

React 與 Google Maps 即時空汙地圖 UI 設計方案

為一般大眾使用者設計一套前端空氣污染地圖介面時，需要兼顧資料豐富度與易用性。以下將從介面模組設計、圖層呈現方式、模型資料處理、Google Maps API 整合，以及前端效能優化等面向提供建議。

介面模組設計與使用者流程

介面可切分為數個主要模組，並設計順暢的使用者操作流程：

- **地圖主視圖 (Map Display)**：中心區域顯示 Google 地圖，疊加各種污染物資料圖層。使用者進入時即看到地圖上即時空汙概況。地圖可縮放、平移並支援點擊互動，讓使用者導航到關心的地區 ¹。
- **圖層控制面板**：提供切換不同資訊圖層的介面元件（如側邊欄或浮動按鈕組）。使用者可選擇顯示特定污染物（PM2.5、NO₂ 等）的**觀測值**或**預測模型**圖層，也可開關「風險/不確定性」疊加層等。每個圖層的圖例（Legend）也應一併呈現，使用顏色梯度說明污染物濃度或風險等級 ²。
- **時間軸控制**：由於資料為即時但可能有延遲，應提供時間選擇器（例如時間軸滑桿或下拉選單），讓使用者切換不同時間的狀況（例如觀看預報未來幾小時的變化）。時間軸上標示資料更新時間，以提醒使用者資料的新鮮度（例如「資料更新於2小時前」）。
- **詳細資料檢視**：當使用者點擊地圖上的某個點（例如監測站或網格區域）時，彈出資訊視窗或側欄，呈現該地點詳細的空汙數據。例如顯示該位置各污染物當前數值、預測曲線（小型趨勢圖）、不確定範圍或健康建議等 ¹。這部分讓使用者能深入了解單點的情況。
- **搜尋與導航**：提供地區搜尋功能或定位按鈕，方便使用者快速跳轉到特定城市或自己的所在位置，查看當地空氣品質。

使用者流程範例：

1. **進入首頁**：地圖載入使用者所在位置的空氣品質概況（例如預設顯示 PM2.5 即時濃度熱區圖）。畫面上清楚標示顏色代表意義（如綠色良好、紅色危險）。
2. **切換污染物**：使用者透過圖層控制面板選擇「NO₂」。地圖隨即更新為 NO₂ 的觀測值，改以另一套色階呈現該污染物濃度。若使用者再切換為「預測模式」，則顯示模型預測的 NO₂ 分布。
3. **查看詳情**：使用者點擊一個關心的地點（可為監測站圖示或地圖任意處）。介面彈出該地點的詳細資訊，包括當前 PM2.5/NO₂ 等數值、預測未來幾小時的變化趨勢圖，以及數據的不確定性範圍。若該地區預測有高污染風險，資訊窗中可特別標註警示（例如「未來8小時內有60%機率 PM2.5 超標」）。
4. **檢視風險區域**：使用者開啟「風險區域」圖層，地圖上相對高風險的區域會以半透明紅色網底標示。使用者直觀地看到哪些區域可能超標或對健康有較高風險。
5. **調整時間**：使用者拖動時間軸查看未來預測，地圖動畫或逐步更新，呈現污染物擴散或濃度變化趨勢。過程中各圖層切換平順，確保互動流暢。

