

All

- Loader is unnecessary
- Class, file naming conventions aren't the best
- Lack of comments throughout
- A lot of inline styles being used, recommend moving over the external css.
- Very ugly to look at, recommend formatting using prettier or manually to make it easier to understand.
- Definitely an insane amount of div tags just for a footer that says two things.
- Report an issue button is nested within a UL, doesn't feel right.

Main page

- Simple landing page with 3 selection options.
- Use better alt text
- Contains a report button at the top of the page.
- Embedded stylesheet at top can just be moved to the main css file, especially since it's not a lot - just to make code look cleaner.
- Just call it index.html
- If we remove the loader, the jquery at the bottom is useless. I still don't see its actual purpose.
- A good amount of bootstrap seems to be used for div placement, and can be replaced with grid/flex.
- Random i tag within a button, not sure what its for - line 40.

Palliative Care Home (NCH-Palliative)

- Takes you through a checklist of things to confirm before continuing on.
- Presents you with some options if not a patient.
 - Pressing not a patient takes you to a diagram
 - Pressing guidelines takes you to a guide given by the department.
- Broken span tag
- Js at bottom to hide/unhide button, would feel more intuitive if it was just disable/enable bc at first unclear if there's even a button at all on the front-end.
- Misspelling of Address - line 37

Palliative 1

- Asks the user what the patient is calling for and displays popups containing instructions on what to do
 - Billing
 - Medical Records
 - Appointment - goes to page 2
- Includes a button to go back a step or view guidelines.

- Don't like how buttons with popups have cursor: pointer applied, makes it misleading

Palliative 2

- Asks the user if the patient has HMO insurance and provides options of:
 - Yes - goes to page 6
 - No - goes to page 3
 - Go back step or view guidelines
- Button styles can all be applied in external css to make for easier viewing/modifying
-

Palliative 3

- Asks the user if the patient has a referral for a specific provider, provides options of:
 - Yes - goes to page 5
 - No - goes to page 4
 - Go back step or view guidelines
- Very redundant, same as previous page

Palliative 4

- Tells the user to schedule an appointment with the first available provider according to the guidelines. Given options of:
 - New patient
 - Follow up for existing patients
 - Virtual visit
 - Go back step or view guidelines
- Pop ups for each option, decision tree ends here

Palliative 5

- Same as page 4, only difference is "schedule appointment as per guidelines"

Palliative 6

- Asks if patient has an active referral, given options:
 - Yes - goes to page 7
 - No - gives popup, ending decision tree
 - Go back step or view guidelines

Palliative 7

- Asks the user if the PCP on the referral the same one listed as the PCP according to RTE
 - Yes - goes to page 8

- No - gives popup, ending decision tree

Palliative 8

- Tells the user that it's on the schedule with the provider according to appointment guidelines. Gives options:
 - New patient
 - Follow up for existing patients
 - Virtual visit
 - Go back step or view guidelines
- The decision tree ends here.

Overall thoughts

- The code is very ugly to look at and therefore hard to tell what's going on at first glance.
- Way too many inline styles are reused here when they can just be mass applied on an external sheet.
- A large chunk of the bootstrap being used seems redundant or unnecessary and can be replaced with simple css, along with using grid/flexbox to organize divs.
- Many pages are copy/paste versions of each other, need to find a way to just have one version of each page instead of having 9 different copies of the same thing.
- Feels like there's way too many divs on the page for what is actually happening.
- Code will feel much more polished by just cleaning up the naming, formatting, and inline styles

