IDEA and USERS

What problem(s) does your app solve?

My app streamlines the process of charting for CNAs in long term care facilities

How does it solve those problems?

This app will focus on a directional flow of information, going from general to specific. This will eliminate the need for pop ups and excessive text.

Who is your target user?

CNAs

How much experience do they have with technology?

Experience with technology varies widely among CNAs, and so the technology must be easy to adapt to. Clear directions and a simple interface are key.

FEATURES

Write use cases for your app.

- For example, as a user I can store a list of my favorite food.

Use color to separate features by 'Minimum Viable Product' and 'Bonus features'

As a user I can:
select a resident from a list of residents
select an ADL
Choose the support given and quality of that support
Record measurements and percentages for various ADLs
Edit ADLs
Input Vital signs

Add a resident to a subgroup for easy organizing
Use Alexa to chart ADLs
View a list of previous responses
Add ADLs/customize ADLs
Print out a summary of responsibilities/tasks for a subgroup of residents

IDEA and USERS

What problem(s) does your app solve?

My app streamlines the process of charting for CNAs in long term care facilities

How does it solve those problems?

This app will focus on a directional flow of information, going from general to specific. This will eliminate the need for pop ups and excessive text.

Who is your target user?

CNAs

How much experience do they have with technology?

Experience with technology varies widely among CNAs, and so the technology must be easy to adapt to. Clear directions and a simple interface are key.

FEATURES

Write use cases for your app.

- For example, as a user I can store a list of my favorite food.

Use color to separate features by 'Minimum Viable Product' and 'Bonus features'

Select meal items from a menu
Add incidents
Set reminders
See charts and trends
Shower sheet

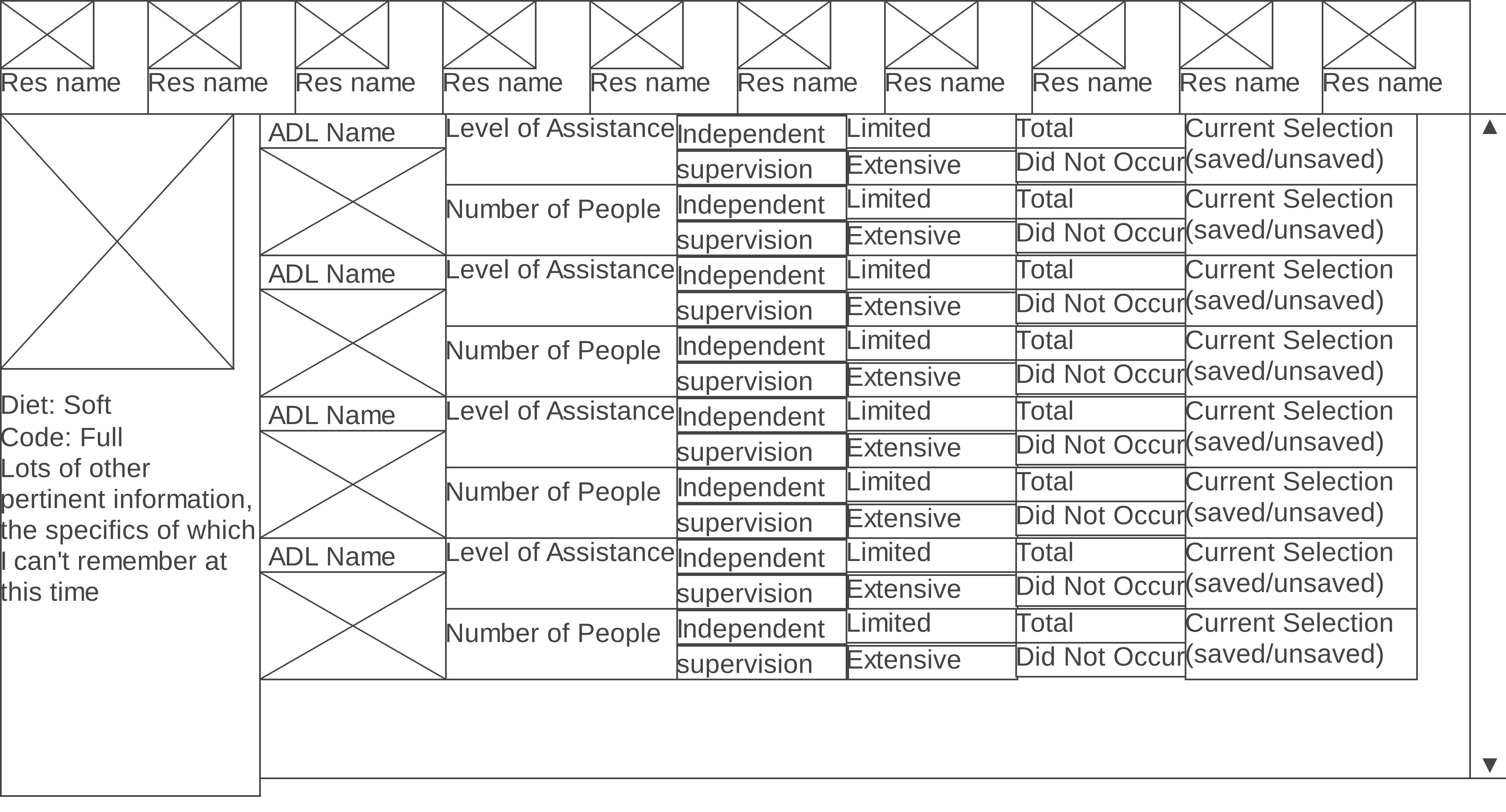
VIEWS

What views do you need to create to meet each feature in your app? Draw below.

