Project Title: uTECH GMNS Software Development

Name: Yuwen Huang

NetID: yh2324

Major, Year: Systems Engineering, M-Eng ’25

In Collaboration with / Special Thanks:

In collaboration with team members in Urban Transportation, Environment, and Community Health Hub (uTECH) and Gao Labs.

Special thanks to Dr. Oliver Gao and Dr. Cafer Avci for their valuable expertise in developing individualized exercise programs, and to Professor David R. Schneider and Professor Wes Hewett for their guidance in optimizing the teaching program.

Tools Report

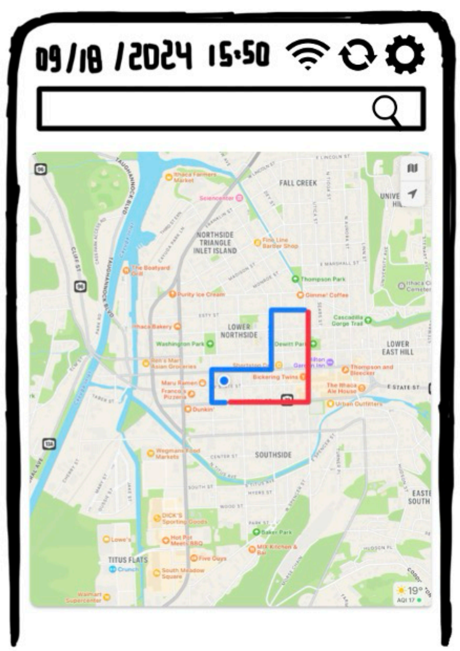
1. Customer Value Proposition

The Customer Value Proposition (CVP) outlines the core value that the Healthy Route App delivers to its users: healthier travel by minimizing exposure to air pollution while considering user priorities such as time and distance. This tool was used at the project's inception to define the app's competitive edge and user appeal. Through the CVP, we learned that users highly value real-time data and route calculation accuracy with personalization features, which informed our focus on integrating tools like AERMOD for air pollution data and user-specific health preferences. By refining this proposition, we better defined the app's primary objectives, such as delivering clear comparative metrics for different routes. The CVP guided further design work by prioritizing features like advanced forecasting and interactive notifications, which align with user needs and support long-term retention strategies.

Customer Value Proposition (CVP)

1. Real-Time Air Pollution Data

The real-time air pollution data feature was added to meet the need for up-to-date information on air quality which enables users to get current pollution levels to avoid unhealthy routes, so the app integrates real-time data from sources like AERMOD to provide the most accurate route recommendations.



2. Pollution-Based Route Calculation

Because of the need to reduce exposure to pollutants during commutes, a pollution-based route calculation feature was added to calculate routes not just based on distance or time but on the lowest pollution exposure, ensuring users can choose the healthiest path available.

3. Real-Time Alerts for Pollution Thresholds

Real-Time Alerts for Pollution Thresholds feature was added to meet the need for safety notifications when air quality worsens, where users are alerted if pollution levels rise during their journey, allowing them to adjust their route accordingly for better health protection.

Figure1. Customer Value Proposition

2. **Annotated Concept Sketch, Storyboarding**

The Annotated Concept Sketch and Storyboarding provided a visual representation of the Healthy Route App's structure and functionality. These tools were used early in the design process to conceptualize the app's structure and flow, ensuring alignment with user expectations. Through this tool, we identified key functional components, such as the route input screen, real-time air quality overlay, and comparison features for pollution exposure. It helped clarify the interaction pathways, like transitioning from inputting start and end points to viewing route options and analytics. Updates to the sketches, based on feedback, highlighted the importance of intuitive layouts, leading to a better-defined user interface and system functionality. This tool prompted additional design work by emphasizing accessibility improvements, including voice-guided navigation, which were integrated to enhance usability for various user demographics.