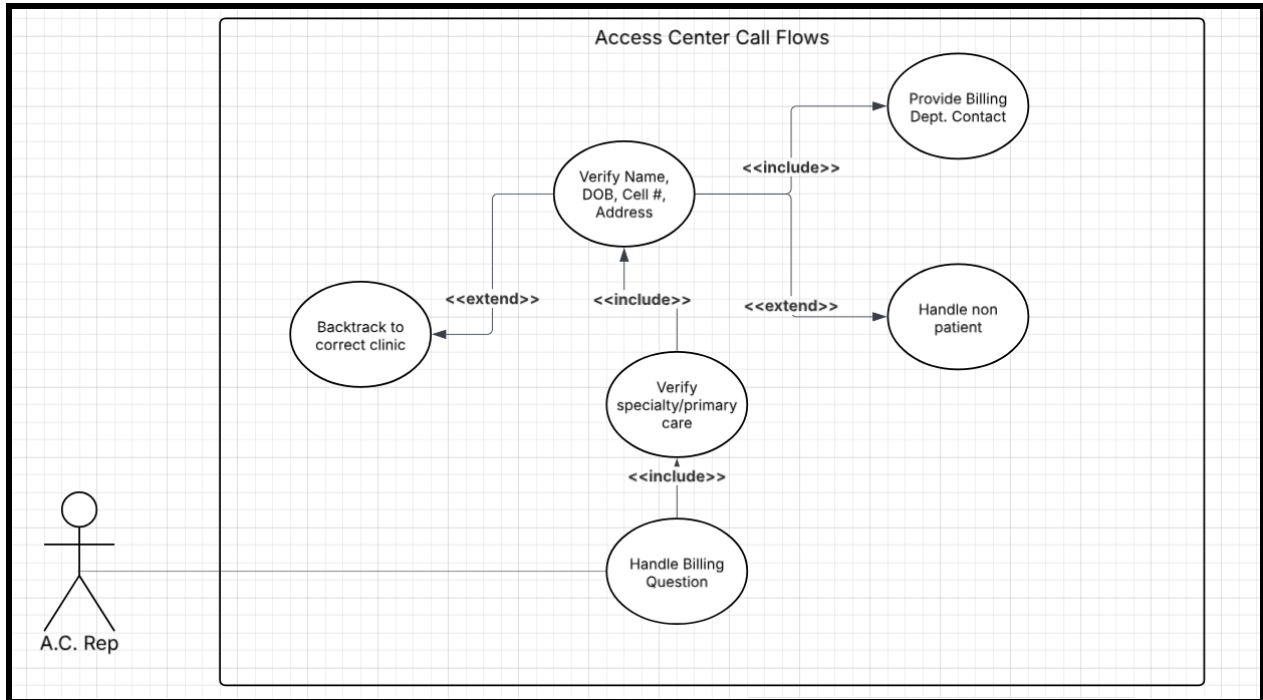
Possible tech stack

- Front-end
 - ReactJS + Vite + Tailwind/Normal css
 - Can have pages built as reusable components, eliminating the need for multiple copies of a single page
 - Tailwind just to have cleaner css and avoid inline styles, can settle for standard external css.
 - Good if we plan to scale in the future and uses modern practices (good for me)
 - Only issue is it's not maybe as lightweight as you'd like
 - Alpine.js + tailwind - **(likely best option if we want to keep it super lightweight)**
 - Alpine is similar to Vue but lighter and offers simple ways to show popups or hide/disable buttons if we wanted to keep that functionality.
 - Same reasons for tailwind
 - Svelte
 - Good for dynamic flow
 - Very lightweight, minimal, and easy to maintain.
 - Good code to feature ratio
- Back-end
 - Node.js + Express

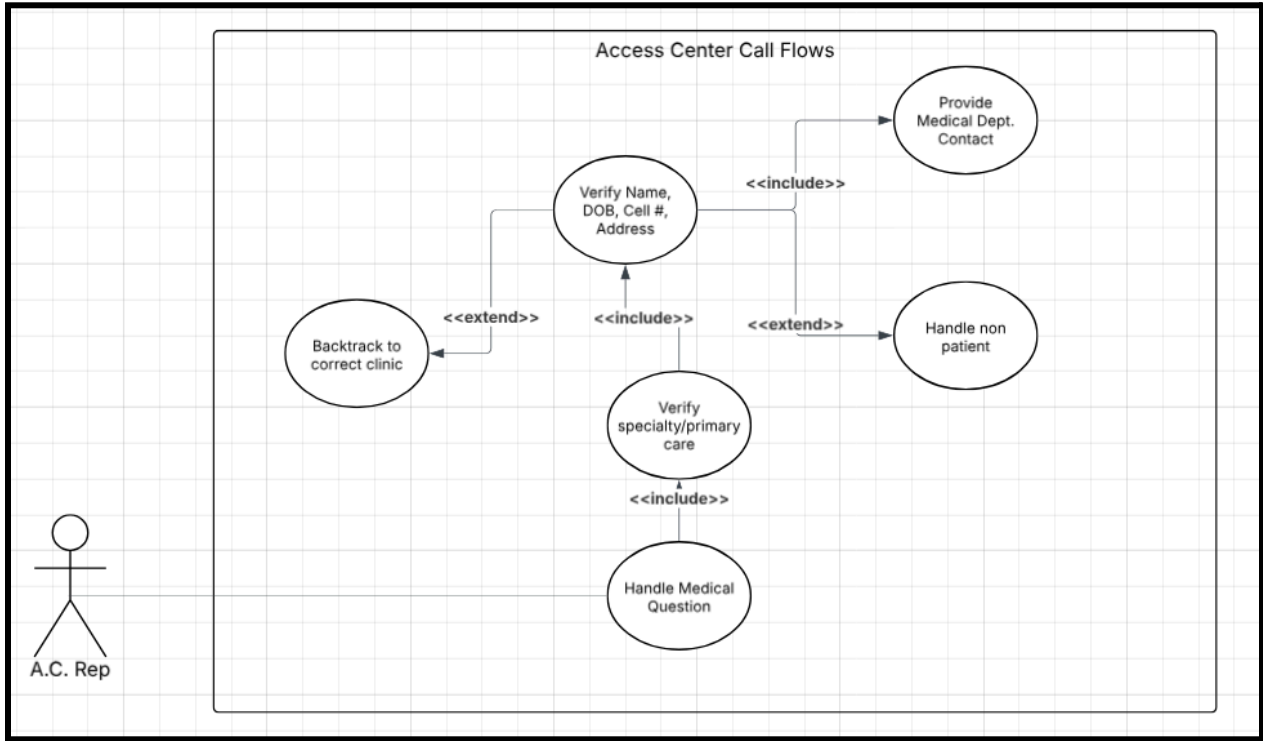
- Simple and fast
 - Good for routing and APIs
 - Easy to deploy to internal servers
 - Works easily with react/alpine
- Flask
 - Easy db integration
 - Easy to build full apps or APIs
- Database
 - PostgreSQL
 - Easy to manage content
 - Can plug directly into front end
 - Provides a clean structure for decision trees