上述模組分區和流程確保一般使用者能輕鬆切換污染物與圖層、查詢特定地點和時間的空汙資訊，並取得直觀的風險提示。

圖層呈現方式與資料視覺化

為有效傳達各種空污資訊，地圖上可採用多種圖層呈現方式：

- **觀測值圖層**：將空氣品質監測站或感測器位置以**點標記**顯示。透過顏色區分污染程度（例如 AQI 標準的綠黃紅紫色標記），讓使用者一望即能分辨各地空污水平²¹。點標記上可標註污染物濃度或AQI數值，並在使用者懸停/點擊時顯示詳細讀數。若監測站數量很多，可使用**聚合技術**（Marker Clustering）在低縮放層級合併顯示，避免地圖過於擁擠。
- **模型預測圖層**：對於區域性的污染物濃度分布預測，可採用**熱區圖（Heatmap）**或**等值面填色**方式。熱區圖以連續色彩漸層覆蓋地圖，代表各區域的預測濃度強度³。例如高濃度區域以紅色顯示，低濃度為藍/綠色。Google Maps API 提供 HeatmapLayer 可將加權地理座標資料轉為熱區覆蓋⁴；但對於大量網格數據，亦可考慮使用像 deck.gl 之類的 WebGL 圖層實現以提升渲染效能³。另一種方式是繪製**等值線/等值面**（contour）圖層：根據模型網格數據算出特定濃度的等值線，將區間以不同色塊填充。這能清晰表現污染物擴散邊界，但實現上較複雜，適合重點區域放大檢視時提供。
- **化學反應/衍生資訊圖層**：針對模型提供的化學反應模擬結果（例如 NO₂+VOCs 在強日照下生成臭氧的預測），可以設計專門的圖層或提示。比如**臭氧生成潛勢**圖層：當模型偵測某區域NO₂和揮發性有機物濃度高且日照強時，可在地圖上以特殊圖示或顏色標記該區域，提示可能出現臭氧高值。這類衍生資訊可以用半透明的覆蓋顏色或圖案表示，例如紫色陰影代表「臭氧生成熱區」。使用者勾選該圖層即可看到模型推算的化學反應影響區域。必要時，也可在詳細資訊窗內呈現此類推估，例如「此區域因NO₂+VOCs作用，預計午後臭氧濃度上升」。整合不同模型的化學反應輸出，有助於將複雜的二次污染形成機制以視覺方式傳達給用戶。
- **風險區域圖層**：透過模型預報的不確定性資訊，將**超標機率**或健康風險以圖層方式顯示。例如，以**顏色不透明度**或**色調深淺**代表某污染物超過安全標準的概率：淺橘表示中等風險，深紅表示高機率超標。研究指出，只呈現單一預測值可能掩蓋不確定性並低估風險，而加入不確定性資訊能讓使用者做出更謹慎的決策⁵⁶。因此，可考慮將每個區域超標的機率以顏色填充地圖（類似「機率熱區圖」），讓用戶一目瞭然風險分布⁷。另一種方式是在高不確定性的區域以**紋理/顆粒**效果呈現：例如在預測值圖層上加入隨機抖動的點陣或閃爍效果，表示該區預測結果不穩定。這類「頻率框架點陣圖」或動態像素圖的方式，已被用於空污地圖中表達預測區間範圍⁵⁸。針對一般大眾，用色彩和透明度表達風險機率會較直觀。同時提供圖例說明，如「紅色表示該區有80%以上機率 PM2.5 超過35μg/m³」。
- **圖層疊加與比較**：介面允許同時打開多個圖層時，需設計好疊加順序和透明度。例如使用者可以同時開啟「PM2.5 預測濃度」和「高風險區域」兩層，此時可將風險圖層以半透明顯示在上方。不建議一次顯示過多圖層以免混亂，因此控制面板可限制同時激活的圖層數量，或提供**圖層組合快捷選項**（例如「顯示污染熱區+風險疊加」一鍵切換）。

透過以上呈現方式，使用者可以自由切換視圖來探索不同污染物及其預測結果，直觀理解空氣品質現況與未來走向。色彩編碼與圖層圖例確保了即使非專業人士也能解讀地圖資訊²。互動式的地圖標記與圖層開關也提高了應用的可用性和趣味。

模型資料處理與整合策略

為了呈現即時且多模型的預測資料，需要在前端對資料更新與整合進行周全設計：

- **即時資料與延遲處理**：空污觀測通常以小時為頻率更新，而模型預測可能每幾小時產出新結果。前端應設計資料快取與更新機制：定期輪詢後端 API 或透過 WebSocket 訂閱最新資料，將新觀測值及預測結果更新到對應圖層。由於可能有數小時延遲，建議在介面上顯示「資料時間戳記」讓用戶了解數據時效。例如在地圖

下方或圖例處標明「資料時間：10:00（延遲2小時）」。當新資料可用時，可以柔和地過渡更新圖層（避免突然跳變）。用戶切換時間軸時，前端查詢緩存或發請求獲取該時刻預測，確保不同時間的地圖順利載入。