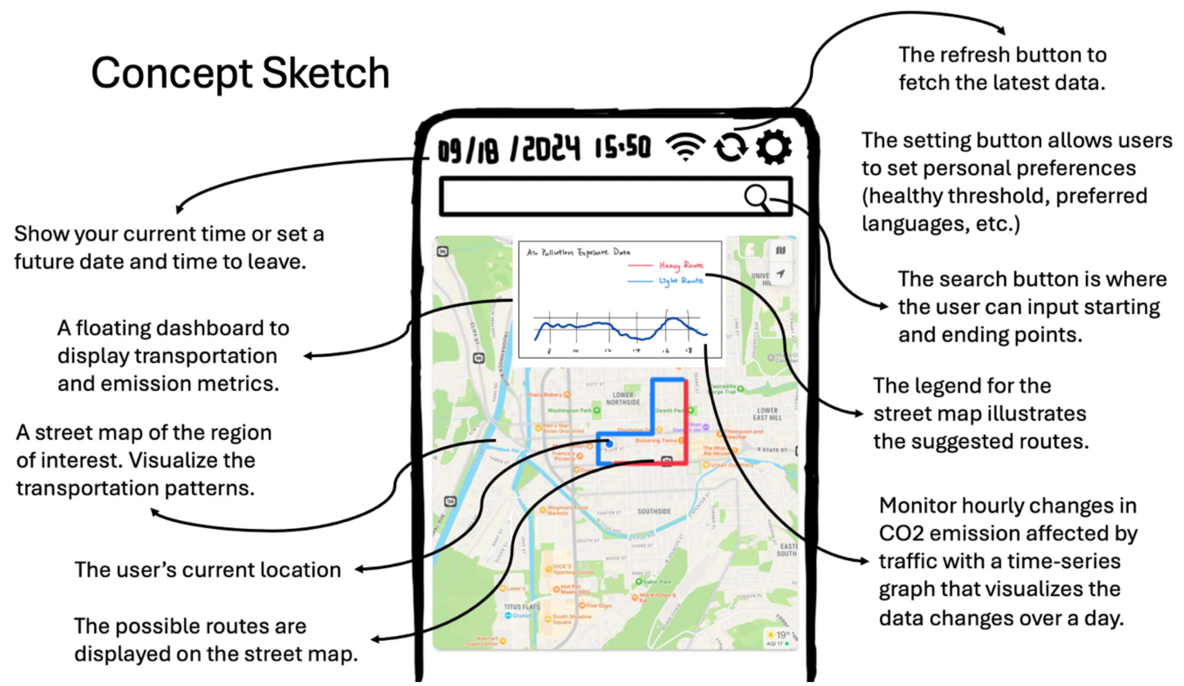


Figure 2. Annotated Concept Sketch

Storyboard

Frame 1: Welcome Screen
• Text: “Welcome to Healthy Route App – Find the Healthiest Path.”
• Visuals: A clean, modern interface with a map background, showing pollution levels color-coded across the city.

Frame 2: Input Start and Destination
• Text: “Enter your start and destination points.”
• Visuals: A search bar where the user types the addresses or selects from a map.

Frame 3: Route Calculation with Pollution Layers
• Text: “Calculating routes based on air pollution exposure.”
• Visuals: The app shows three route options: green for the healthiest, yellow for moderate, and red for the most polluted, with icons representing vehicles and industrial areas.

Frame 4: Route Options Displayed
• Text: “Choose the healthiest route.”
• Visuals: A split screen: the map with routes and a sidebar showing pollution data, estimated exposure time, and a health score for each route.

Frame 5: Healthiest Route Selected
• Text: “You’ve chosen the healthiest path!”
• Visuals: The route is highlighted, showing pollution stats like exposure level (low), estimated time, and air quality details.

Figure 3. Storyboarding

3. **Customer Affinity Process Results**

The Customer Affinity Process was instrumental in grouping user needs and concerns into actionable insights. We used the toy case in the lecture, for example, to imitate the real-world scenario for understating and classifying the customers’ needs. By analyzing survey results and user interviews, we identified critical priorities with three main groups - features, versatilities, and engagement. By categorizing these needs hierarchically, we better defined the system’s feature set, prioritized core functionalities, and revealed deeper insights into different levels of needs. The outcomes guided further design work by informing the development of premium features for the toy, ensuring the toy has both immediate and long-term user requirements.