Use cases

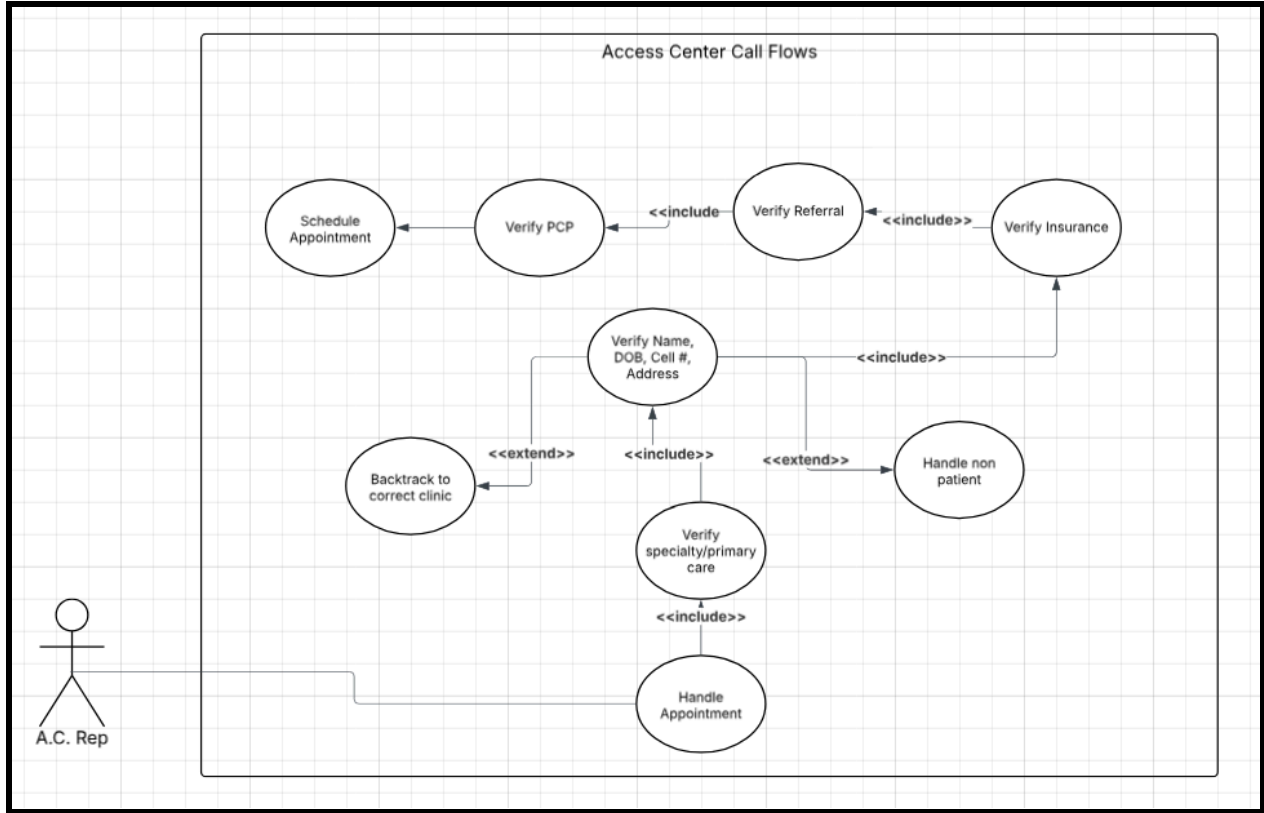
Use case:	Caller has billing questions
Primary Actor:	Access Center Representative
Goal in context:	Successfully direct the caller to the billing department.
Preconditions:	Caller must know the department they need information from.
Trigger:	Caller presents a question regarding billing
Scenario:	<ul style="list-style-type: none">• User must know if the caller is calling for a speciality or primary care<ul style="list-style-type: none">◦ Click 'NCH Specialties' or 'NCH Primary Care'• Know the particular clinic they are trying to reach.<ul style="list-style-type: none">◦ Click any 1 of the 8 clinics• Confirm the caller's name, DOB, address, and phone number.<ul style="list-style-type: none">◦ Click 'Name', 'Date of Birth', 'Address', 'Phone Number'◦ The 'Next' button will activate; click it.• Redirect the caller to the billing dept's phone number.
Exceptions:	<p>User clicks on incorrect clinic</p> <ul style="list-style-type: none">• Press the "previous step" button to go back to the previous page <p>Caller not a patient</p> <ul style="list-style-type: none">• Press the "Not a patient" button for instructions



