

Ride for life project design notes

Contents

1	Project specification.....	2
2	Nested and non-nested routes used in the design.....	2
2.1	Mom pages.....	2
2.1.1	MomDriversList component hiearachy.....	3
3	Nested redux reducers with persist for one sub-reducer	4
4	Custom hook to measure the width and height of an element.....	4
5	Two configurations for axios for API calls	5
6	To-do list for future work.....	5

1 Project specification

PITCH

Pitch: Have a chance to work with a real client! SAFE serves mothers in Uganda to improve maternal and child survival. One of the biggest obstacles women face while pregnant is getting to a health facility to safely deliver. Most people do not own a car or a motorcycle, and a majority live greater than 5 km from a health center (an hour by foot). Can you imagine trying to walk an hour on foot when you were pregnant, and in labor? SAFE provides motorcycle ambulances to improve this problem, but mothers have a hard time contacting them in time. This app identifies and sends a text to the nearest motorcycle ambulance driver.

MVP

This app contains two user types. A ride requester (pregnant mom, or their caregiver) and a trained motorcycle ambulance driver.

1. An on-boarding process for a new general user (user that will consume the service)
2. On-boarding process for driver (create their profile with name, location, and price)
3. Ability for driver to edit and delete their profile
4. Ability for user easily create, edit and delete driver reviews.
5. Ability for user to search for drivers based on their location and view their profile page.

2 Nested and non-nested routes used in the design

Nested and non-nested routing is used for different URL paths in this project. This is done with some routes private and some public.

There are two kinds of private routes. The first one is mom-private that is only accessible by a logged in mom. The second one is driver-private that is only accessible by a logged in driver. If a user tries to navigate to a path that they are not allowed to, they will be redirected to their profile page.

2.1 Mom pages

The route prefix `/mom` renders the component `MomPage`.

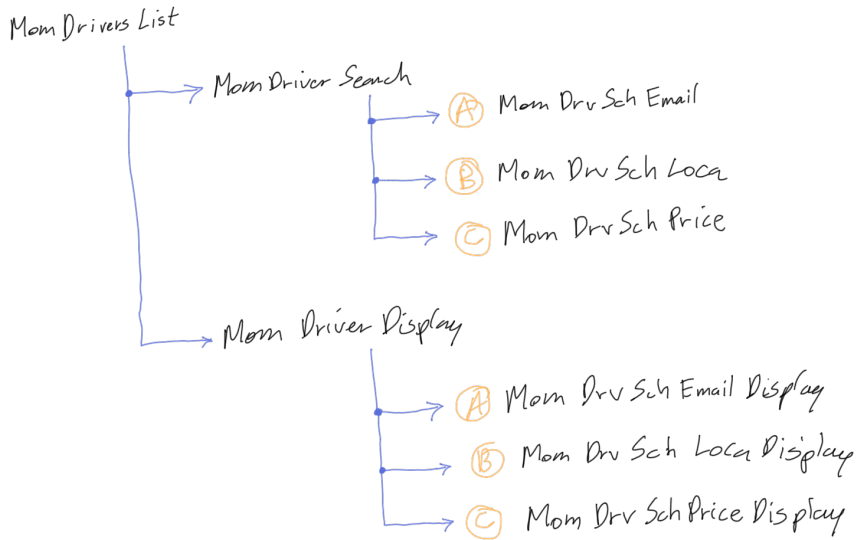
The `MomPage` component then renders the following components based on nested routing:

Nested Route	Component rendered	Type of route	Function of the route/page
<code>/mom/profile</code>	<code>MomProfile</code>	mom-private	Show logged in mom profile info with link to update it
<code>/mom/driversList</code>	<code>MomDriversList</code>	mom-private	Ability for mom to search for drivers by location or price
<code>/mom/reviewsList</code>	<code>MomReviewsList</code>	mom-private	Ability for moms to add, update, delete driver reviews

The mom page to update a mom profile is rendered by the component `MomRegisterUpdateProfile` (with its `useForm` attribute set to update) at the non-nested and mom-private route `/mom/updateprofile`.

In addition the mom page to register a mom is rendered by the component `MomRegisterUpdateProfile` (with its `useForm` attribute set to `register`) at the non-nested and public route `/mom/register`.

2.1.1 MomDriversList component hierarchy



The 'A' components are rendered for a mom to do a search by driver email.
The 'B' components are rendered for a mom to do a search by driver location.
The 'C' components are rendered for a mom to do a search by driver price

3 Nested redux reducers with persist for one sub-reducer

The diagram below shows how the reducers are nested.

```
root Reducer = {  
  user Data : user Data Reducer  
  mom Data : mom Data Reducer  
  driver Data : driver Data Reducer  
  ui Data : ui Data Reducer = {  
    ui Mom Drv List : ui Mom Drv List Reducer  
    ui Mom Rvw List : ui Mom Rvw List Reducer  
  }  
}
```

The userDataReducer controls the user data saved in the redux store. This is the only redux store data that is persisted.

The momDataReducer controls the data saved in the store to help display the mom pages.

The driverDataReducer controls the data saved in the store to help display the driver pages.

The uiDataReducer controls the data saved in the store to help control the UI.

The uiMomDrvListReducer controls the data saved in the store to help control the UI for the display of mom searching for drivers.

The uiMomRvwListReducer controls the data saved in the store to help control the UI for the display of mom reviewing drivers.

Also localStorage is used to store the JWT and userId of the user.

4 Custom hook to measure the width and height of an element

The useSizeObserver is a custom hook that uses the resize-observer-polyfill npm package to measure the width and height of an element on a page. This can then be used to customize CSS code (applied through the styled-components npm package for this project)

This hook is not generally written though and needs further work. It can only be used in a React functional component that contains a React DOM element to be measured that is always present as long as the React functional component is rendered. In other words, you cannot use it in a React function component to measure a React DOM element that it contains and where the React DOM element is conditionally rendered.

5 Two configurations for axios for API calls

The axios npm package is used to do HTTP API calls. It has two configurations:

- one for calls that send the JWT. This is called axiosWithAuth
- one for calls that do not send the JWT. This is called axiosNoAuth

6 To-do list for future work

- 1) For the warning before deletes, make a component for it
- 2) Add missing please wait messages. Highlight these by using the added feature on the backend of artificial delay.
- 3) Implement auto-logout after 10min interval
- 4) Add tests
- 5) Add average rating to driver review page
- 6) Compress images so that they load faster
- 7) In update profile pages, show the current values