	A	B	C	D	E	F	G	H
1	Lvl 1 Group Title	# Comments in Each Lvl 1 Group	Lvl 1a Group Title (not all branches of the tree have to be of equal depth)	Sum # of Comments in Each Lvl 1a Group	Group Lvl 2 Titles	Sum # of Comments in Each Lvl 2 Group	Group Lvl 3 Titles (aka Area Titles since they are at the top level)	Sum # of Comments in Each Lvl 3 Group
2	Material Preferred	1			Construction	15	Features	36
3	Durability	4	Robustness	7				
4	Quality	3						
5	Safety Concerns	7						
6	Battery Usage	3			Features	21		
7	Aesthetics	5						
8	Sound	3						
9	Multiple Figures	4						
10	Technological Features	3						
11	Uniqueness	1						
12	Puzzles or Problems	2						
13	Learning Component	6	Inspiring	11	Play Experience	27	Versatility	49
14	Creative Play	5						
15	Indoor/Outdoor Play	2						
16	Independent Play	4						
17	Parent-child Interaction	4						
18	Social Interaction	2						
19	Collectibility	2						
20	Active	2						
21	Assembly	2			Assembly	4		
22	Instruction	2						
23	Portability	2			Portability	4		
24	Storage	2						
25	Compatibility	3			Play Value	7		
26	Affordability	4						
27	Tactile	2			Instructions	7		
28	War Toys	2						
29	Buiding	3						
30	Space	1			Maintainance	5	Engagement	24
31	Supervision	1						
32	Cleaning	2						
33	Packing	1						
34	Functionality	3			Functions	6		
35	Replaceable	1						
36	Customization	2						
37	Accomplishment	1			Achievement	4		
38	Rationale	1						
39	Reality	2						
40	Collaboration/Friendship	3			Union	5		
41	Family Imitation	2						
42	Graphic/Media	2			Look	4		
43	Appearance	2						
44		109				109		109
45								

Figure 4. Customer Affinity Process Result

4. **Context Diagram**

The Context Diagram mapped the Healthy Route App's interactions with external systems, users, and data sources. It was used to visualize the app's ecosystem, including key components like air pollution data providers (e.g., AERMOD), geolocation services, and user devices. By creating a diagram with at least eight external blocks, we learned how data flows into and out of the system, highlighting dependencies such as real-time data accuracy and connectivity requirements. This tool helped better define the app's interfaces and communication protocols, such as API integrations for data retrieval. It prompted further design work to address risks related to connectivity interruptions, leading to the implementation of offline functionality and local data caching.