- **不同預測模型的整合**：若有多套預測模型，可採**並行比較**或**集成融合**兩類策略：
 - 並行比較：在UI上提供選項讓使用者選擇模型數據來源（例如「模型A預測」「模型B預測」切換），或在地圖上以並排/分割視窗對比不同模型。同一污染物的不同模型預測可以採用相同色階，讓使用者快速比較差異。如果模型眾多，不建議全部顯示在主UI，可挑選重要的兩三種提供切換對比功能。
 - 集成融合：將多模型結果透過後端處理融合為單一預測圖層（例如取平均或以統計方法合成）。世界空氣品質指數（WAQI）即採用**綜合模型**，結合多個預報系統輸出來生成預測結果⁹。這種「集成元模型（meta-model）」能平滑單一模型的誤差，提供更可靠的預報。前端只需呈現這融合後的結果，並可在資訊處註明「結合X模型結果」。若模型間結果差異很大，也可透過額外的「不確定性/分歧圖層」呈現（例如高分歧區域標示為不確定性高）。
- **化學模組結果整合**：對於模型輸出的化學反應相關資訊（如臭氧生成），需決定是否以單獨圖層呈現或融入污染物預測中。建議重要的衍生污染物（如臭氧 O₃）直接作為獨立污染物圖層供切換，以強調其意義；同時在界面上將其與前驅污染物的關聯以說明或可選圖層顯示。例如當開啟臭氧圖層時，讓使用者同時可選擇顯示「前驅物質貢獻」疊加圖層，直觀了解哪些區域的高臭氧是因NO₂/VOC造成。資料上需要後端提供這類推估資訊，前端則做好對應圖層的繪製和交互。
- **時間與空間查詢**：支援使用者查詢不同時間或特定區域的空污狀況，實質上涉及資料的篩選與重繪。時間維度上，如前述提供時間軸控制；空間上，可提供地區清單或搜尋框。一旦使用者選定時間/地點，前端就從已載入的資料集中篩選出對應的區域數據（或請求後端獲取該區域細節），更新圖層和詳情視窗。例如用戶搜尋城市名稱時，前端平移地圖至該城市並載入/顯示該區域的監測站點與預測圖層。這要求資料格式組織上支持按區域/時間查詢，前端應對常查詢區域做好預載或緩存，以提升響應速度。

通過以上資料處理策略，即使資料有延遲或來源複雜，使用者仍能獲得一致且及時的體驗。他們可以在地圖上選擇不同預測模型結果或查看融合後的預報，同時明瞭資料時效和可信度。

與 Google Maps API 的整合建議

將 React 應用與 Google Maps 平台結合時，需要在技術上妥善整合地圖與資料：

- **React 與地圖的結合**：建議使用現有的 React 專用封裝庫來嵌入 Google Map。例如 Google 與 vis.gl 團隊提供的 `@vis.gl/react-google-maps` 套件，可讓我們以聲明式方式在 React 中使用 `<Map>` 元件¹⁰
¹¹。這種方式可將地圖視為 React 元件的一部分，方便與其他介面元素協調更新。使用該庫還能輕鬆接入 `deck.gl` 等高階可視化層¹¹，在 Google 地圖上繪製 2D/3D 資料。若不使用第三方庫，也可以在 React 的 `useEffect` 中手動初始化地圖 (`new google.maps.Map(...)`) 並管理其生命周期，但需注意避免與 React 的 DOM 更新衝突。
- **地圖圖層 API 利用**：Google Maps JavaScript API 提供多種方式添加自定資料圖層：
 - **Markers**：使用 `google.maps.Marker` 或新版的 `AdvancedMarkerView` 來顯示監測站點圖示，可自行定義圖標或顏色。點擊 Marker 可觸發 `InfoWindow` 顯示詳情。
 - **Data Layer**：使用 `google.maps.Data` 可載入 GeoJSON 資料，適合顯示多邊形區域或大量點位，並能套用樣式函數依屬性著色。可將預計超標的區域（多邊形）或網格點資料以 Data Layer 承載，根據數值設定不同顏色或透明度。
 - **Heatmap**：Google 提供 `HeatmapLayer`（在 `visualization` 函式庫中）來將一組帶權重的座標渲染為熱度圖⁴。可將預測網格點作為輸入產生平滑的濃度分布圖。但要注意預設 heatmap 會將點模糊為漣漪狀分布，可能需要調整參數（如半徑、gradient）以符合空污資料特性。
 - **Canvas/WebGL 疊加**：對於更客製化的圖層（如不確定性點陣或動態效果），可以使用 `google.maps.CanvasLayer` 或新版的 `WebGLOverlayView` 在地圖上繪製自定圖形。也可以利用

deck.gl 提供的圖層，如 `HeatmapLayer`、`ScatterplotLayer`等，直接疊加在 Google Map 之上³。這些方法適合渲染大量資料點且提供流暢的互動。

