on efficiency. Despite the inconveniences the user has to go through when initially using this application, the application is simplified such that the user’s goals can be met with a couple of simple clicks. Having the person’s basic information, their upcoming events, and a calendar that they can interact with within one page on the dashboard allows for increased efficiency when scheduling for things and creating events/notifications. The file managing page allows the user’s efficiency by allowing them to view their recent files near the top of the page, just in case they want to re access them, folders to organize files, and tags to filter out certain documents. There also exists a search bar for the user to look up doctors and institutions in the case where they are looking for one.

However, there are still mistakes that may or may not occur. Although minor, it should be noted that since reaching the goal takes a few steps coupled with possible infrequent usage of the application, mistakes still occur. Things such as accidentally uploading/deleting things and accidentally changing a person’s information are some of the things that may occur.

Design 2

Henry logs onto *docdoc* and creates an account. He first sees the account creation page, where he populates information for himself and his kids, Bob and Jannette, which include forms that have any relevant information for each kid. There are also fields that allow Henry to fill in the contact info and names for the schools that Bob and Jannette currently go to, which are displayed on their page.

EDIT Profile



Medications

+

Allergies

+

Conditions

+

Blood Type: ▾

Notes

+

Name*

Age

Phone - -

Email

Address

Locate me

Access: Admin Dependent

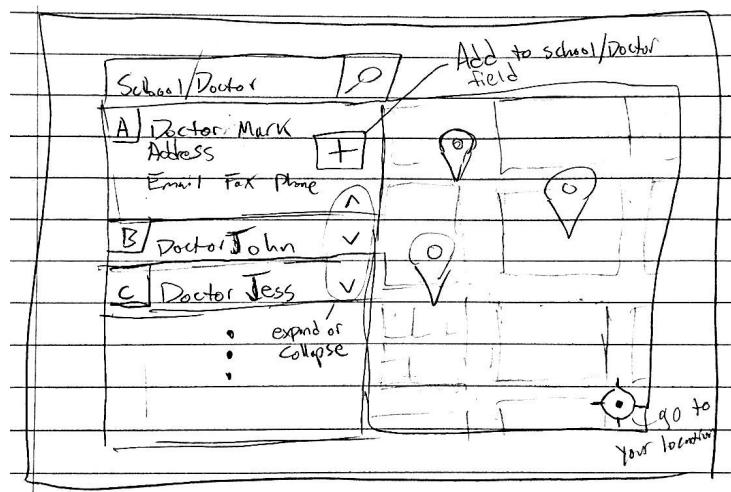
Schools:

Name	Address
Email	Fax
	Phone

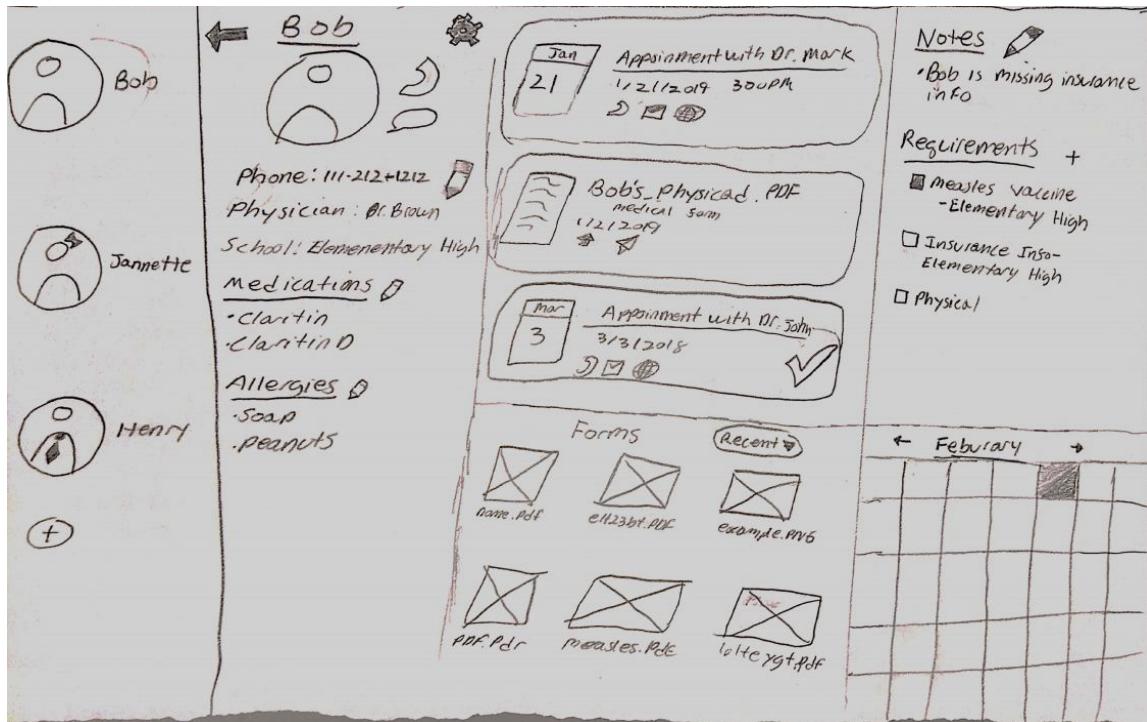
Physicians:

Name	Address
Email	Fax
	Phone

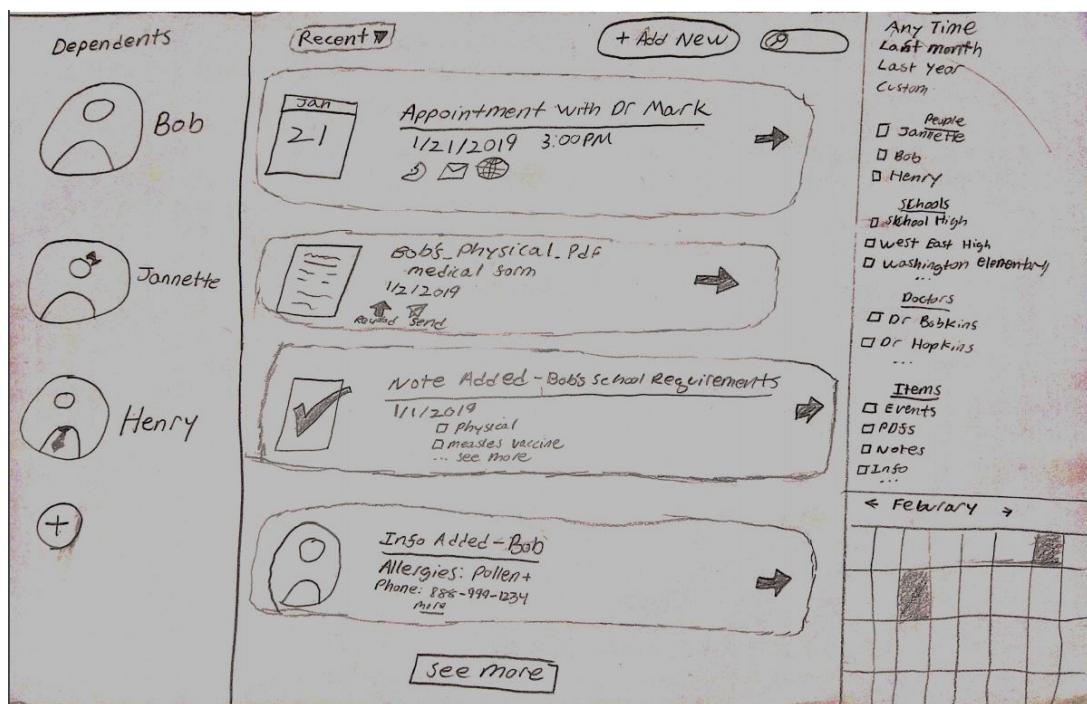
Not having all the contact info for Bob and Jannette's GP on hand, he clicks on the handy "Search" button on account creation, which links him to the institution search feature. A list of nearby doctors and schools pulls up. He quickly finds Dr. Mark, and clicks on the "+" button to easily auto-populate the forms.



Later, Bob needs to schedule an appointment with Dr. Mark for bob. On Bob's profile page, he sees Dr. Marks contact number, and calls to schedule an appoint for January 21 (goal 1). He double clicks on 1/21 on the calendar in the lower-right corner, which brings up the standard 'add event' dialogue for that date. He fills in information to remind him of the appointment, which shows up on the newsfeed for Bob as his most recent items.



Later, Henry needs to remember when his appointment with Dr Mark was. He opens up the homepage and sees that it's indeed January 21st. He then adds a note to himself, to remind him that he needs to get a copy of the physical information this time (goal 2). He clicks on the "Add Event" button on the home screen, which brings up the same interface as before.



Add Event

Date PDF Note

+ title

description (optional)

+ [bullet]

After his visit to Dr. Mark, he remembers to ask for a copy of the physical forms, thanks to *docdoc* (goal 4). He uploads the files, using the same “Add event” interface on *docdoc*’s home screen (goal 3).

Add Event

Date PDF Note

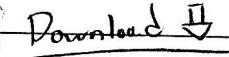
Title

Date

Description

File upload

The file for Bob's physical then appears on top of the home screen. Henry clicks on it, bringing up a preview. He clicks on the "send:" button, selects Henry's school, and easily forwards the pdf to the school using the info he already had filled out. (goal 5).

Preview:	
Send Form.	Patient: Bob Henderson
to: 1	Form: Physical
Dr. Brown Elementary High	
Download 	

Design 2 Analysis: This design is good for focusing on efficiency. The initial setup for learning the interface, and the amount of information that has to be pre-populated when setting up an account make the learning curve somewhat steep. However, the design is streamlined so that typical repetitive actions, such as scheduling and sending forms regularly, are done in only a few buttons. The generous use of space makes finding information easier and helps the user not get lost, but also may make it more difficult to learn for novice users. The interface also makes assumptions for efficiency that more recent or upcoming events are more useful, and that the patient uses nearby doctors. Minor mistakes are relatively easy to recover from, due to the relatively few steps and infrequent use of the app. Fields can be changed easily, and files can be easily removed or added. Some tradeoff is made for efficiency when uploading, so it is possible to make errors or delete files by mistake, which might be missed.

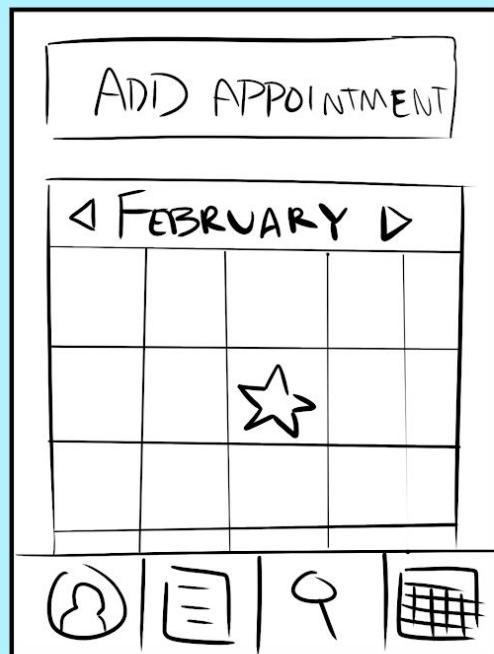
Design 3

Design 3: DocDoc Mobile App

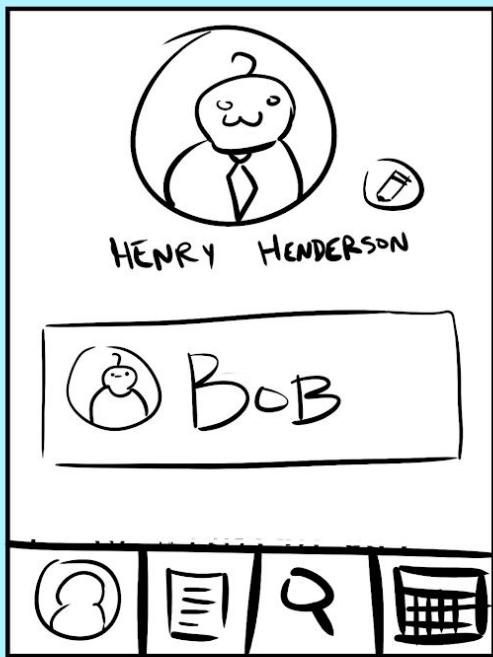


Henry goes to search for a pediatrician, he presses the search icon on the icon dock, and then searches for "Chicago Pediatrician". He finds the doctor and his contact info so that he can schedule an appointment

Henry then presses the calander icon and then sees he can "Add Appointment". He presses the button and then schedules the appointment to the date with a note attached reminding him what the appointment is about. In this case, it is for his child Bob.