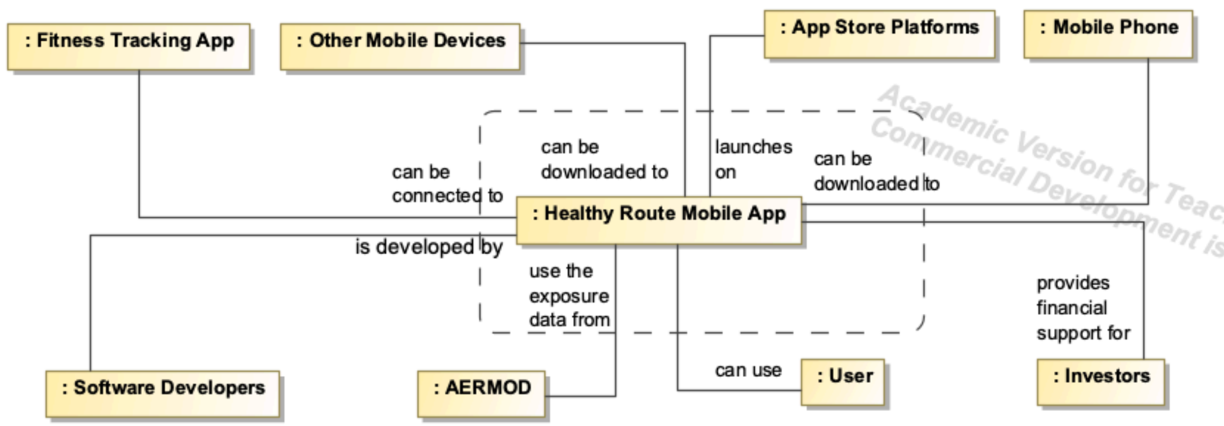


Figure 5. Context Diagram

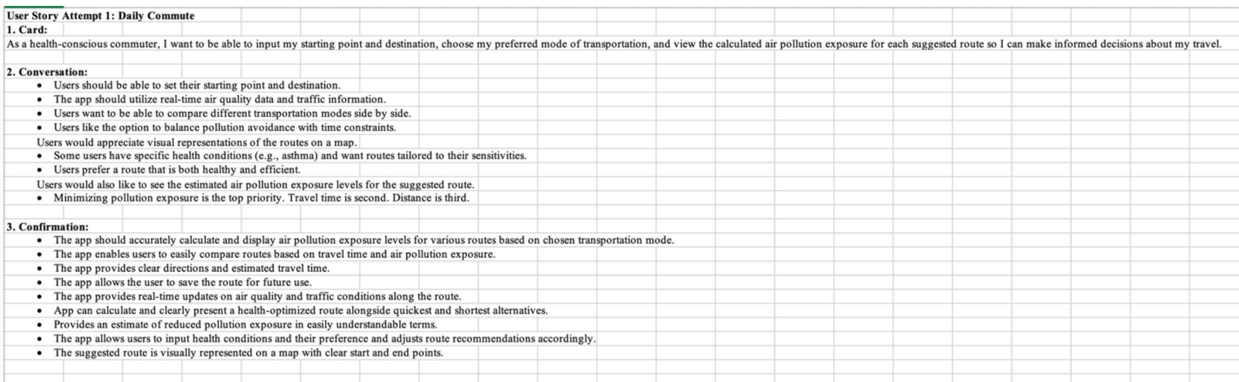
5. Use Cases & User Story

The use cases and user stories captured detailed scenarios of how users interact with the Healthy Route App, covering at least 20 specific cases and one overarching story. These tools revealed critical system behaviors, such as handling location inputs, traffic modes selection, and alerting users about high pollution levels on their selected routes. By formalizing these interactions, we were able to better define system requirements, including input validation and real-time alert systems. The insights gained from these use cases prompted further design work on error-handling mechanisms and robust notification frameworks to ensure a seamless user experience.

Use Case	Priority
User opens the app	H
User enters the app home page	H
User chooses preferred language	H
User inputs starting and ending points	H
User enters different traffic modes (e.g., walking, cycling, public transit, driving)	M
User receives calculated exposure to traffic-related air pollution of each route	H
User chooses to accept the route or request alternatives	M
User saves the suggested route for later use	M
User competes with others in-app and earns rewards for choosing healthy routes	L
User shares the route with friends or on social media	L
User sets preferences (e.g., max acceptable detour time, preferred road types)	L
User enters desired arrival time and location	M

User defines personal air quality thresholds	M
User gets alert when selected route exceeds threshold	M
User gets the historical data to suggest healthiest route	L
User receives suggestions for optimal departure times to minimize exposure	L
User connects fitness tracking device	L
User receives personalized route recommendations based on health goals	L
User sets up daily notifications	L
User views personal air pollution exposure trends over a period	L

Table 1. Use Cases



User Story Attempt 1: Daily Commute

1. Card:

As a health-conscious commuter, I want to be able to input my starting point and destination, choose my preferred mode of transportation, and view the calculated air pollution exposure for each suggested route so I can make informed decisions about my travel.

2. Conversation:

- Users should be able to set their starting point and destination.
- The app should utilize real-time air quality data and traffic information.
- Users want to be able to compare different transportation modes side by side.
- Users like the option to balance pollution avoidance with time constraints.

Users would appreciate visual representations of the routes on a map.

- Some users have specific health conditions (e.g., asthma) and want routes tailored to their sensitivities.
- Users prefer a route that is both healthy and efficient.

Users would also like to see the estimated air pollution exposure levels for the suggested route.

- Minimizing pollution exposure is the top priority. Travel time is second. Distance is third.

3. Confirmation:

- The app should accurately calculate and display air pollution exposure levels for various routes based on chosen transportation mode.
- The app enables users to easily compare routes based on travel time and air pollution exposure.
- The app provides clear directions and estimated travel time.
- The app allows the user to save the route for future use.
- The app provides real-time updates on air quality and traffic conditions along the route.
- App can calculate and clearly present a health-optimized route alongside quickest and shortest alternatives.
- Provides an estimate of reduced pollution exposure in easily understandable terms.
- The app allows users to input health conditions and their preference and adjusts route recommendations accordingly.
- The suggested route is visually represented on a map with clear start and end points.

