

1. PDV 2: Definition Open Draft

This is an enhanced version of [digital-guard](#)

- 1. Requirements
 - 1.1. Functionalities Inherited from the Previous Version
 - 1.2. Design Features
 - 1.3. Technical Requirements
- 2. Proposed Libraries
- 3. Roadmap
 - 3.1. Stages
 - 3.2. Estimated Labor
- 4. References
 - 4.1. React Map Libraries
 - 4.2. React UI Libraries
- 5. Notes

1. Requirements

1.1. Functionalities Inherited from the Previous Version

- ☐ Load GeoJSON files from GitHub [digital-guard / preservCutGeo-BR2021](#)
- ☐ Grayscale base map (default)
- ☐ Base map selector for: Google, Bing, OpenStreetMap
- ☐ Mosaic (choropleth map) with color by density
- ☐ Zoom level dependent labels
- ☐ Tooltips
- ☐ **on.click** in mosaics navigates to geohash's addresses
- ☐ Dot/Pin marker depending on the current zoom level
- ☐ Fixable popups with address details
- ☐ External toggles (fix-popups, remember-selection...)
- ☐ Geohash table with external selectors and column sorting
- ☐ Retrieve URL parameters in order to open the requested set of points (permalinks)
- ☐ Constrain the view to the bounding box of the mapped data
- ☐ Limited zoom-out

1.2. Design Features

- ☐ Single-page app
- ☐ *Full screen map vs Horizontal split* still requires some discussion 🙅🌀

1.3. Technical Requirements

- ☐ React Web app that is suitable for desktop and mobile use

2. Proposed Libraries

Having done some research on the more established React projects for mapping and UI listed [below](#), here is a *tentative proposal* of libraries to use.

Mapping: [react-map-gl](#) + [MapLibre](#). See some examples [here](#)

UI: [React Bootstrap](#)

3. Roadmap

This project is intended to be developed in the following 4 stages. Of course, they are open to discussion and modification if necessary.

3.1. Stages

1. **Design:** At this stage we should produce a set of agreed upon design concepts. It would be much better if it includes final sketches, mock-ups and proof-of-concepts. The effort should focus on creating a detailed feature list and "ideally" a very basic prototype of the future application.
2. **Development:** Coding, coding and more coding to obtain a fully functional application that implements all the features defined in the previous stage.
3. **Testing and Deployment:** This stage consists mainly of reviewing and testing all functionality from the client side (Peter) to find and fix bugs, errors and weaknesses that could affect the performance of the application in a production environment. When everyone is satisfied, the application will be ready for deployment.
4. **Maintenance:** Well, let's talk about future.

3.2. Estimated Labor

stage	deliverables	hours
1	App prototype, SRS + SOW	20
2	Development App v0.x	40
3	Production App v1.x	20
4	Fixes and Updates App vx.x TBD	
-	Estimated time (hours):	80

4. References

4.1. React Map Libraries

1. [deck.gl](#): WebGL-powered framework for visual data analysis of large datasets. 👍 It is very memory-intensive. 🙄
2. [react-map-gl](#): It is a suite of React components for Mapbox GL JS-compatible libraries. 👍 It also binds Mapbox GL forks like MapLibre. 🙄
3. [React Leaflet](#): React components for Leaflet maps. It's fine for very basic apps, but the interaction with map objects is too limited, just like Leaflet JS. 🙄

4. [React Simple Maps](#): SVG maps developed in React with d3-geo and topojson. It focuses more on representation than analysis.

4.2. React UI Libraries

1. [MaterialUI](#): It is one of the best ReactJS UI frameworks. It is also the most popular React Component UI library. Data Grid components would need upgrade 💰 to PRO version. 👎
2. [React Bootstrap](#): **Bootstrap** is the most popular front-end framework, rebuilt for React. 👍
3. [Ant Design](#): It provides a design system for enterprise products. React is used to encapsulate a library of components which embody its own design language. It adds an abstraction layer written in TypeScript to translate their design patterns into React components.

5. Notes

[react-map-gl with MapLibre](#)

```
npm install --save react-map-gl maplibre-gl
```