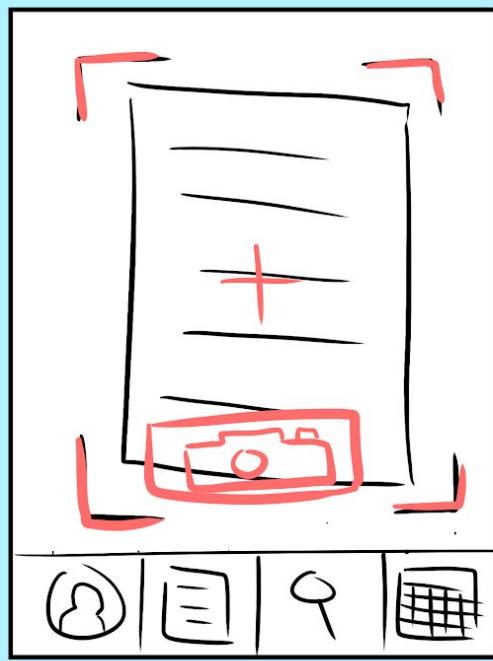


Design 3: DocDoc Mobile App



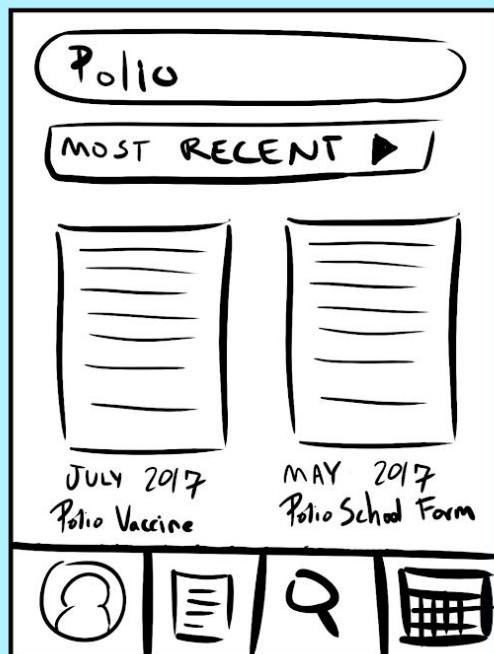
Henry sees that he will need to find Bob's profile to find what he needs, he presses the profile icon and is sent to the profile page which lets him navigate to Bob's profile

Henry wants to save the documents into Bob's profile, so he presses the document icon which allows him to quickly pull up his phone's camera and then take a picture of the document to quickly save it into Bob's profile.



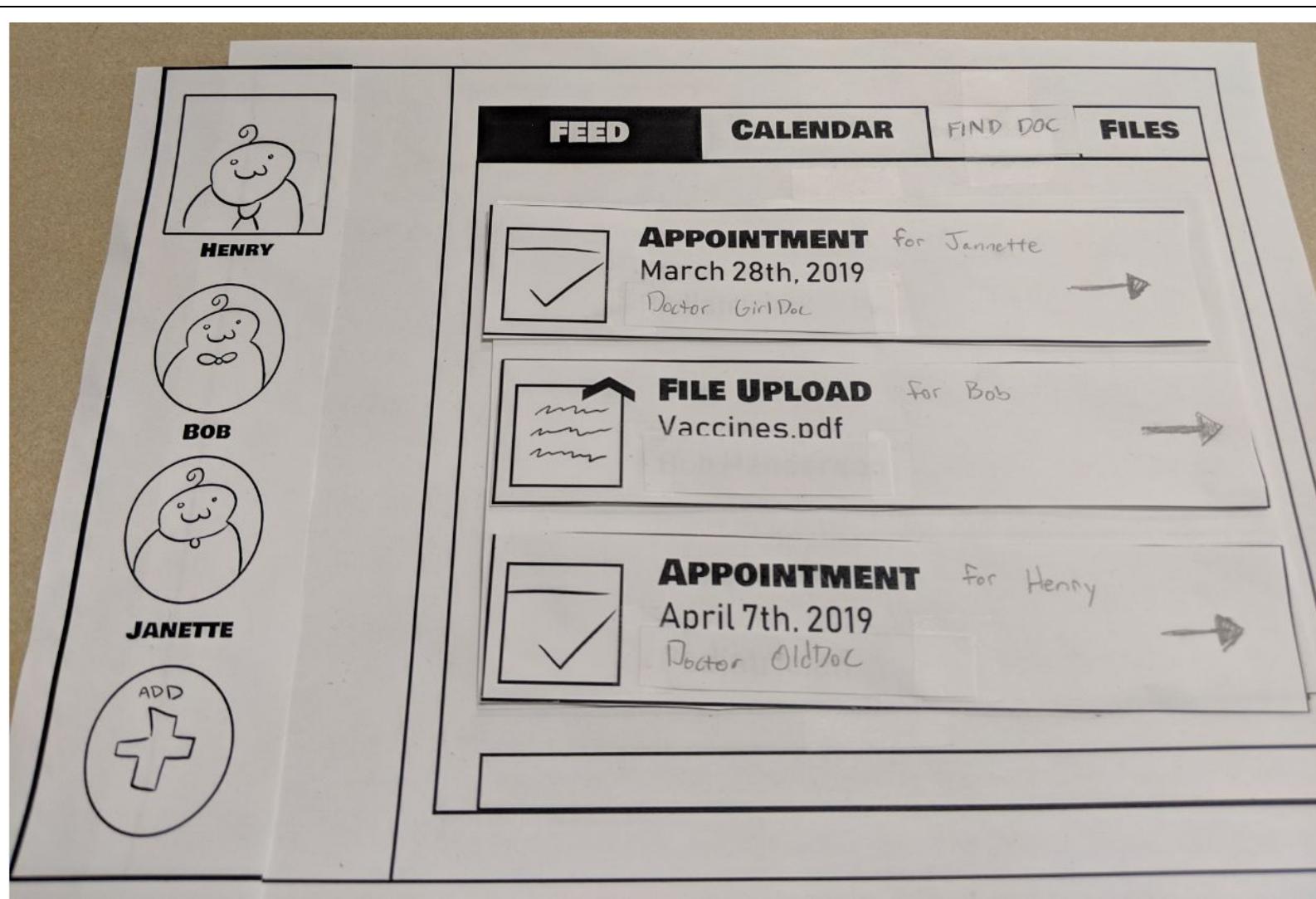
Design 3:

Henry then needs to look through Bob's files he has saved in the past, he finds it and sees a share button after he selects it so that he can quickly email the image to the school.



Analysis:

Design 3 is a mobile app UI which enables the user to quickly figure out how to use the UI and keeps complex things organized and easy to manage and navigate through. This is done through the use of an intuitive database for tracking forms and information. The simplicity of the app allows for quick learning maximizes learnability. The app focuses on important features, which keeps efficiency high for common tasks, but limits the flexibility or efficiency for more complicated tasks the user may want.

GR3 -Paper Prototyping:**Images of Prototype:**

Feed/Homepage for the app. "Map" tab has been renamed to the "Find Doc" tab. Appointments in the feed now also include annotations for the person appointments are with.

BOB HENDERSON

Allergies:
Peanuts

Medication:
Blue Pills

Age: 14
Weight: 150
Blood Type: O+

Doctor Info

+FIND DOCTOR+

School Info

Name:
The Middle of the Elementary High School

Email:
nurse@mehs.com

4880 School Dr.

FEED **CALENDAR** **FIND DOC** **FILES**

FILE UPLOAD
Vaccines.pdf

for Bob

Selecting Bob's picture on the right brings up Bob's account information on the left side. While Bob is selected, information displayed in the tab is filtered by those associated with Bob, and the "Find Doc" saves new doctors to Bob as a default.

SHOWING: PEDIATRICIANS NEAR ME

FEED

CALENDAR

FIND DOC

FILES



Dr.Realman
Pediatrician
A

420 Lake Shore Dr.
(312) 555-5555



Dr.Fakerman
Pediatrician
B

550 N Mountain Rd.
(312) 542-9999



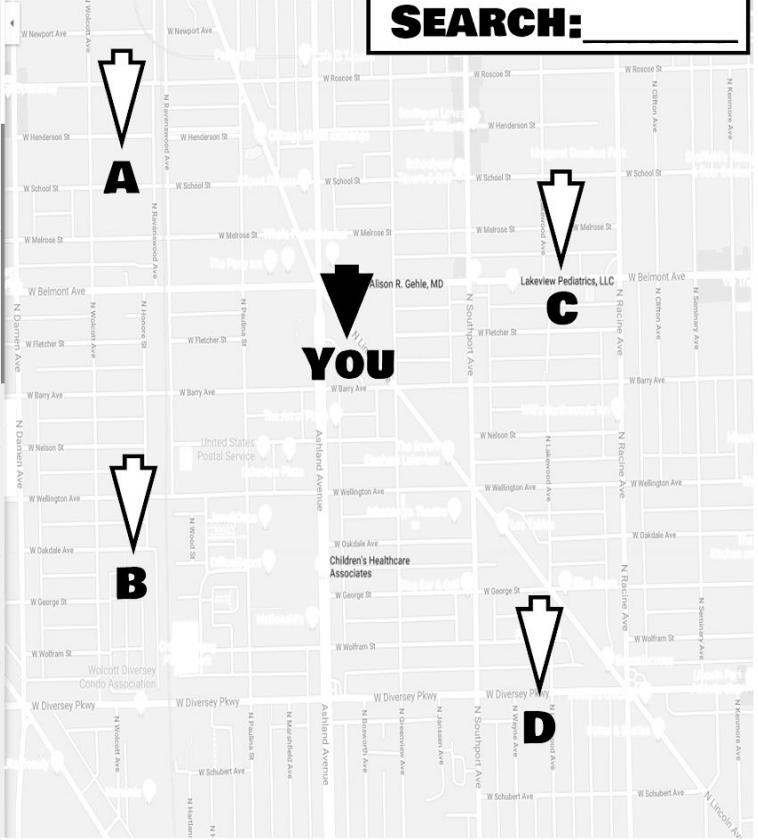
Dr.Feelsbad
Pediatrician
c

8520 W. Chicago Blvd.
(312) 995-9111



Dr.Gandalf
Pediatrician
D

7892 E. Forsure Rd.
(312) 911-8989



SEARCH: _____

"Find Doctor" tab now shows a map as well as the location of the current user.

SHOWING: PEDIATRICIANS NEAR ME

FEED

CALENDAR

FIND DOC

FILES

SEARCH:



Dr. Realman
Pediatrician

A

420 Lake Shore Dr.
(312) 555-5555



Dr. Fakerman
Pediatrician

B

550 N Mountain Rd.
(312) 542-9999



Dr. Feelsbad
Pediatrician

C

8520 W. Chicago Blvd.
(312) 995-9111



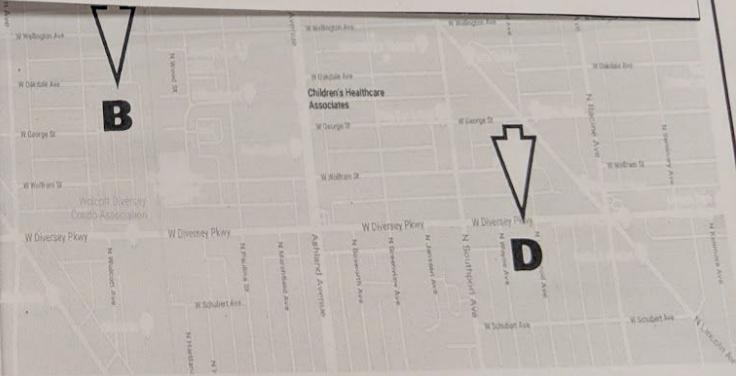
Dr. Gandalf
Pediatrician

D

7892 E. Forsure Rd.
(312) 911-8989



B



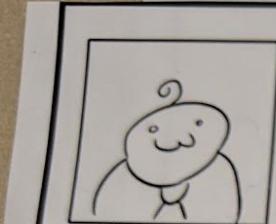
Dr. Realman

Phd. Pediatrician

420 Lake Shore Dr.
(312) 555-5555
email@faker.com

Add to:
 Bob
 Janette
 Henry

Save



HENRY



BOB



JANETTE



ADD

Selecting a doctor in the "find doctor" tab brings up a window that lets you save the doctor to the accounts of different users. User-binding is done with a selection menu that would be pre-populated if a user is selected on the right.