Use case:	Caller has medical questions
Primary Actor:	Access Center Representative
Goal in context:	Successfully direct the caller to the medical department.
Preconditions:	Caller must know the department they need information from.
Trigger:	Caller presents a medical question
Scenario:	<ul style="list-style-type: none"> User must know if the caller is calling for a speciality or primary care Know the particular clinic they are trying to reach. Confirm the caller's name, DOB, address, and phone number. Redirect the caller to the medical dept's phone number.
Exceptions:	<p>User clicks on incorrect clinic</p> <ul style="list-style-type: none"> Press the "previous step" button to go back to the previous page <p>Caller not a patient</p> <ul style="list-style-type: none"> Press the "Not a patient" button for



Use case:	Caller wants to make an appointment
Primary Actor:	Access Center Representative
Goal in context:	Successfully create an appointment for the caller.
Preconditions:	Caller must know the department they need an appointment for.
Trigger:	Caller presents a need for an appointment
Scenario:	<ul style="list-style-type: none"> • User must know if the caller is calling for a speciality or primary care • Know the particular clinic they are trying to reach. • Confirm the caller's name, DOB, address, and phone number. • Redirect the caller to the billing dept's phone number. • Caller must know PCP, referral, and insurance information
Exceptions:	<p>User clicks on incorrect clinic</p> <ul style="list-style-type: none"> • Press the "previous step" button to go back to the previous page <p>User chose incorrect step in decision tree</p> <ul style="list-style-type: none"> • Press the "back" button to go to the previously asked question <p>Caller not a patient</p> <ul style="list-style-type: none"> • Press the "Not a patient" button for instructions



Performance by Page

Page	Updated			Original		
Metric	LCP	CLS	INP	LCP	CLS	INP
Landing Page	0.47s	0.04	40ms	0.08s	0	–
Specialties	0.41s	0.12	48ms	0.13s	0.87	–
Checklist	0.47s	0	32ms	0.12s	0.88	24ms
Reason for calling	0.42s	0	24ms	0.08s	0.87	—
Tree-1	0.43s	0	16ms	0.07s	0	16ms
Tree-2	–	–	–	0.13s	0.87	16ms
Tree-3	–	–	–	0.12s	0	32ms
Tree-4	–	–	–	0.11s	0.87	24ms
Tree-5	–	–	–	0.11s	0.87	16ms
Tree-6	–	–	–	0.04s	0.87	16ms
Tree-7	–	–	–	0.13s	1.22	24ms

Future Updates -

- Part of the process (Rounding Insurance of Epic Checklist box and reminder box before proceeding to schedule) after HMO. After HMO did you run the insurance probably suggest how to run the insurance. Eventually we could include links as long as we get a checkbox and make sure they
 - Before moving forward with the appointment they will have an Acknowledgement check box before they continue.