This prototype is aim to explore the feasibility of some requirement for CPIS Backend, CPIS Frontend 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.
   2. **Your data** iselements and styles that edit and save on local data file or under user’s MapBox account projects.
   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.
* After creating map on MapBox Editor, using HTML 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.

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 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 achieve Animation.

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.

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.