Figure 6. User Story

6. **SysML Use Case Diagram**

The SysML Use Case Diagram provided a graphical representation of the app's functional capabilities, capturing at least ten use cases, such as route calculation, geolocation access, and data synchronization. This tool was used to communicate system functionality to stakeholders and identify key relationships between various use cases and sub-use cases. Assigning high, medium, and low priorities to different use cases clearly showed the importance level, which is significantly helpful in encompassing all the needed design ideas. It also led to a better definition of functional allocations, ensuring all critical features were accounted for. The diagram guided further development by revealing potential overlaps in subsystem responsibilities, prompting refinements in data flow management.

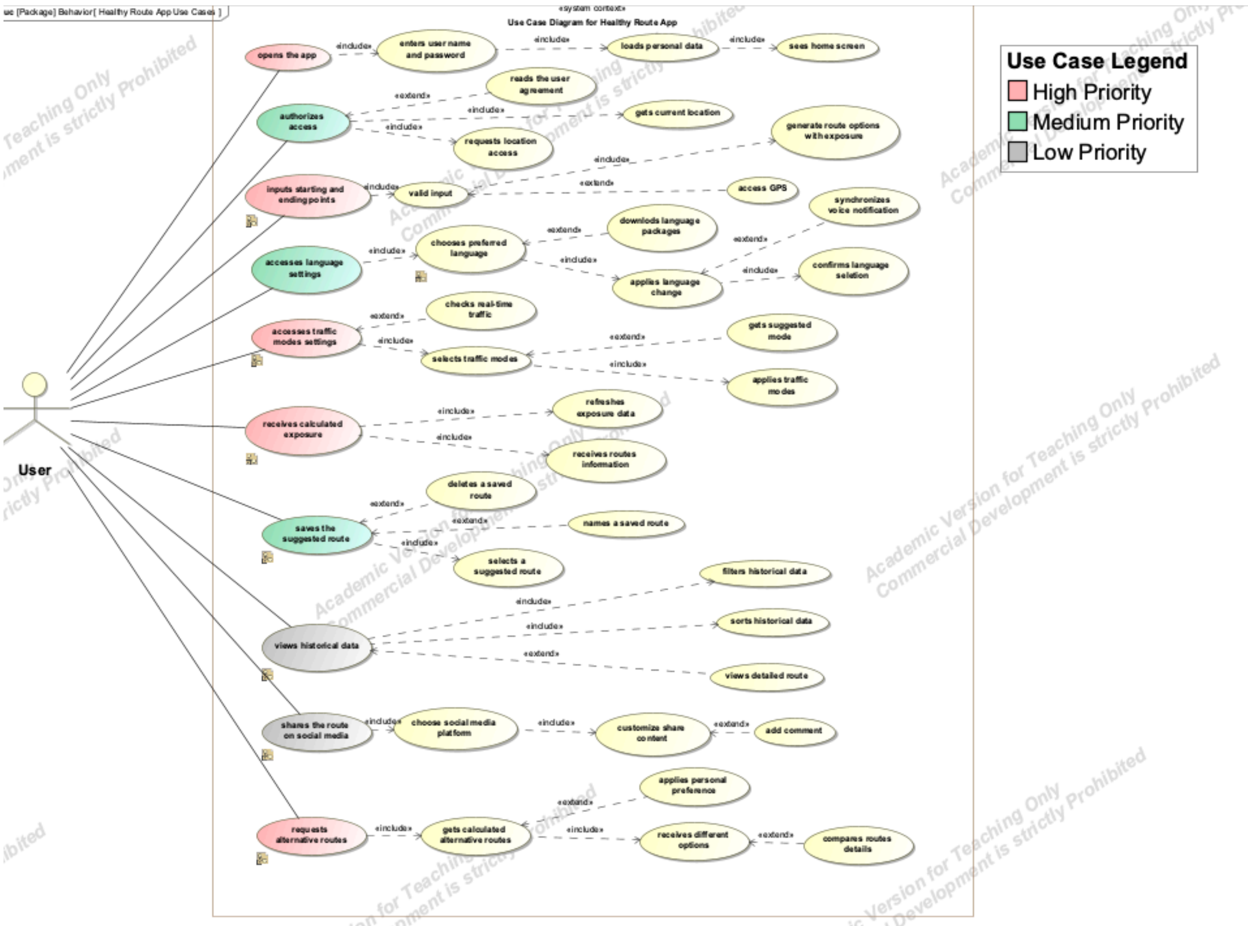


Figure 7. Use Case Diagram

7. **Use Case Behavioral Diagrams**

The Use Case Behavioral Diagrams modeled the app’s behavior for four key use cases, such as calculating the healthiest route or alerting users about high pollution. These diagrams were used to visualize step-by-step system responses, helping identify potential delays or bottlenecks in processes like user input starting and ending points. By detailing interactions between the operator and system, we learned how specific inputs (e.g., user locations) propagate through the system. This defined timing constraints for actions like displaying route options or generating alerts. These diagrams informed further works by highlighting the need for asynchronous data fetching to maintain system responsiveness.

Use Case 1: User opens the app		
Initial Conditions		
1. The app is not running (it is closed but not the first time using).		
Operator (User)	System (App)	User Interface
The user taps the app icon.		
	The system shall detect the tap input.	
	The system shall initiate app launch sequence.	
	The system shall begin loading necessary resources.	
		The loading screen or splash screen is displayed.
The user enters user name and password.		
	The system shall verify user credentials.	
	The system shall transite to the main interface.	
		The main interface (home screen) is
Ending Conditions		
1. The app is now running and ready for user interaction on the main screen.		
Notes		
1. Assume the system has internet access for authentication and loading data.		
2. The splash screen is shown while the app loads in the personal information and background image.		

Figure 8. Use case Behavior Diagram 1