HENRY



BOB

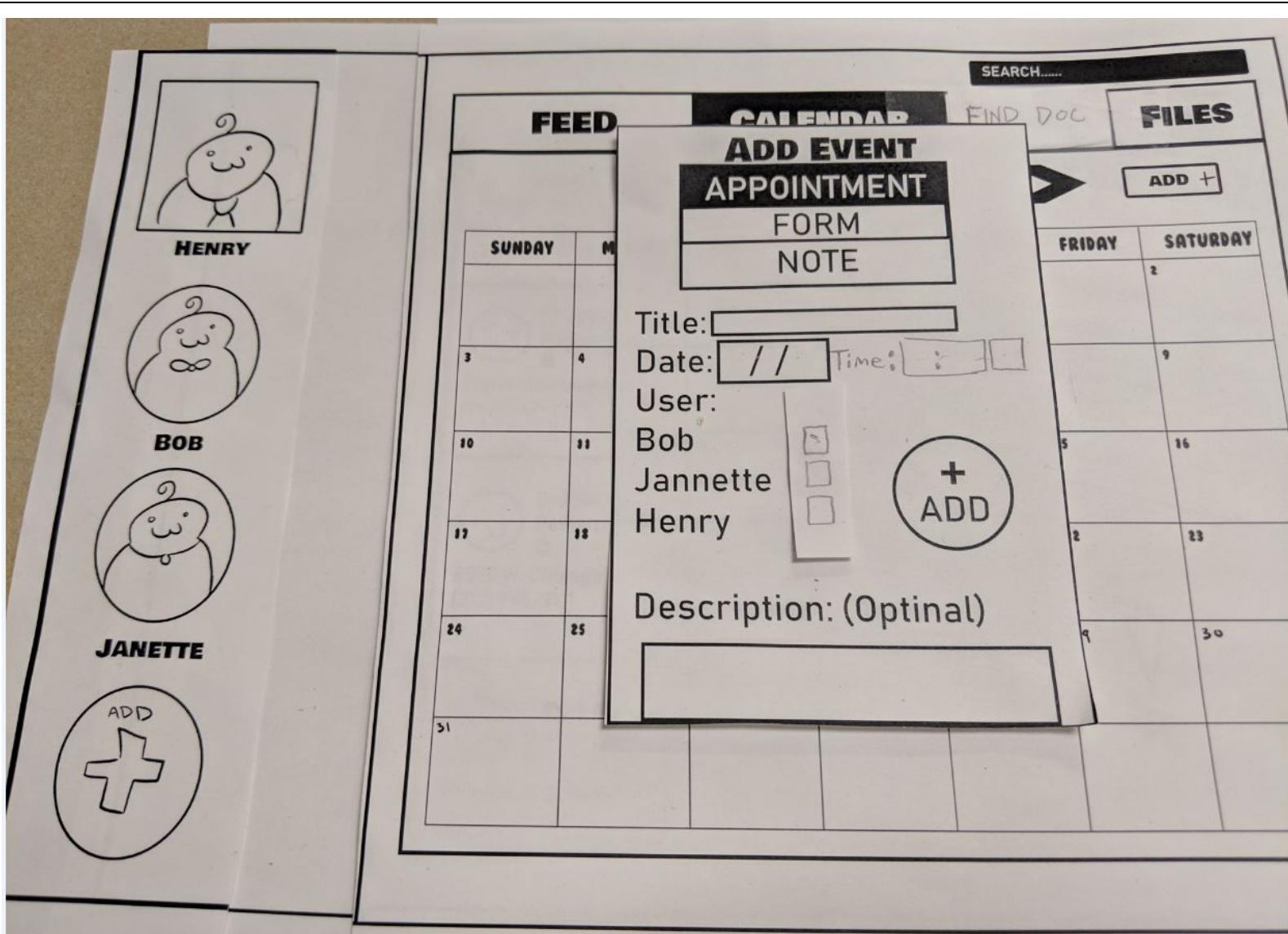


JANETTE



CALENDAR						
FEED		CALENDAR			FILES	
MARCH						
SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Calendar Tab. Events can be added via the "Add+" button, or directly manipulating the calendar. Calendar displays "appointment" events by month.



Clicking on the “Add+” button brings up a dialogue to add a new appointment reminder, note, or upload a file, which is done by selecting from the list. When the user is in the calendar tab, the type defaults to appointment.

SEARCH.....

FEED **CALENDAR** **FIND DOC** **FILES**

RECENT FILES:

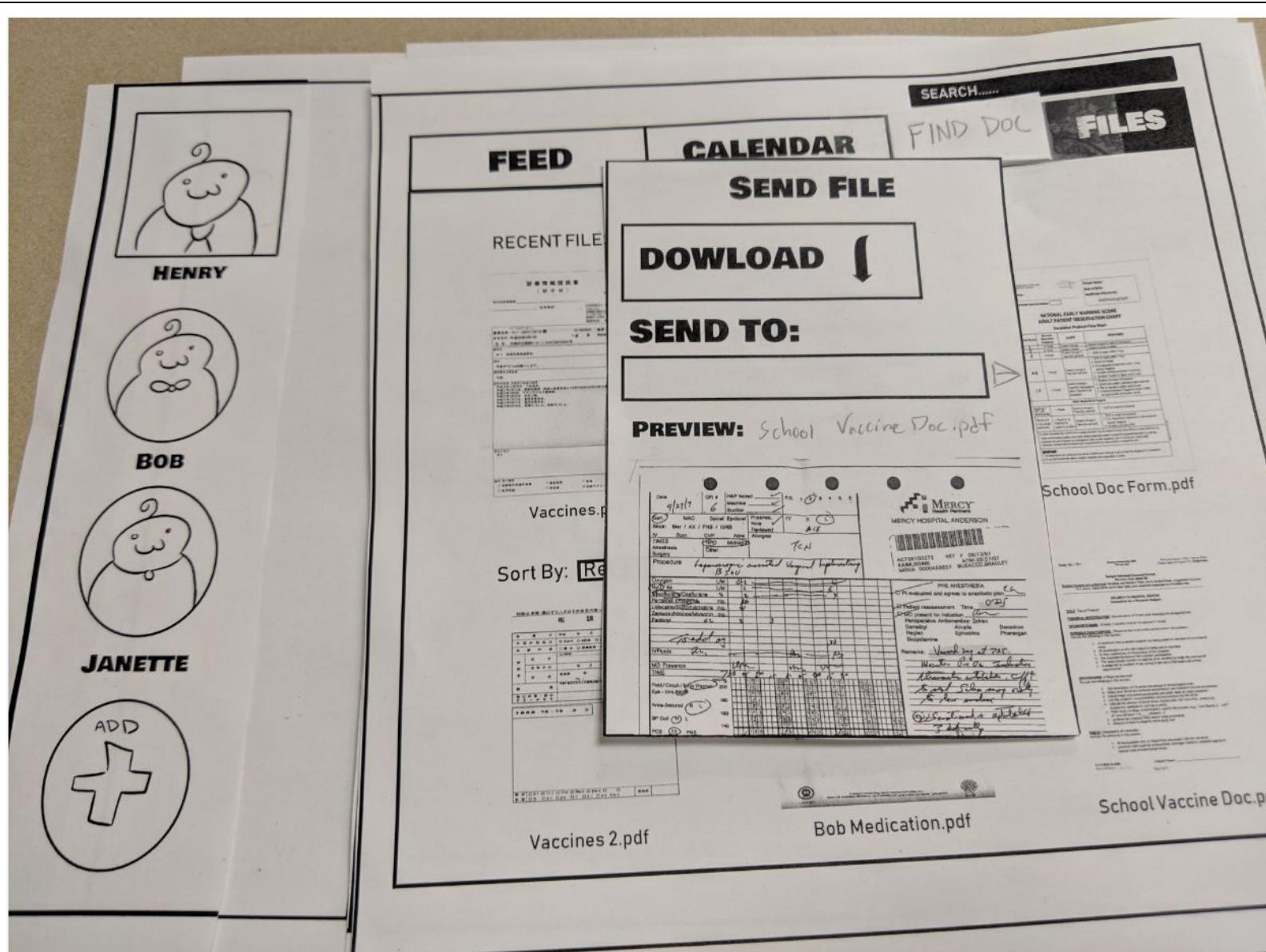
Vaccines.pdf Medication.pdf School Doc Form.pdf

School Vaccine Doc.pdf

Sort By: **Recent**

Vaccines 2.pdf Bob Medication.pdf

“Files” tab that lets the user look at recent files.



Clicking on a file tab brings up a popup view that shows a preview of the document, and allows the user to select a doctor or school that was previously saved, and forward the file to them.

Briefing:

DocDoc is an application that organizes all of your medical needs for your children. By keeping track of their doctors and documents, DocDoc allows you to track appointments, keep documents updated, and easily send them to your child's doctor or school.

Users can view a feed of all recent uploads and upcoming events, a list of all their medical documents, and a calendar. Users can also search for nearby doctors using our map feature.

By selecting a specific dependent, a user can filter their documents and events as well as view and edit specific information for that dependent.

DocDoc is best suited for parents who need help keeping track of and organizing all medical-related aspects of their children. Users must have basic computer skills for using our site and uploading relevant documents.

Thank you for participating in our user study. The objective of this study is to help improve our design so users are able to accomplish their goals easily, not to test your ability to use our application.

Scenario Tasks:

1. Find a nearby Doctor's contact information (Dr. Realman) to schedule an appointment for Bob
2. Save the details for your appointment, which is on March 8th, for Bob into DocDoc
3. Send School Vaccine Doc form to Bob's Middle school

Observations:

During testing there were several users who had issues with the first task of adding a doctor's contact information. Naming the tab is an issue, since "Map" didn't clearly imply that the tab is intended for looking up doctors. In the second iteration, there were fewer issues after renaming, but one user did initially think that "find doc" was used for searching documents, and tried to navigate to it when looking up a form.

The interface for finding and adding a doctor caused some confusion for the patients. Many had trouble identifying the 'closest' doctor, as it wasn't clear how the results were ordered. One user also noted that it was unclear what the "add" button did, and was too similar to the "add +" symbol on the sidebar for adding accounts.

After testing several users voiced opinions that feedback for their actions would be appropriate. One user noted that there was not notable feedback to show that their form was sent after they left the sending dialogue.

During testing, there were two ways of getting to the "add doctor" feed. However, almost every user went to the "map/find doc" tab to accomplish the task. One user selected Bob on the sidebar before navigating to the "map" tab, but didn't understand that clicking on the blank doctor field navigated to the 'find doctor' interface. A similar issue occurred when sending forms, where the users usually stalled briefly looking for the "send" icon. These both suggest that "button" inputs need a clear, visible way to annotate that they can be interacted with, as well as their functions.

Interestingly, the task with the fewest issues was adding something to the calendar. Mostly every user was able to navigate to the calendar directly. Most also tried to directly click on the date, and only one user clicked on the "Add+" button at the top of the Calendar tab.

