This prototype is aim to explore the feasibility of some requirement for CPIS Admin console and CPIS of XTech project.

MapBox background:

1. A Mapbox contains 3 type of data
   1. **Mapbox data** is the baselayer that provided, can be road network, satellite or terrain. Not editable.
   2. **Your data** iselements and styles that edit and save on MapBox server under projects of user’s account.
   3. **Local data** is data that only to display on screen without a need to upload to MapBox server.

Refer to <https://www.mapbox.com/guides/how-mapbox-works/>

1. API and examples

<https://www.mapbox.com/mapbox.js/api/v2.2.2/l-mapbox-map/>

<http://leafletjs.com/reference.html#polygon>

<https://www.mapbox.com/mapbox.js/example/v1.0.0/>

The expected outcome of this prototype should be able to verify whether using C#, HTML5, JavaScript and integrate with MapBox can satisfy the following requirement.

1. Load and Display existing Map

Acceptance Criteria: click a button to select a file that contains customized map info, the map shall be displayed.

Result:

After creating map on MapBox Editor, using HTML is able to display created map with added elements on browser.

1. Edit and save Map

Acceptance Criteria:

* 1. Add, modify, delete route
  2. Add, modify, delete legend
  3. Add, modify, delete overlay
  4. Pin, unpin markers
  5. Click a button to save the file with map info to local driver

Result:

* Using free desktop application Mapbox Studio Classic enables user to edit the style of Map.
* Using web application MapBox Editor, enables user to edit and save Map.
  + Change style
  + Add/edit/remove Marker
  + Add/edit/remove Lines
  + Add/edit/remove Polygon

However, it doesn’t support editing layers.

* After creating map on MapBox Editor, using HTML/JavaScript is able to
  + display created map with added elements on browser
  + manipulate layers, features individual or by groups
  + Add/remove ImageOverlay
  + Add/remove markers
  + Basically save only Vector Layer as GeoJson

1. Overlay select and display

Acceptance Criteria:

* 1. Show different overlay and allow selection
  2. On selection, only selected overlay should be displayed.

Result: using HTML/JavaScript is able to achieve.

1. Retrieve data from server to achieve real-time monitoring

Acceptance Criteria: The server should be able to keep pushing data to browser with a certain interval. After receiving the data, the browser should display it immediately.

Result:

Using .NET Fleck library is able to build web-socket between browser and web server to achieve real-time communication.

1. Mobile App User Real-time Location tracking

Acceptance Criteria: According to the user geo location info sent from server, the map on browser should reflect its location change.

Result:

Using leaflet JavaScript lib’s L.RotatedMarker can reflect animated user location changes on map.

1. Click the location of Mobile App User on map, his information shall be displayed

Acceptance Criteria: Specified user’s id should be printed on console.

Result:

1. Highlight Emergence request, blink

Acceptance Criteria: Click a button, specified user’s location point should begin blinking. Click another button to remove the blink.

Result:

Problems:

1. Mapbox for starter (free) account, only 1 project is allowed. The limitation of map data stored in Mapbox server is 20 polylines or polygons and 2000 markers.
2. Mapbox provided Editor only enables user to load and save map data stored in its server, data that stores in other means has to be loaded and saved manually.
3. With Mapbox.js and leaflet.js lib, it may be possible to build an editor that can fully control the map (adding marker, lines, polygon with title, description, image and styles etc.). And it also supports loading and saving data to our own database. However, trade-off is the high cost of building such a map editor.

Solutions:

1. Using Mapbox Editor: all the static map data will be stored in Mapbox server. To create overlays which allows CPIS user to select and display, the title of feature have to be named following some rule to distinguish them. For example, “race\_route.water\_point” or “evac\_route.assembly\_point”.

Advantages

* Provides almost all functionality required
* Well-design and tested, user-friendly and free to use

Disadvantages

* Feature limits
* Naming rule of feature’s title

1. Build a map editor

Advantages

* Get fully control of map editing

Disadvantages

* Will spend a great effort on requirement, design, implement and testing.
* Map editor is not the objective of this project

Decision: On 16th Oct. meeting, customer agree in principle that we should go with the 1st solution. The process will be