- **Tile Overlay**：若後端已生成空污分布的切片地圖影像（例如PNG瓦片），可透過 `google.maps.ImageMapType` 疊加自定圖磚。這對於預測模型輸出可視化特別有用：例如預先將未來幾個小時的PM2.5濃度生成網格影像圖層，前端只需根據時間切換圖磚層的 URL 即可。Tile Overlay 能將繁重的渲染工作轉移到伺服器，前端效能開銷小，但缺點是互動性較低（無法直接讀取像素值，需另行處理點擊查詢）。
- **地圖控制與樣式**：整合 Google Maps 時，可針對一般大眾需求調整地圖介面。例如隱藏不必要的默認控制項，只保留縮放按鈕、全螢幕切換等基本功能，避免干擾。我們也可使用自定樣式（JSON 地圖樣式）減弱底圖的顏色/細節，突出污染圖層。比方說將地圖底色調淡、道路邊界變簡潔，讓濃度色塊和風險陰影更明顯。使用 Google Maps API 的 `StyledMapType` 可達成此目的。
- **Google 環境資料 API**：值得注意的是，Google Maps Platform 新提供了空氣品質等環境資料 API¹²。雖然本專案主要使用自有的模型數據，但也可考慮結合 Google 的 Air Quality API 取得全球各地的即時空污指數或預報，作為輔助資料來源。該 API 支援查詢多種空污指數、污染物濃度及健康建議，並提供4天內逐時預報和30天歷史數據¹²。如果應用需要覆蓋沒有自有數據的區域，或做驗證比對，可透過這接口取得資訊。

總之，在 React 中運用 Google Maps API，需要妥善封裝地圖元件並選擇適合的資料圖層方式。透過官方提供的 React 整合庫或 Deck.gl 等工具，可以順利實現各種豐富的地圖視覺效果，將空污數據直觀地呈現在 Google Maps 之上。

前端效能與地圖互動優化策略

由於地圖應用涉及大量資料點和繪圖疊加，前端必須採取優化措施，確保即時互動的流暢與穩定：

- **資料載入與更新優化**：實施惰性載入（Lazy Load）與快取策略。初始只載入用戶視野範圍內及主要圖層所需的資料；當用戶平移或縮放到新區域時，再動態載入該區域資料。時間序列預測可在背景預先載入即將播放的幾個時步數據，減少切換時延遲。同時對重複請求的資料（例如過去時刻的觀測值）做緩存，以免頻繁拉取。
- **圖層渲染效能**：控制前端一次渲染的圖形元素數量。對監測站 Marker，可在高縮放時才顯示全部站點，低縮放時用聚合圖示代表。同時使用 Canvas 或 WebGL 批次繪製點比大量 DOM 元素更有效³。例如，可將成千上萬個預測網格點值繪製在一個 canvas 上作熱區圖，而非產生等量的獨立 Marker。Deck.gl 等 WebGL方案能利用GPU加速大量資料點的繪製，提高幀率³。
- **React 更新優化**：將地圖及圖層相關操作與 React 狀態更新解耦，避免不必要的重繪。使用者操作（如切換圖層、拖動時間軸）時，只更新相關的子元件狀態，不整體重載地圖。可利用 `React.memo` 或 `shouldComponentUpdate` 等方式，防止大量不變的Marker元件隨每次狀態改變而重新渲染。特別是整合 Google Maps 後，許多地圖操作實際在外部進行（imperative API），要小心管控不要每次狀態改變都重新 init地圖。
- **地圖互動性能**：對地圖的互動事件作防抖/節流處理。例如監聽地圖 `dragend` 或 `zoom_changed` 事件後延遲執行資料更新，避免使用者拖動過程中持續大量請求新數據。點擊地圖查詢時，可先顯示載入指示，同步在 Web Worker 中計算附近最近監測站或插值該點預測值，以免阻塞主執行緒。
- **資料量與精度權衡**：針對不確定性圖層或高解析度模型輸出，適當降採樣資料以減輕瀏覽器負擔。例如模型輸出如果是10m解析度格網，在可接受的情況下於前端僅取每50m一點進行繪圖，或在後端先行壓縮。使用向量資料（GeoJSON）的情況下，可在後端或載入時簡化多邊形（降低頂點數），透過 `google.maps.Data` 的 `setStyle` 以低細節顯示，除非使用者放大查看才載入高細節版本。這類多層