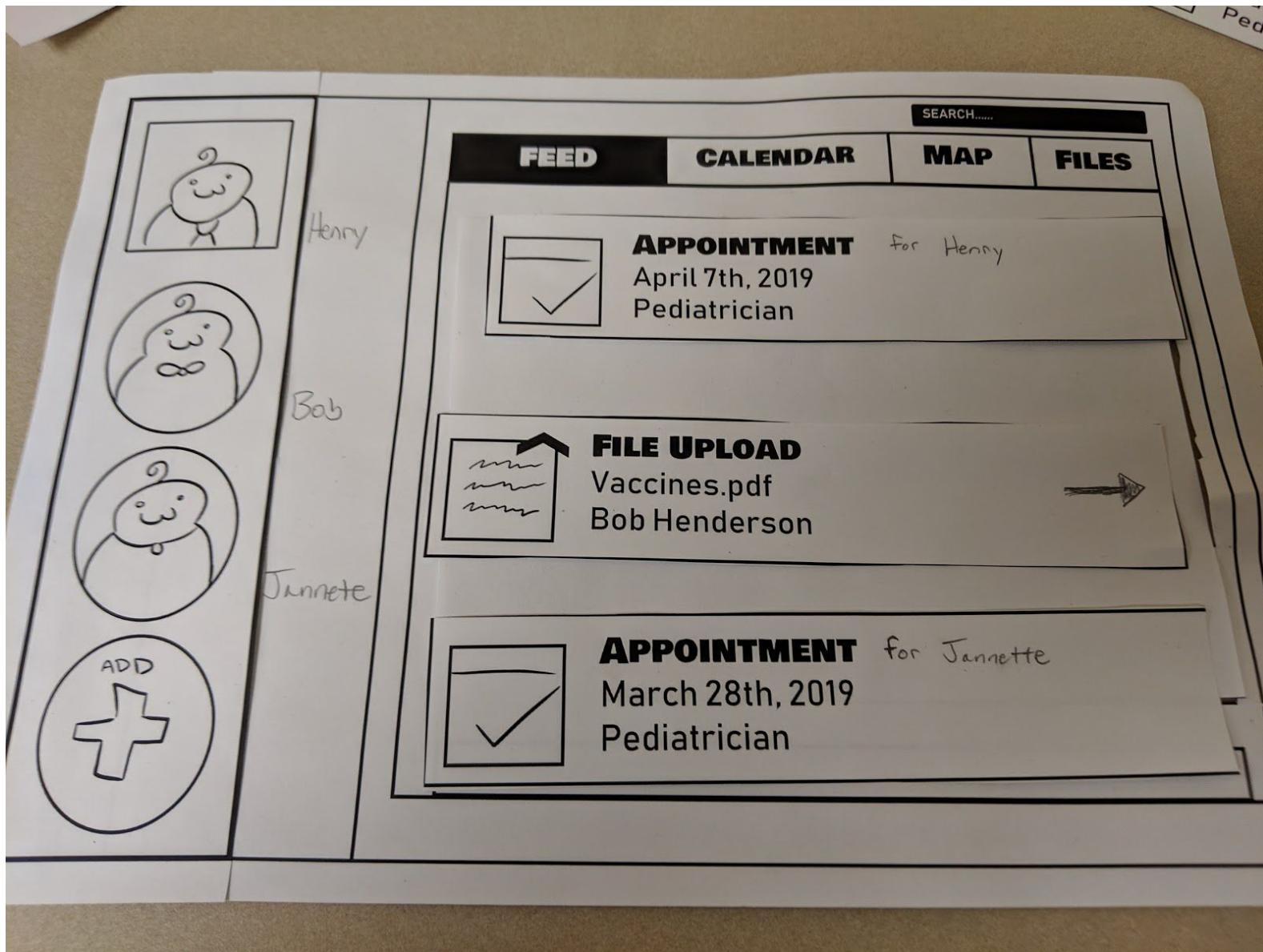
Finally, a safety vs efficiency design consideration was brought up by one user. When task with sending a file, the user erroneously sent a file with a similar, but different, name as the intended file. This was likely due in part by the fact that this file was in the feed where the user accessed it, so he didn't have the correct file there for reference. This brings up a safety issue - do we need to design the system so that all files are forced to be given specific labels and tags, making it safer, but more cumbersome, or do we simply let the user name the files themselves, leaving more room for slips?

Prototype Iteration:

Iteration 0: [link](#)

Iteration 1: [link](#)

Images



Feed page with Dependent list to left, Feed items moveable depending on which dependent is selected.

mediat

BOB HENDERSON

Info

Allergies:
Peanuts

Medication:
Blue Pills

Age: 14
Weight: 150
Blood Type: O+

Doctor Info

Empty

School Info

Middle School Jr. high
742 Academia Ln.
Email!
Nurs@e @ MSSH.edu

SEARCH.....

FEED CALENDAR MAP FILES

FILE UPLOAD

Vaccines.pdf
Bob Henderson

Clicking on Bob opens his scrollable info page. Feed/calendar would be updated to show only information for Bob.



Henry



Bob



Jannete



ADD

Nearby Pediatricians

SEARCH.....

FEED

CALENDAR

MAP

FILES

A Dr. Realman
Pediatrician

420 Lake Shore Dr.
(312) 555-5555

B Dr. Fakerman
Pediatrician

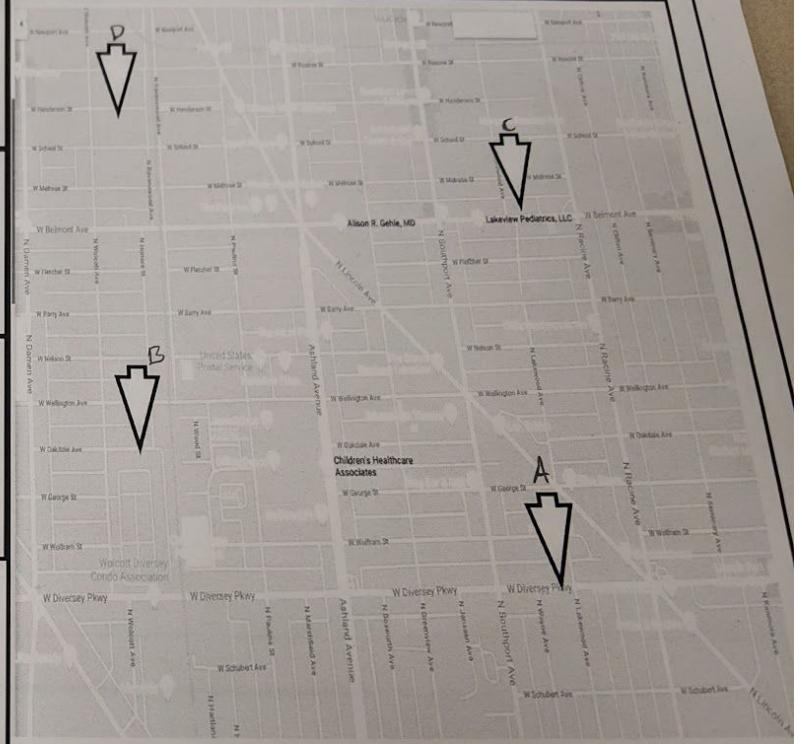
550 N Mountain Rd.
(312) 542-9999

C Dr. Feelsbad
Pediatrician

8520 W. Chicago Blvd.
(312) 995-9111

D Dr. Gandalf
Pediatrician

7892 E. Forsure Rd.
(312) 911-8989



Map view currently automatically looks for nearest doctors based on user location, can be opened with either all or specific dependent selected.

Henry

Bob

Jannete

ADD

Nearby Pediatrician

FEED	CALENDAR	MAP	FILES
Dr. Realman Pediatrician 420 Lake Shore Dr. (312) 555-5555	Dr. Fakerman Pediatrician 550 N Mountain Rd. (312) 542-9999	Dr. Feelsbad Pediatrician 8520 W. Chicago Blvd. (312) 995-9111	Dr. Gandalf Pediatrician 7892 E. Forsure Rd. (312) 911-8989
Dr. Realman Phd. Pediatrician 420 Lake Shore Dr. (312) 555-5555 email@faker.com			

Clicking on Dr. Realman opens his info as well as option to add to a profile.

SEARCH.....

FEED CALENDAR MAP FILES

< MARCH > ADD +

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Appointment for Janette

Henry

Bob

Janette

ADD

Calendar view shows current appointments (or added appointment), reflects information based on selection of dependents.

SEARCH.....

FEED **CALENDAR** **MAP** **FILES**

RECENT FILES:

Sort By: [Recent]