Use Case 2: User Inputs Starting and Ending Points			
Initial Conditions			
1. The system is waiting for user input, with no active route.			
Operator (User)	System (Navigating App)	Routing	
The user already opens the navigation app.			
	The system shall display the interface for inputting locations.		
The user types in a starting point.			
	The system shall store the starting point information.		
		The starting point of this route is defined but not processed.	
	The system shall validate if the starting point is a valid location.		
The user types in an ending point.			
	The system shall store the ending point information.		
		The ending point of this route is defined but not processed.	
	The system shall validate if the ending point is a valid location.		
	The system shall prepare to calculate the exposure of each possible route from the starting point to the end point		
Ending Conditions			
1. The starting and ending points are successfully input and validated by the system.			
2. The system is ready to display route options.			
Notes			
1. Assume the user inputs valid locations that the app can process.			

Figure 9. Use case Behavior Diagram 2

Use Case 3: User Receives Real-Time Pollution Alert		
Initial Conditions:		
1. The system is actively monitoring pollution levels during the user's navigation along a selected route.		
Operator (User)	System (Navigating App)	Routing
The user is navigating along a selected route.		
	The system shall continuously monitor the air pollution levels along the selected route.	
The pollution levels on the route exceed a predefined threshold.		
	The system shall detect that pollution levels have reached critical values.	
	The system shall generate an alert to notify the user.	
		The app prepares to recalculate the route or provide alternate options based on lower pollution exposure.
The user receives a real-time alert about the high pollution levels.		
	The system shall display the alert on the screen and suggest alternative, healthier routes.	
The user selects one of the suggested alternative routes.		
	The system shall update the user's current route to the newly selected one.	
		The system recalculates the exposure and updates navigation.
Ending Conditions:		
1. The system successfully switches the route based on the user's selection and recalculates the healthiest path.		
2. The system continues to monitor air pollution levels on the updated route.		
Notes:		
1. Assume the user has enabled real-time alerts and route recalculation.		
2. Pollution thresholds and alert settings are customizable by the user in the app settings.		