次細節(Multi-LOD)策略類似於地圖瓦片概念，能確保無論縮放級別如何，都用最小必要量的資料達到良好效果。

- **順暢動態呈現：**當播放預報動畫或時間改變時，採用漸變過渡而非瞬間跳換。可以對連續的圖層（如熱區圖）進行淡入淡出處理，或雙層切換（先將新時間圖層疊加淡入，再移除舊圖層）。這避免頻繁閃爍並提升視覺體驗。若數據更新頻率高，也可限制更新幀率，例如每分鐘最多更新一次圖，以平衡即時性與性能。

透過上述優化措施，整個前端應用即使面對大量即時資料和多圖層互動，依然能保持順暢的操作體驗。良好的效能與互動設計將確保使用者在探索空污地圖時感到反應快速、視覺清晰且功能便利，達成此應用的目標。

總結：本設計方案運用 React 結合 Google Maps，透過清晰的模組劃分與使用流程，配合熱區圖、等值圖、風險機率圖等多元圖層呈現空氣污染的觀測與預測資訊。同時整合多模型輸出、強化不確定性傳達，並採取多項效能優化手段確保地圖互動流暢 ¹ ⁵。如此可為一般大眾提供一個即時且直觀的空污資訊平台，讓使用者能方便地了解空氣品質現況、未來風險並做出更明智的健康決策。

參考資料：

- Tejas K. *Mapping Air Quality Index: A Deep Dive into the AQI Google Maps Project – Open-source air quality map combining Google Maps and real-time AQI data* ² ¹ .
- Annie Preston et al. *Communicating Uncertainty and Risk in Air Quality Maps* – 研究顯示在地圖中包含預測不確定性資訊有助於使用者更謹慎決策 ⁵ ⁶ .
- Lydia Lucchesi. *Vizumap: visualizing uncertainty in spatial data* – 提供多種在地圖上表達不確定性與超標機率的方法 ⁸ ⁷ .
- World Air Quality Index Project. *Air Quality Forecasting Models* – 說明WAQI如何結合多套預報模型形成綜合預測 ⁹ .
- Google Maps Platform Blog. *Streamline the use of Maps JavaScript API within React* – 介紹React整合Google Maps及結合deck.gl強化圖形的方案 ¹⁰ ¹¹ .
- Google Developers. *Google Maps JavaScript API – Heatmaps* – 說明如何使用熱區圖圖層，及引導使用deck.gl進行更高效的熱區渲染 ³ ⁴ .
- Google Maps Platform. *Air Quality API* – 提供超過70種空污指數、逐時預報與歷史數據的API服務 ¹² .

¹ ² Mapping Air Quality Index: A Deep Dive into the AQI Google Maps Project – Tejas Kamble
<https://tejaskamble.com/projects/mapping-air-quality-index-a-deep-dive-into-the-aqi-google-maps-project/>

³ Heatmap Layer | Maps JavaScript API - Google for Developers
<https://developers.google.com/maps/documentation/javascript/heatmaplayer>

⁴ Heatmaps | Maps JavaScript API - Google for Developers
<https://developers.google.com/maps/documentation/javascript/reference/visualization>

⁵ ⁶ Communicating Uncertainty and Risk in Air Quality Maps - PubMed
<https://pubmed.ncbi.nlm.nih.gov/35486550/>

⁷ ⁸ GitHub - lydialucchesi/Vizumap: R package for visualizing uncertainty in spatial data
<https://github.com/lydialucchesi/Vizumap>

⁹ Air Quality Forecasting Models: Real-time and Historical Analysis
<https://aqicn.org/forecast/models/>

10 11 **Blog: Streamline the use of the Maps JavaScript API within your React applications – Google Maps Platform**

<https://mapsplatform.google.com/resources/blog/streamline-the-use-of-the-maps-javascript-api-within-your-react-applications/>

12 **Environment APIs - Google Maps Platform**

<https://mapsplatform.google.com/lp/environment-apis/>