Vaccines.pdf

Medication.pdf

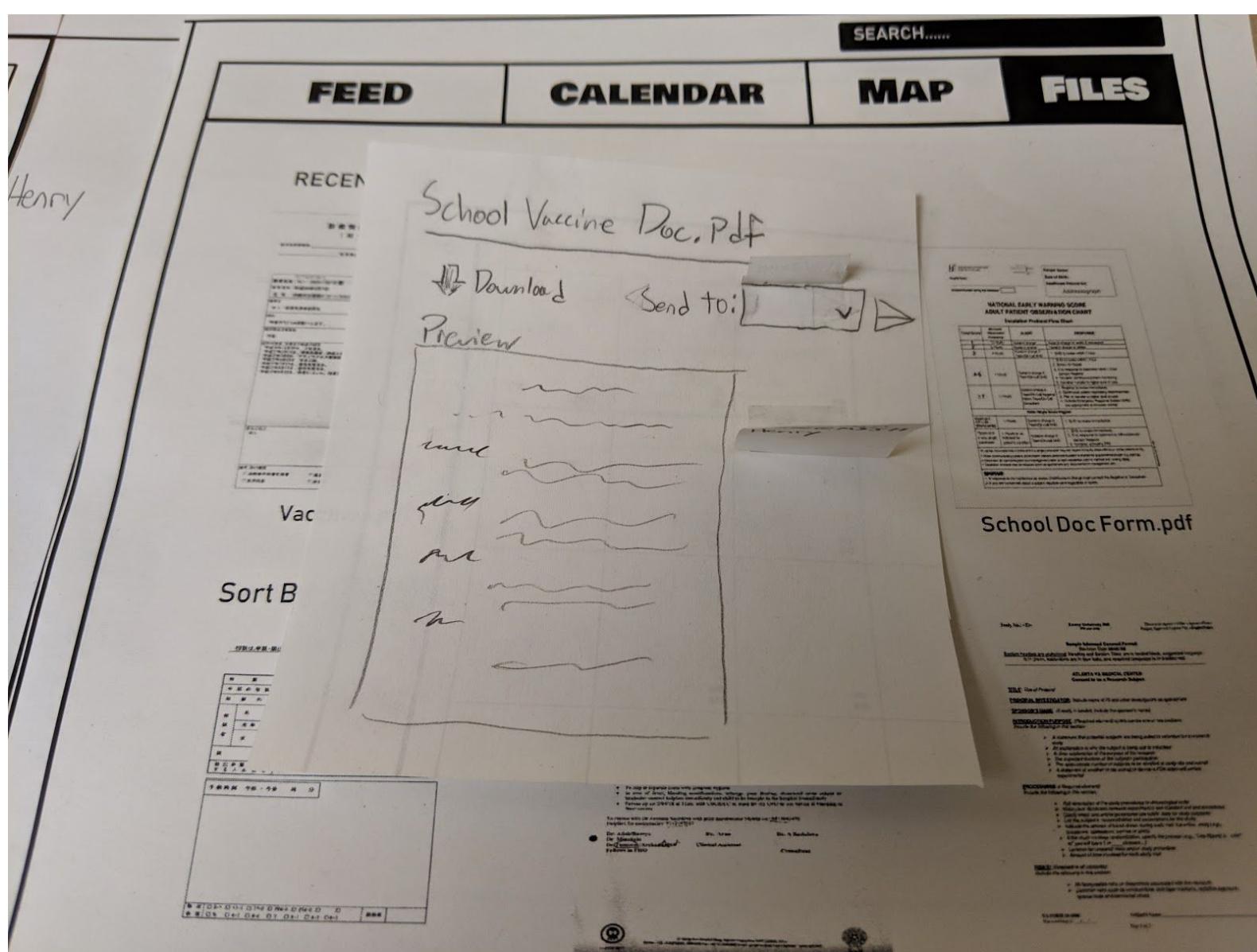
School Doc Form.pdf

Vaccines 2.pdf

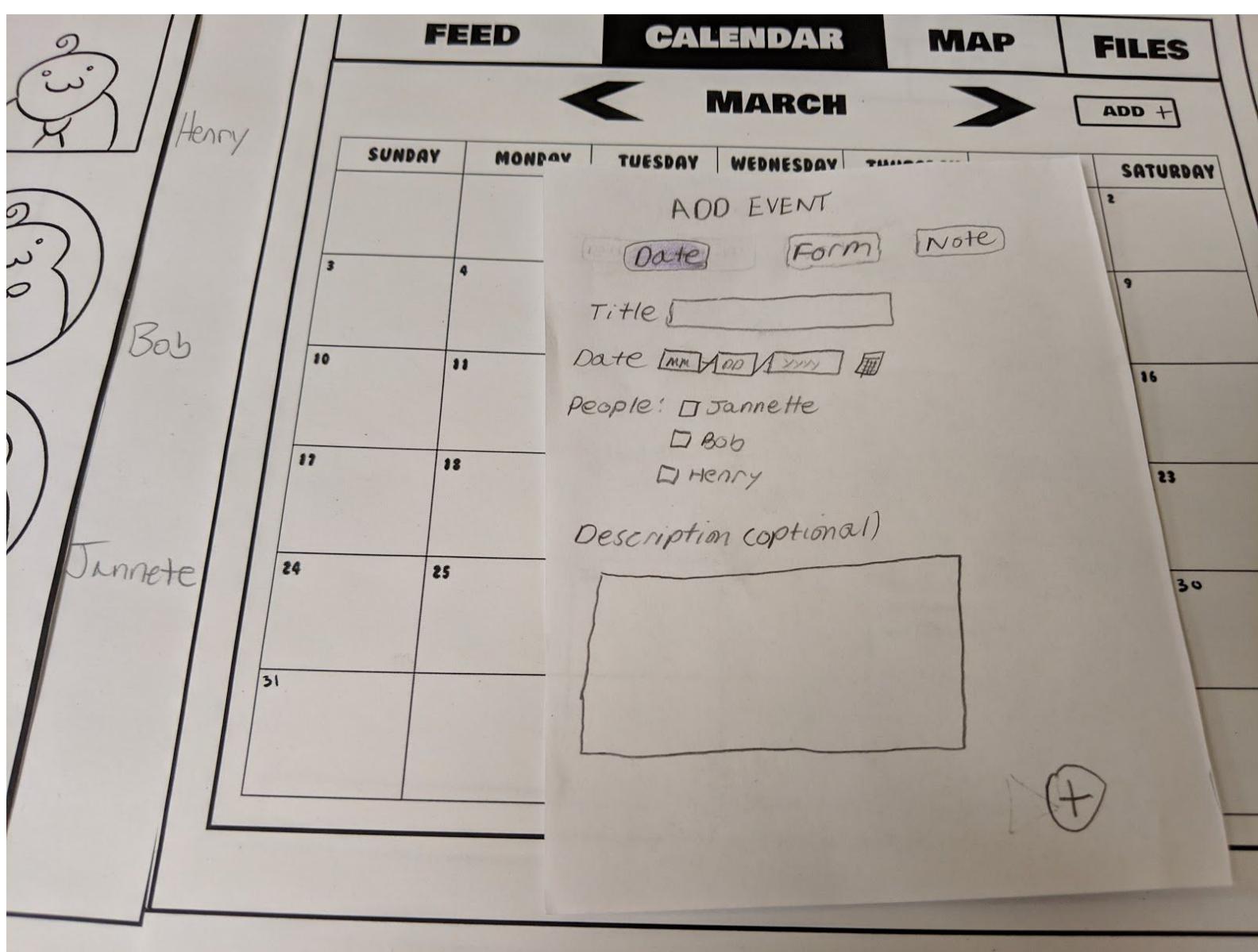
Bob Medication.pdf

School Vaccine Doc.pdf

File page shows files...Currently doesn't show association of files to any particular user/isn't physically dynamic depending on dependent selection



Hard-coded popup menu shows school vaccine doc.pdf selected, with options to pick who to sent it to.



Adding event popup over calendar, giving option for title, date, and people to send message to

Task 1: Find nearest doctor
for Bob (Dr. Realman)

Doctor Realman
Added to
Bob
OK

Task 2: Add reminder for
Appointment in Calendar
for March 8, 2019 @ 11:30AM

Event Added
OK

Task 3: Send "School Vaccine Doc.pdf"
form to Bob's
Middle School

Message Sent!
OK

List of tasks and popup messages to give feedback on completing an action

Outcomes

Iteration 1 was tested on 3 users. Key issues identified for improvement in iteration 2 included:

- Changing the “Map” tab to “Find”, to make it more intuitive.
- Adding short annotations to the ‘add’ symbol under the patient info tab so that it is more self-explanatory.
- Changing the “add” feature for doctors/schools so that the associated dependents are linked to them via a multiple-selection menu that is pre-populated if a person is selected in the sidebar.
- Add more descriptive placeholder text to the school/doctor fields on the user information section so it’s more clear what those are for.

- Add a symbol to denote where the user is located on the map in the “Find” tab.

Iteration 2:

Images (external links)

- [Base Files](#)
- [Paper Images](#)

Outcomes

Iteration 2 was tested on 3 users. Issues and associated design changes were:

- Change the “+” symbol in the “find a doctor” tab to “Save”, since the “add” was too similar to adding a account.
- Rename “find doc” to “find nearby”, so it’s not confused with a document search, and can include schools as well.
- On the “Send/Preview Files” popup, change the send icon so it’s more obvious and explicitly says send, since users had trouble identifying it.
- Show distances to the user’s current location in the “Find Nearby” Tab for the search results.
- Add “action” in the news feed for changes to the account such as sending a file, or adding a doctor. These actions, which can’t be directly manipulated (and thus aren’t clickable) should be stylistically different.
- Add action button with annotations to “clickable” items in the Feed tab, denoting what actions are available. These would be “view” for Appointments and Forms, “edit” for Appointments, and “forward/send” for Files.

GR4 - Computer Prototyping:

Platform Details:

Users should have a modern web browser (preferably Google Chrome) and a desktop computer (preferably with a large screen).

Instructions:

Go To: <https://tehwentzel.github.io/DocDoc/>

Source Code at: <https://github.com/tehwentzel/DocDoc>

If there are any issues with accessing the interface, one of us can be reached at:
awentze2@uic.edu

Shallow Parts:

- Editing user data hasn’t been implemented for the following features:
 - User data isn’t editable and pre-populated.
 - New accounts can’t be added.

- Adding a calendar event will show a confirmation but not show up in the calendar or feed.
- The ‘Find’ tab has a static map image and a pre-populated list of doctors.
- The ‘Search’ in the navbar does not fetch any results.
- The feed isn’t updated dynamically
- Events in the feed don’t bring a popup.
- The popup for selecting a file is fixed.
- Doctor data is always added for both users

GR5:

Platform Details:

Users should have a modern web browser (preferably Google Chrome) and a desktop computer (preferably with a large screen).

Links:

Go To: <https://tehwentzel.github.io/DocDoc/>

Source Code at: <https://github.com/tehwentzel/DocDoc>

General Feedback:

Based on feedback from the project discussions and GR4 grade comments, we decided to change the color-scheme of the interface. We moved the main colors to a less saturated teal from red. Additionally, we introduced colors for each user, so that different items in the right-view could be easily mapped to the individual users, making the linking between the two views more apparent.

Individual Contribution:

Andrew (Profile):

My main area was the profile view on the left of the page, with all associated functionality, as well as the navbar on the top of the menu. This included the overall page layout and the functionality allowing the different tab views to change, and some filtering in the other views when a user was selected. I also implemented code that allowed for new accounts to be added and tracked, and that feed events were automatically populated based on what’s in the calendar, so new events could be added. I also worked on merging the other components into the overall layout and keeping some design details consistent.

Most of the feedback related to the profile bar (on the left hand side) had to do with the mapping from the active accounts to the main feed. To help this, colors were added to each user’s name so that they can be color-coded to map to items in the other windows. Filters we put in so that selecting a user would cause only items specific to each user to show up in the feed or files tabs. Finally, some style changes were made to the profile and top-tab to better follow the lecture on visual affordances.

- Changed ‘add doctor/school’ buttons to ‘find doctor/school’ - which appear when a user doesn’t have a school or doctor assigned to them in their profile data - and styled them to look like actual buttons (consistency).

- Standardized all button styling for internal consistency.
- Added headers for the Feed and Files tabs with a dummy ‘sort-by’ button.
- Changed navbar so that the selected tab is colored and underlined, instead of given a different background (to prevent giving the affordance of a button).
- Nav-bar’s tabs’ ‘hover’ event changed so that the text size increases, rather than show a colored border.
- Users have a default color assigned to them which is underlined in their name
- All alerts were changes to be in the center and larger using the sweetalert api, rather than the default alert.
- Clicking on an avatar is disabled once it’s already been clicked on (safety).
- Added ‘add account’ to the bottom of the add account button (plus sign on the left bar).

Edward (Feed):

My work revolved around the feed for DocDoc. Originally I had made the feed too wide and used colors which didn’t really represent anything. Later I had proceeded to fix the style and presentation of the feed information to match the overall aesthetic which our website was aiming for after looking over heuristic evaluations provided for our group.

The feedback I received was in regards to the style and layout of the feed. At first the user feedback was that the feed was unintuitive and a bit confusing. Through my process I looked at what would look better, talking it over with my group and ending with a design that matched the website aesthetic as well as provide much more intuitive user feedback to each feed item.

- Created the layout and items in the feed
- Filled in appropriate description and information for the feed items
- Color coded the feed outline according to user
- Placed icons along side the feed to represent file type
- Matched the feed hover function to maintain consistency with overall UI design
- Made changes to the original design based off heuristic evaluation
- Reworked the sizing and containment of the material within feed items to be more presentable
- Feed items maintain design when filtered for each user
- Scroll bar added to prevent items from not showing and overflowing without proper correction
- Events in the feed now contain descriptions and the titles of the events

Shantanu (Calendar):

My main work was to design and program the calendar tab. The tab displays saved events and lets the user add new events. I received mostly positive feedback for the calendar during the heuristic evaluation. Some changes were incorporated after the heuristic evaluation and studio session with the instructor in terms of internal consistency. My tasks included:

- Designing a simple and easy-to-use calendar with hover effects and click functions.
- Designing modal to show a saved event.
- Designing a modal to add an event.
- Creating a form to add an event.
- Using HTML local storage to get the user's event information and fetch it when needed.

- Creating a simple JSON store to maintain the demo events information.
- Using white space instead of grid lines to separate the days in the calendar. This was done keeping in mind the Gestalt Principle that we studied in Layouts.
- Following the instructor's suggestion of removing the close button from the modals, added a functionality to close the modal when you click anywhere on the screen apart from the modal itself. Thus, making the target for the user very big (Fitts law).
- Used colored bars to denote what profile the event belongs to instead of making the user click on the event to know the profile details.
- Added a functionality to read the date automatically, when a user clicks on it to add a new event, thereby reducing the number of clicks needed by the user to save the date in a new event.

Ryan (Find Nearby tab):

My task was to work on the “Find Nearby” tab, which is responsible for showing the user a list of nearby pediatricians accompanied with a map that shows their location relative to the user. The main feature of this tab is being able to quickly find a doctor and save their contact information to a particular user’s profile. I initially had an image to represent the map and had basic functionality that assigned the doctor’s information to the user.

The main issue brought up from the discussion for the Find Nearby tab was the amount of clicks needed to perform the task of adding a doctor to a particular user. More generally, there was another issue of connecting the state of the sidebar with the rest of the tabs.

Feedback from the Heuristic evaluations mostly focused on whitespace and a lack of an interactable, clearly-labeled map; below are the changes made based on the feedback:

- Doctor information box
 - “Save” button switched to “assign” to minimize confusion
 - “Assign” button responds to state of user selection window
 - No user selected:
“Assign to...” button opens up window to select hard-coded Bob or Jannette buttons
 - User selected:
“Assign to [username]” button adds doctor info to that user immediately
 - Removed drop-down checkbox and confirm button to reduce redundancies
 - Before:
(Bob -> add doctor -> select doctor -> save -> select Bob -> confirm)
Or
(Find nearby -> select doctor -> save -> select Bob -> confirm)
 - After:
(Bob -> add doctor -> select doctor -> Assign to Bob)
or
(Find nearby -> select doctor -> Assign to Bob)
 - Added letters to match marker labels on map
- Map area

- No longer a fixed picture, using google API
 - Added hard-coded custom markers that respond to clicks like the doctor boxes
 - Removed a lot of default settings for google maps to focus the user on the task of finding nearby pediatricians
- Other
 - Added a header labeled “Nearby Pediatricians” to give the user information about the tab.
 - Removed ‘x’ (close) button on information boxes, click outside of window to close
 - Safety - Popups for doctor already added and replace doctor
 - Fixed boot-strapped column layout to reduce wasted white space

Jon-Michael (Files):

My job was to work on the files tab, which is responsible for containing the user’s documents for both themselves and their dependents. Not only is it to store documents, users are able to send them to specific doctors and/or institutions.