How will the user get to each view? Draw below.

Revisit this regularly. Simplify each time. Focus on the user.
Draw each view and map out how the user will get to each of them.

The page below would be the main view. The user would get here from the landing page.


Res name
Res name
Res name
Res name
Res name
Res name
Res name
Res name
Res name
Res name
Diet: Soft
Code: Full
Lots of other pertinent information, the specifics of which I can't remember at this time
ADL Name
Level of Assistance
Independent
supervision
Limited
Extensive
Total
Did Not Occur
Current Selection (saved/unsaved)
Number of People
Independent
supervision
Limited
Extensive
Total
Did Not Occur
Current Selection (saved/unsaved)
ADL Name
Level of Assistance
Independent
supervision
Limited
Extensive
Total
Did Not Occur
Current Selection (saved/unsaved)
Number of People
Independent
supervision
Limited
Extensive
Total
Did Not Occur
Current Selection (saved/unsaved)
ADL Name
Level of Assistance
Independent
supervision
Limited
Extensive
Total
Did Not Occur
Current Selection (saved/unsaved)
Number of People
Independent
supervision
Limited
Extensive
Total
Did Not Occur
Current Selection (saved/unsaved)
ADL Name
Level of Assistance
Independent
supervision
Limited
Extensive
Total
Did Not Occur
Current Selection (saved/unsaved)
Number of People
Independent
supervision
Limited
Extensive
Total
Did Not Occur
Current Selection (saved/unsaved)

COMPONENT TREE

Create a component tree for your app with all the components and their initial state. Make a connection to each endpoint it needs to use to get its data.

App

State:
none

Methods:
none

Routes

Login

State {
username:""
password:""
}

Methods
Login
Handle Changex2

Store

state {
username:""
name:""
userid:0
}

action builders:
Login
Logout

Dashboard

State {
shift: ""
facilityid:""
Section:
Subsection:
}

Methods:
selectShift
Handle
Change
x3

Resident List

State {
residents:[
{resName:""
resPic:""}
]
}

Methods:
pickRes
handleChange

Resident

state {
resName:
resImg:
OtherHealthInfo:""
ADLList:[]
}

Methods:
getList

Vitals:

ADL

state {
ADLList:[
{ADL:"",
Level:"",
Number:""
Other ...

Methods:
displayChoice
highlightChoice
handleClickx2

ENDPOINTS

List the url, REST method, and a sample of the data being sent or received for every endpoint you need.

POST
‘/api/login’
Receive:
req.body {
username:””
password:””
facilityid:””}
Send:
{username:””
firstName:””
lastName:””
userid:0,
residentList:[
residentids]
}

(cont.) {
groups:{
groupName:””,
groupRooms[]}

POST
‘/api/residents’
Receive:
req.body {
[residentList]
}
send:
[
{residentName:””,
residentid:0,
residentADLs:[ADLids]
]

PATCH
‘/api/edit/:residentid’
Receive:
req.body {
ADLid:0}
req.query {
primaryChoice:4
secondaryChoice:4
tertiaryChoice:4 }

GET
‘/api/history/:residentid’
send:
[{ADLid:4,
caregiverFirstName:””,
caregiverLastName:””,
timestamp:time,
primaryChoice:4,
secondaryChoice:4,
tertiaryChoice:4

DELETE
‘/api/delete/:adlid’
Receive:
req.params {adlid:4}
send:
GET history

SCHEMAS

Create your schema(s) here. Make sure to include data types and any foreign key constraints.

residents				
column name	**data type**	**constraints**	**notes**	
id	serial	primary key		
facilityID	integer	references facility (id)		
firstName	varchar(30)			
lastName	varchar(30)			
diet	varchar(30)			
code	varchar(30)			
caregivers				
id	serial	primary key		
firstName	varchar(30)			
lastName	varchar(30)			
adlSchema				
id	serial	primary key		
name	varchar(30)			
primaryChoices	array		usually self-performance	
secondaryChoices	array		usually assistance provided	
tertiaryChoices	array			
specialCode	integer			
other (choices)	array			
ADL				
id	serial	primary key		
residentID	integer	references residents (id)		
caregiverID	integer	references caregiver (id)		
ADLid	integer	references adlSchema (id)		
primaryChoice	integer		usually self-performance	

secondaryChoice	integer		usually assistance provided	
tertiaryChoice	integer			
timestamp	datetime			
facility				
id	serial	primary key		

FEATURE PROGRESS TRACKER

List the features below	Component file created	Create dummy data for component	Test component is functioning with dummy data	Build endpoint for data	Test endpoint with Postman	Connect endpoint to frontend	Test component is functioning with actual data	Component styled
Text								

	Component file created	create dummy data for component	test component is functioning with dummy data	build endpoint for data	test endpoint with postman	connect endpoint to frontend	test component is functioning with actual data	component styled
Auth0								
Create Resident								
Add ADLs								
Edit ADLs								
Navbar								
Resident Picker								
ADL Picker								
Resident Info								
Delete entry								
Meal %								
History								
Groups								
Print tasks								
Fill out menu								
Change in condition								
Charts and trends								
Shower sheet								