Figure 10. Use case Behavior Diagram 3

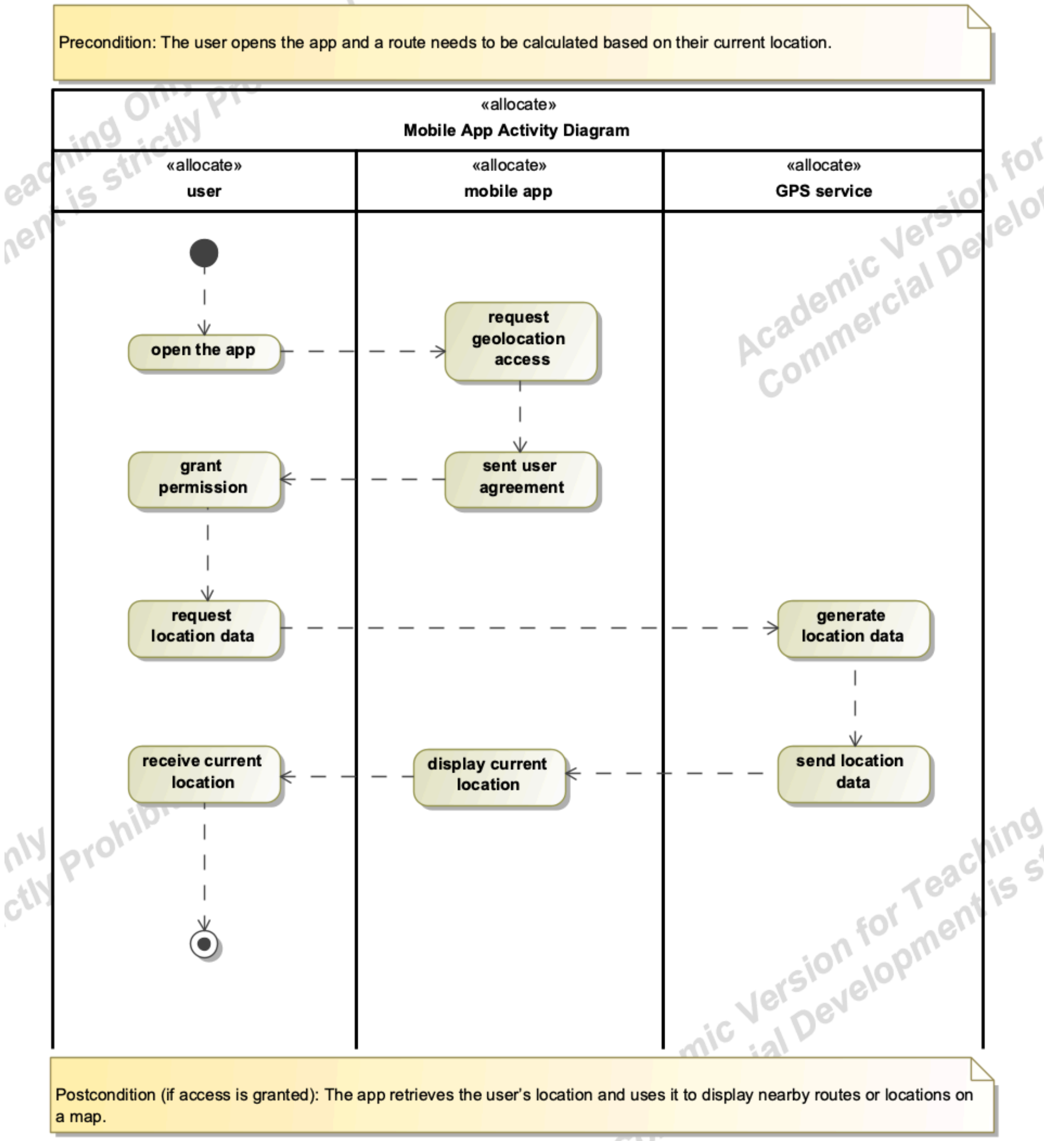


Figure 11. Use case Behavior Diagram 4

8. **SysML Activity Diagram (Behavioral Dia in SysML)**

The SysML Activity Diagram captured the detailed workflow of a key behavior, for instance, requesting and accessing the user's geolocation. It was used to analyze the sequence of actions, starting from receiving user actions to overlaying location access requests and presenting preference options. This tool helped us identify dependencies, such as the need for data privacy availability. Additionally, it facilitated a better understanding of system operations by specifying critical paths and fallback options for incomplete data. The activity diagram has provided a valuable opportunity to enhance our design approach, focusing on creating smoother transitions between each step. This is especially important in cases where geolocation access may be interrupted, allowing us to ensure a more seamless experience.