The feedback that I have received in regards to the files tab was mixed after reading through the heuristic evaluations and hearing feedback from the instructor and TAs. As a result, changes were applied and some tasks included:

- Displaying a list of arbitrary, yet related, documents with a small view of the respective document.
- Worked on the organization of the items listed using things such as margins, padding, and borders around the file images.
- Designing a modal that contained a download button in cases where the user needs to download a copy from the application, an input field to allow the user to send the selected document over to various things, and a full size document viewer that enabled the user to see and clearly read the document and its contents.
- Initially had it where the user had to click the **X** in the top right of the modal to close it but was later deprecated in favor of click-outside-the-modal-to-close-it.
- Initially had a large download button which was later scaled down in size.
- Following heuristic evaluations, made a small text overlay utilizing the **img**’s **alt** to display the name of the document.
- Designed the overlay, the modal, and the list view to match the theme of the whole application
- Added in an *fade in from top* animation that, when clicked, does what it sounds like. This is implemented as a result of the other tabs having similar animations when certain elements are clicked by the user.
- Following heuristic evaluations, removed the second **div** which contained pretty much the same thing as the first one. Pretty much got rid of the redundancy that was noted.

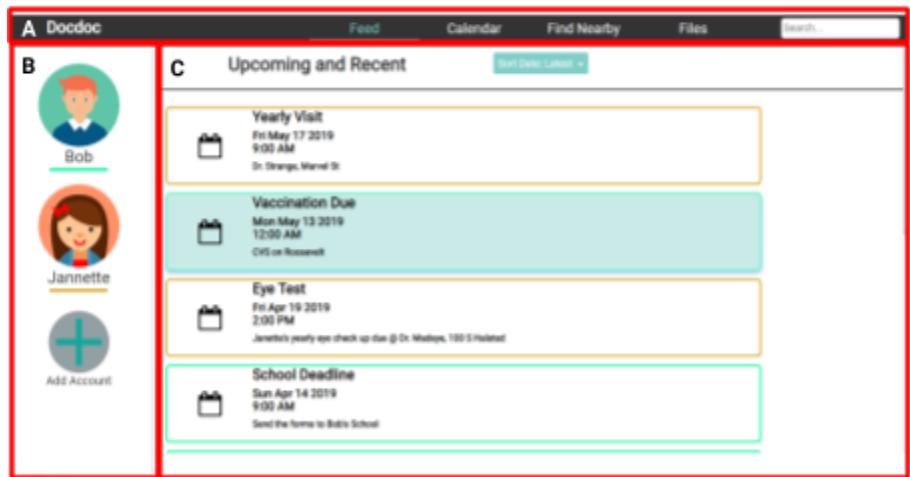
GR6 - User Testing:

Design:

Layout:

For our design we divided up the layout into 2 sections: a persistent bar that showed the account settings, and separate view that showed one of four different tabs that were selected by the user. The tabs were designed around the user tasks identified in our scenario: a calendar for scheduling appointments, a map-view for finding nearby doctors, and a tab for viewing user files.

Finally, we made the first tab a feed-view, which was designed as a way to provide a quick overview of the most relevant information from the other tabs. The tabs could be selected via a nav bar. Since only part of the layout changes when navigating the tabs, the app is designed as a single page, with only the sub-views changing.



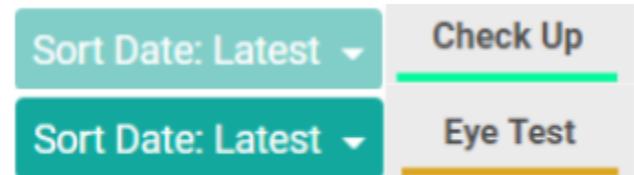
Layout of the interface. (A) The navbar with different tab features and a search bar. Selecting a tab changes the view in C. (B) The accounts for each dependent of the user. Selecting a user changes the view to provide details for the selected account. (C) The active view for the given tab (Feed, Calendar, Find Nearby, or Files).

Styling:

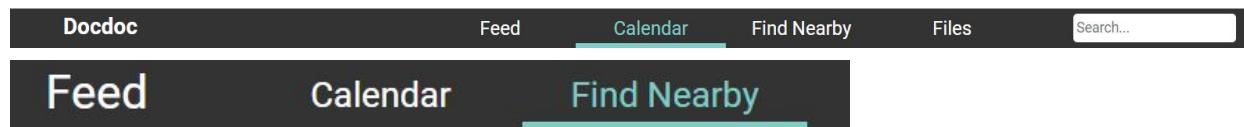
For our color palette we went with two different shades of teal for buttons and some other styled components, alongside basic white and black for the background. These colors were chosen to mirror the color scheme used in the UI Health website. Items that were tied to a specific user were given user-specific colors, with green and orange being assigned to the two default users, and new users getting a random color. This color-code was used in lieu of explicit labels as color is a preattentive visual cue that users can process more quickly than labels or other types of annotation, providing better information scent.

All fonts used were Roboto for visual consistency. Tab-headers and Titles were in bold. All text that wasn't clickable was in black, and clickable text, save for dropdown menus, (e.g. buttons and tabs) were in white, to give better contrast as they were given colored backgrounds. The selected tab is colored teal to show that it was selected.

Nav Bar:



Examples of the color palette as seen in inactive (top left) vs active (bottom left) buttons, as well as user colors for Bob (top right) and Jannette (bottom right).



The nav bar was designed so that the app title (Docdoc) was located over the account bar. Tab buttons were shown on the right over the area of the tabbed view. The Selected tab is shown in the light-teal color used in our color palette, and underlined. To provide visual feedback, hovering over an inactive tab will cause the size to increase (Feed in the picture).

Accounts:

The screenshot shows the 'Accounts' section of the Docdoc app. On the left, there's a list of users: Bob (active, green background) and Jannette (inactive, orange background). Below them is an 'Add Account' button with a plus sign icon. In the center, there's a detailed profile for Jannette, which includes an avatar, name, and various information cards.

Info	
Allergies:	Sandwich Crusts
Medication:	Smuckers Uncrustables
Age:	12
Weight:	120
Blood Type:	O-

Doctor Info	
Find A Doctor	

School Info	
Name:	College Elementary School
Number:	834-432-4333
Email:	mam please dont email us
Address:	445 fake st.

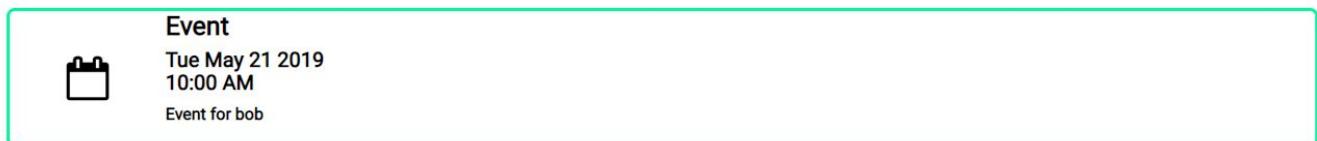
Accounts for each dependent are shown in a bar on the left side of the window. As this window serves as an indicator of the state of the system (active user), and provides account information on the account informations that assists in user recall when remembering information when filling out forms or identifying color-coded items in other tabs. The buttons also serve double-duty as user filters for the files and feed tabs.

Accounts are shown as circular avatars with a picture for each user, to be consistent with other account-based applications. The name and a user-specific color are marked below the avatar. Finally, an “Add Account” button is placed at the bottom of the menu. The button was styled to be similar to the user accounts for consistency, as well as to suggest that the account icons are also clickable, as they have the same affordances as the ‘Add account’ button.

Selecting a user moves that user to the top of the menu, and removes other users. Bootstrap cards are used to separate personal information, with information pertaining to the doctor and school assigned to the given patient. When a user isn’t assigned a doctor (or school), a button is placed in the center of the menu, to improve efficiency and suggest to the user that there are ways of adding a doctor.

Clicking on the add account brings up a modal in the center of the screen with a form to fill out for user account information. Examples are given for each of the form's fields. For simplicity, the user is limited to selecting for 3 different icons for their user avatar. Finally, for security, the submit button ensures that the name and icon fields are filled out, as that is the minimum required information to create a new account.

Feed:



The feed keeps track of the users latest activity. There are two kinds of feed items, events and files. In each feed item there is a title and information. Above we see an event feed item, this is depicted by a calendar icon, allowing for our users to easily recognize that this is an event at a glance. In the information section of an event, the day that event is scheduled for appears along with some description of that event.



Depicted above is a File feed item, here the user is able to view a file by clicking the feed item. The feed file is recognized by our users through a generic paper file icon since the types of files stored are images of documents. Here we can also see the name of the file as the title of the feed item, and the date the file was added.



This image shows us that feed items are color coated. The color corresponds to the user, and can be seen on the sidebar of the application appearing under the users icon. This allows for users to easily distinguish between files and their respective users.

	Event
	Tue May 21 2019 10:00 AM Event for bob
	Vaccination Due
	Mon May 13 2019 12:00 AM CVS on Roosevelt
	School Deadline
	Sun Apr 14 2019 9:00 AM Send the forms to Bob's School
	Check Up
	Wed Apr 10 2019 10:00 AM Dr.Realman, 420 Lakshore Dr.
	File Added: Bob Vaccines.pdf
	March 3rd, 2019: BOB
	File Added: Bob School Forms.pdf

Here we see the feed can be filtered to show one specific user's files. This comes in handy for organization and efficiency when using the application because it helps our users sort through items faster. This is done by clicking on a users icon, from there items in the feed are sorted accordingly.

Upcoming and Recent

Sort Date: Latest ▾

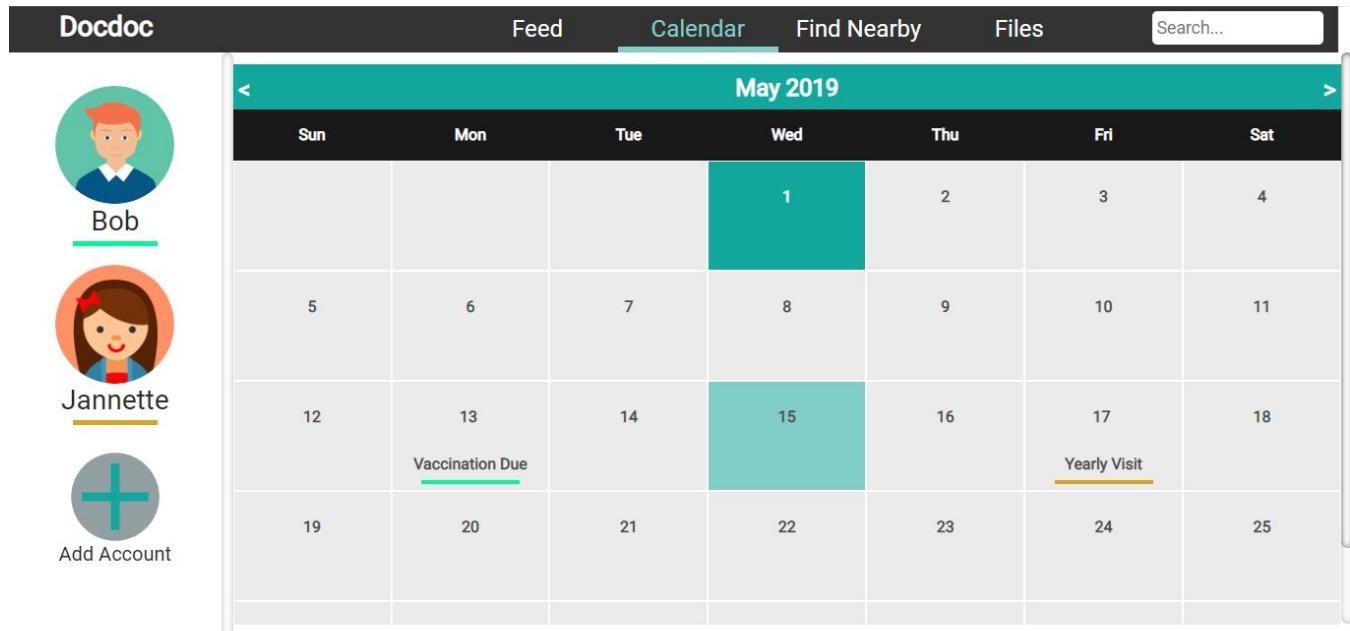
Furthermore, the feed contains more sorting efficiency and organization through allowing our users to filter items by latest or earliest.



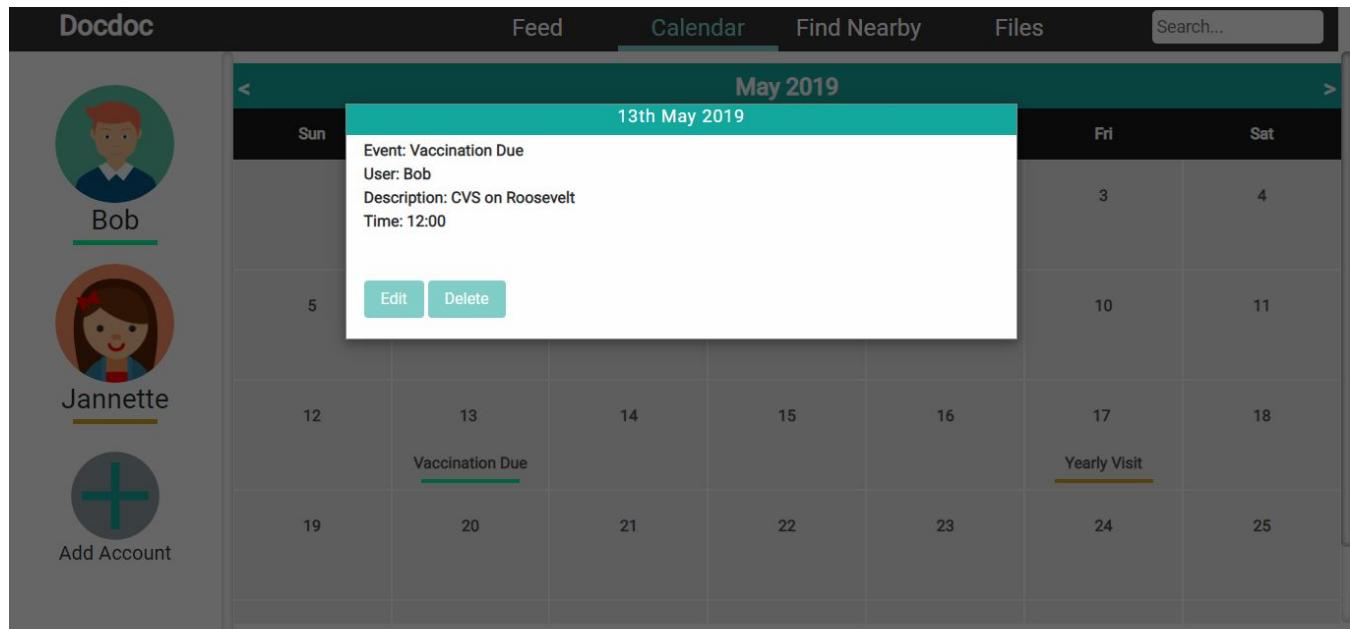
File Added: Bob Vaccines.pdf
March 3rd, 2019: BOB

When hovering over a feed item, the background turn a light blue color to display to our users that an item is clickable, after clicking an item, a modal will appear allowing the user to take quick actions without having to navigate to other tabs.

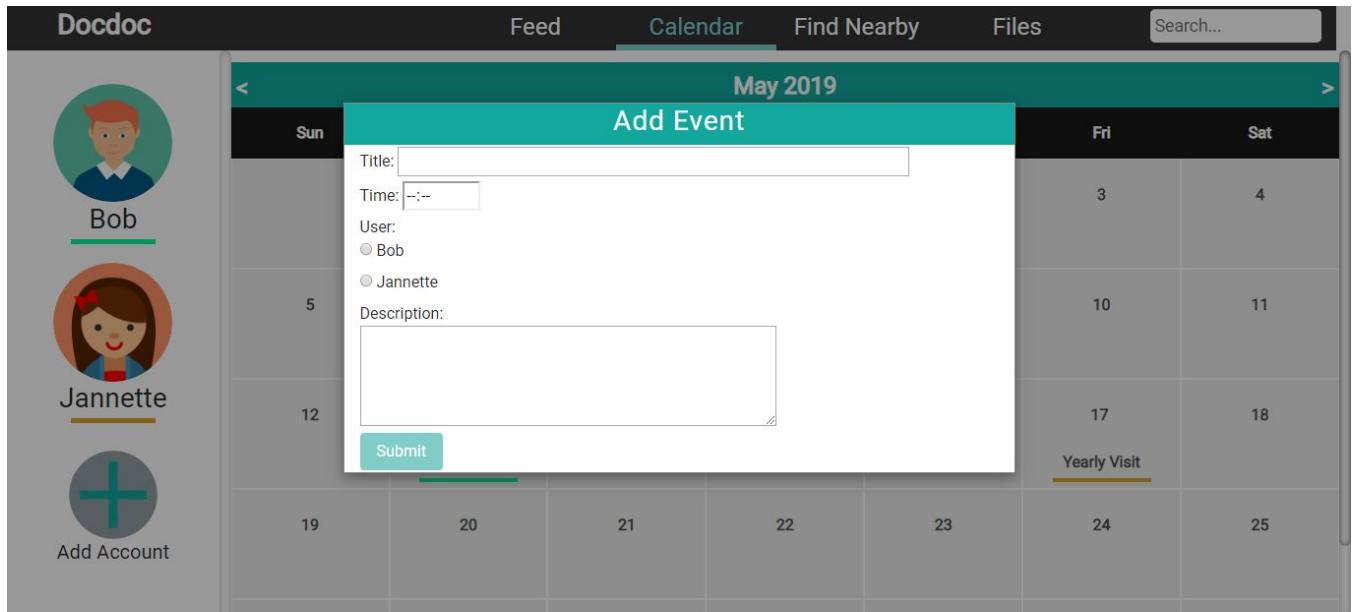
Calendar:



The calendar has a simple design showing the current month by default. Events are matched to users using a colored bar under the event title, providing easy visual cues for which events are for each dependent, in order to improve information scent for the user. White spaces are used instead of grid lines to separate the days in the calendar following the Gestalt Principle.



The user can click on an event to view more details about it, this information is displayed by using a modal. The modal contains the date as the header. The event title, user information, description, and time are displayed below it.



If a user clicks on an empty date, they can add a new event. A modal which contains a form pops up and prompts the user to fill in the details of the new event. It automatically reads the date the user clicked and saves it as the event date. When the user submits the information, the event appears both in the calendar and in the feed.

The modals were designed to be closed by clicking anywhere on the screen except the modal itself to maintain internal consistency with the other popup forms in the system. Allowing the user to click anywhere offscreen instead of using a close button improves efficiency by making the target effectively infinite (Fitts' Law).

Find Nearby:

Docdoc Feed Calendar Find Nearby Files Search...

Nearby Pediatricians

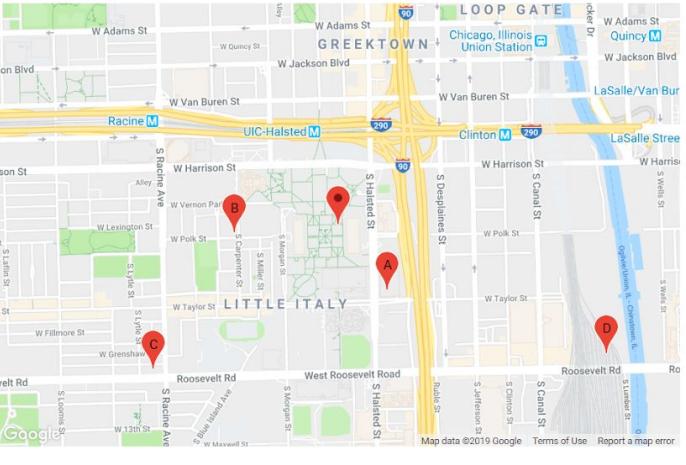


Bob

Jannette

Add Account

	Dr. Realman Pediatrician 420 Lakeshore Dr.	A
	Dr. Octavia Pediatrician 628 Spider Lane	B
	Dr. Strange Pediatrician 221b Baker St.	C
	Dr. Doctor Pediatrician 301 Doctor Dr.	D



The Find Nearby streamlines the process of finding a nearby pediatrician for the user by automatically showing google maps results of nearby pediatricians. Information sent is used here by stripping down some of the unnecessary map interface, leaving only the relevant pediatrician and user locations on the map.

Docdoc Feed Calendar Find Nearby Files Search...

Nearby Pediatricians



Bob

Jannette

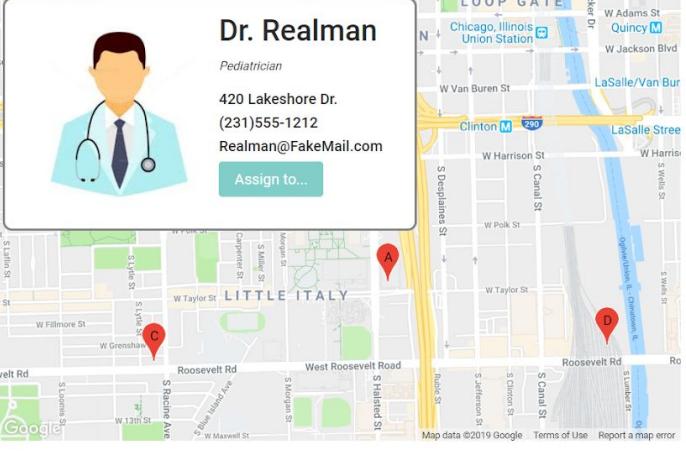
Add Account

	Dr. Realman Pediatrician 420 Lakeshore Dr.	A
	Dr. Octavia Pediatrician 628 Spider Lane	B
	Dr. Strange Pediatrician 221b Baker St.	C
	Dr. Doctor Pediatrician 301 Doctor Dr.	D



Dr. Realman
Pediatrician
420 Lakeshore Dr.
(231)555-1212
Realman@FakeMail.com

Assign to...



When selecting a doctor, there is an assign button that will automatically populate the account info with the doctor's contact information. If no account is currently selected, clicking the assign button will then ask the user which account the information should be added to.

The screenshot shows the Docdoc mobile application. At the top, there is a navigation bar with tabs for "Docdoc", "Feed", "Calendar", "Find Nearby" (which is highlighted in teal), and "Files". A search bar is located at the top right. On the left, there is a sidebar for "Bob" (represented by a green circular icon) containing sections for "Info", "Doctor Info", and "School Info". The "Info" section lists allergies (Pollen, Bees), medication (PollenBeeGone), age (13), weight (140), and blood type (A+). The "Doctor Info" section has a "Find A Doctor" button. The "School Info" section lists "Name: Middlest High School". The main content area is titled "Nearby Pediatricians" and displays four doctor profiles: Dr. Realman (A), Dr. Octavia (B), Dr. Strange (C), and Dr. Doctor (D). Profile A is expanded, showing Dr. Realman's details: Pediatrician, 420 Lakeshore Dr., (231)555-1212, Realman@FakeMail.com, and an "Assign to Bob" button. To the right of the profiles is a map of Chicago's Loop area, specifically Little Italy, with red pins marking the locations of the four doctors. The map includes labels for streets like W Adams St, Quincy M, W Jackson Blvd, LaSalle/Van Bur, Clinton M, W Harrison St, S Clark St, W Polk St, W Taylor St, Roosevelt Rd, and S Dearborn St.

If an account currently is selected, then the button will automatically add the information to that account.

Three modal dialogs are shown side-by-side:

- Doctor Assigned!**: Shows a green checkmark icon. Text: "Dr. Realman's information saved to Bob's profile". Buttons: "OK".
- Doctor already assigned**: Shows a blue info icon. Text: "Dr. Realman's information already saved to Bob's profile". Buttons: "OK".
- Doctor already assigned**: Shows an orange exclamation mark icon. Text: "Replace Dr. Realman with Dr. Octavia for Bob?". Buttons: "Cancel" (gray) and "OK" (red).

Safety features are also in place to let the user know the state of the Doctor info for a particular account when trying to change it.

Files:

The screenshot shows the Docdoc application's interface. At the top, there is a navigation bar with tabs: 'Docdoc', 'Feed', 'Calendar', 'Find Nearby', 'Files', and a search bar labeled 'Search...'. Below the navigation bar, the 'Recent Files' section is displayed. This section contains five document previews, each with a small icon and the file name below it. From left to right, the documents are: 'Lurie Children's Hospital consent form' (icon of a person), 'CPS enrollment form' (icon of a graduation cap), 'UI Health immunizations' (icon of a syringe), 'DCFS health examination' (icon of a medical cross), and 'Illinois health examination' (icon of a medical cross). To the left of these documents, there is a sidebar with three items: 'Bob' (with a boy icon), 'Jannette' (with a girl icon), and 'Add Account' (with a plus sign icon). The bottom left corner of the screen has a link 'About Material'.

The files tab essentially allows the user to store files related to them and their dependents. Under the files tab, the user is shown previews of the documents that they have stored in addition to their names - which improves the user's efficiency as seeing the previews allows for better recognition coupled with the file names which also aids in both recognition and recalling.



Recent Files

Sort Date: Latest ▾

<p>School Name: Chicago Public Schools School Enrollment Form</p> <p>Child's Name: Name: Bob First Name: Bob Middle Name: None Last Name: Chicago Gender: Male Date of Birth: 01/01/2005 Registration Month/Year: 01/2005</p> <p>Parent, Guardian and Relative Information: Relationship: Father First Name: John Last Name: Chicago Middle Name: None Date of Birth: 01/01/1975 Phone Number: 312-555-1234 Email Address: john.chicago@cps.k12.il.us Address: 123 Main Street, Chicago, IL 60601 City: Chicago State: IL Zip Code: 60601</p> <p>Demographic Information: Race/Ethnicity: White Gender: Male Age: 13 Height: 5'0" Weight: 140 Blood Type: A+</p> <p>Health Information: Allergies: Pollen, Bees Medication: PollenBeeGone Age: 13 Weight: 140 Blood Type: A+</p> <p>CPS enrollment form</p>	<p>State of Illinois Certificate of Child Health Examination</p> <p>Medical Name: Bob Chicago</p> <p>Medical Record Number: None</p> <p>Date of Birth: 01/01/2005</p> <p>Sex: Male</p> <p>Height: 50 inches</p> <p>Weight: 40 pounds</p> <p>Color: Blue</p> <p>Eye Color: Brown</p> <p>Hair Color: Red</p> <p>Medical History: None</p> <p>Immunization Status: None</p> <p>Physical Examination: None</p> <p>Healthcare Provider: None</p> <p>Illinois health examination</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Info

Allergies:
Pollen
Bees

Medication:
PollenBeeGone

Age: 13

Weight: 140

Blood Type: A+

Doctor Info

[Find A Doctor](#)

School Info

Name: Middlest High School
Number: 888-432-4353
Email: nurse@achoo.edu
Address: 4800 Avenue Q

Not only that, because the left panel has the user/dependent names and each item has the affordance of a clickable item, if the user clicks on them, the files will be filtered - showing those that are related to the person the user has selected.

Send File

[Download](#)

Send To

 [Send](#)

Rev. 07/2014

Chicago Public Schools School Enrollment Form

School Name _____

Student Information Student's siblings' names if currently enrolled in CPS: <hr/> <hr/> <hr/>		School Use Only: Prevent duplicate student records. Search in SIM for an existing Student ID before creating a new one.
		<input type="text"/> Last Name <input type="text"/> First Name <input type="text"/> Middle Name <input type="text"/> Generation (Jr., etc)
		<input type="text"/> Gender <input type="text"/> Birth date (mm/dd/yyyy) <input type="text"/> Registration Grade Level (when first entering CPS)
Personal, Immigrant, and Refugee Information To Parent/Guardian: <i>CPS is required to keep a count of immigrant students for Federal and State Guidelines in order to determine if additional resources and services for students are needed.</i> <i>Note that this is <u>not</u> an inquiry on citizenship status, and all information will be kept confidential.</i>		<input type="text"/> Y / N <input type="text"/> Birth Certificate on File <input type="text"/> Birth Verification Type <input type="text"/> * Birth Country <input type="text"/> Birth State <input type="text"/> Birth City <i>* Complete if student was <u>not</u> born in the United States (US) or one of its Territories:</i> <input type="text"/> Date of first enrollment in any US School: _____ <input type="text"/> Full Years completed school in US: _____ <input type="text"/> Student has refugee status: Y / N <input type="text"/> Country of refugee: _____ School Use Only: Note that "Date of first enrollment in any US School" becomes a required field in SIM if "Birth Country" is <u>not</u> the US or one of its Territories.
Student Address/Phone Physical (Home) Address		<input type="text"/> Street Number and Name <input type="text"/> Apt <input type="text"/> City <input type="text"/> State <input type="text"/> Zip Code

Upon clicking on a file, a modal will be displayed with a larger preview of the file that they have selected so that the user has a better view of what the file is. Above is a download button which the user can click in the case that they need to and a text field and send button where they can enter in an address and have the file sent to them. The buttons, when clicked, will change statuses such as Download -> Downloading... -> Downloaded and Send -> Sent!.

Such feedback allows the user to know that the action that they desired has been processed. The reason for having such feedback is that when the user clicks on anything, their attention is directed at the cursor and with that, it easily grabs the user's attention instead of having to pop up from somewhere on the screen far from the scope of the user's vision.

Implementation:

Our interface was implemented as a website in CSS, HTML and Javascript. Bootstrap was used extensively for layout and styling, as well as some of the modals. Jquery was used extensively for the javascript event handling. The Sweetalert library was used to create the user messages and dialogue for the 'Find Nearby' tab.

User accounts were saved using javascript session storage, and accounts were loaded automatically based on the saved data. In the event of no saved data, account information for Bob and Jannette were populated by default.

The calendar makes use of Javascript frameworks such as moment.js and underscore.js. A simple web based JSON store was used to display example events on the calendar and the feed. When a new event is added, the event information was saved in local storage and used to create a new event. Currently, only one custom event is supported, and a new event will overwrite the old one. Sweetalert popups were used in the calendar to notify the user when an event is added or when there is an error while filling the add event form.

The Google Maps API was used with Javascript to create custom markers and labels that were then interactable with the rest of the find nearby tab. The doctor's information was stored in global variables for the find nearby and account information sections that used it.

The Files tab, like many others, was implemented using HTML and CSS to design the modals and previews while actions and animations were utilized with the help of both Javascript, JQuery, and CSS. Insertion, deletion, and proper download/send functions were not implemented as a result of no proper backend to support such features.

By implementing the system as a web app, we were able to make the app very easily distributable and accessible by multiple devices. Bootstrap and proper sizing also made making the layout responsive accessible. By relying mainly on jquery or vanilla javascript for much of the functionality, we were able to have a good amount of design freedom, at the cost of efficiency, which meant that there were many features that weren't implemented. This was also slightly hampered by the lack of a proper backend, as much of the interface would rely on dynamically changing data. Some of the components that did use external apis - the calendar and map - allowed us to make those components more fleshed-out, but also made interfacing those components with the rest of the interface more difficult. This mainly occurred in the calendar, which had arguably the most functionality and most well-received design, but wasn't as well incorporated into the custom functionality in the rest of the app (such as filtering by user or showing events when clicking on them in the feed).

Evaluation:

3 Users were tested using the web interface using a laptop running google chrome on a 1600 x 900 resolution laptop. All users were of the target age for the intended audience (young adults), although only one was currently a parent. All users were given the same briefing (below) and asked to think out loud as they completed the tasks, but otherwise weren't given additional instructions or a demo. Tasks were printed on cards and given to the users as needed. In some cases, users made minor mistakes in the tasks, such as inputting the wrong date or email, which was noted, but users weren't asked to re-do the task.

Tasks:

1. Find the nearest doctor for Bob, and assign him to Bob's profile
2. You Scheduled an appointment with Dr Realman for Bob for May 19, 2019 @ 11:30 AM. Record this information in DocDoc as a reminder
3. Find Bob's "UI Health Examination" form and send it to Bob's school's email.

Briefing:

DocDoc is an application that organizes all of your medical needs for your children. By keeping track of their doctors and documents, DocDoc allows you to track appointments, keep documents updated, and easily send them to your child's doctor or school.

Users can view a feed of all recent uploads and upcoming events, a list of all their medical documents, and a calendar. Users can also search for nearby doctors using our map feature.

By selecting a specific dependent, a user can filter their documents and events as well as view and edit specific information for that dependent.

DocDoc is best suited for parents who need help keeping track of and organizing all medical-related aspects of their children. Users must have basic computer skills for using our site and uploading relevant documents.

Thank you for participating in our user study. The objective of this study is to help improve our design so users are able to accomplish their goals easily, not to test your ability to use our application.

Usability Problems and Possible Solutions:

1. 2 Users had trouble identifying which doctor was the 'closest' when completing task 2. Both didn't find it obvious that the doctors were sorted by distance, and didn't immediately recognize which markers on the map corresponded to which doctors or the user immediately.

Possible solution: Adding in the distance to user in the boxes with the doctor's information would allow for more information for the user. An optional sort filter could be implemented that could also serve as a way to let the user know how the search results are ordered. Finally, making the marker on the map for the user distinct, or adding a function to center of the user could help make the process of searching the map for the closest doctor more efficient.

2. Users did not seem to find the email address of the school easily.

Possible solution: The send feature could have an autofill, or a pre-populated options list of emails to send files to to help efficiency at the cost of user freedom in some cases.

3. Some users didn't recognize that the account icons were clickable, or the different functions that they had (such as filtering and providing school emails).

Possible solution: Change the icons to have stronger visual affordances as a button, such as showing a colored border when being hovered over. Adding a tutorial or more instructions in the interface to suggest functionality could also help, such as tooltips, an about page, or overlays

for first time users. Making more important information such as emails more prominent when showing account information may also help, rather than dividing the cards into user, doctor, and school info.

4. All users tried to add an account for a different month (march or april) than was intended (may).

Possible solution: Add in a field in the forms for the event's date when adding an event, rather than making the event default to the selected day, but pre-populated the date with the selected date to keep efficiency while adding user freedom and giving visual feedback on the date of the event being added. When an event is added, add the long-form date and event title into the confirmation dialogue.

5. All users exceeded the character limit when trying to input the title for the event in task 2

Possible solution: Increase the character limit. Alternatively, leave only a description, and generate a title based on a limited number of event options (such as event type, and associated doctors/dependents/schools), and leave the long-form description to just the description box, as many users expressed that they weren't sure what to put in the title vs the description of the event.

6. One user wasn't sure what file was associated with bob when completing task 3

Possible solution: Add annotations for user-color in the files tab, either via underlining the test, changing text color, adding a colored border, or adding a colored icon (such as a circle) somewhere in the overlay with the document name.

Reflection:

During the initial design process we ended up with a number of high level designs that we could have gone with, such as having a user-based interface, rather than the feed/tab based designed we settled on. However, we had trouble deciding on a final design to use for the user prototypes, and the final design choice was very subjective. Using parallel prototyping during the initial paper prototyping/design phase may have been helpful for identifying one but it may not have costed much extra time as we couldn't have 5 people working on a single paper prototype at once.

During the user evaluations, we relied on three basic tasks, but didn't require the user to use all possible functionality of the app. Doing the formative evaluations earlier on focusing on tasks starting from earlier on, such as creating an account first, could have provided more insight into the workflow of the user. Additionally, by pre-populating a lot of information, we effectively skipped over a portion of the interface that could give the user information on what options in the user accounts.

During user testing some of the most interesting findings were when there were multiple ways of accomplishing a task, such as having an 'add event' button in the calendar, vs clicking on an individual cell. For some other design choices, it might be an effective strategy to provide multiple interaction options and test users responses more extensively for specific tasks, as a way of doing early AB style testing.

Finally, our system had a number of implementation details that weren't dealt with, but would provide design challenges for a full-fledged implementation. Namely, a lot of our inputs have a problem of flexibility versus safety, such limiting some inputs such as email recipients, or restricting doctors to those found through the find-doctor interface. Doing more extensive quantitative testing on smaller, specific tasks where we measure error and time for different implementations could provide better insight into the safety-efficiency tradeoff in the app.