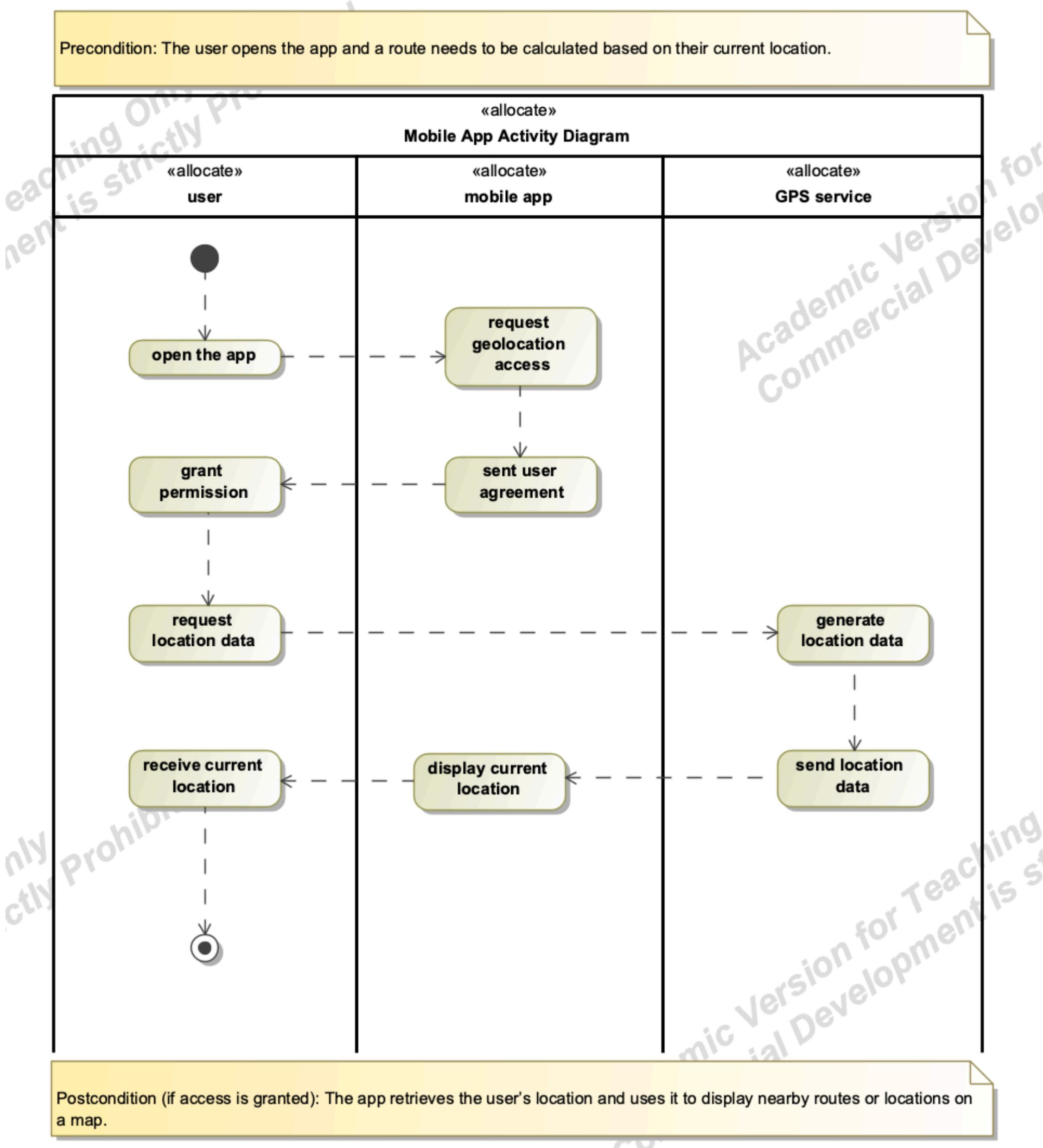


Figure 12. SysML Activity Diagram (Behavioral Dia in SysML)

#	Name	Text
1	1 Detect	OR.1 The system shall detect the tap input.
2	2 Initiate	OR.2 The system shall initiate app launch sequence.
3	3 Loading resources	OR.3 The system shall begin loading necessary resources.
4	4 Verify user	OR.4 The system shall verify user credentials.
5	5 Transite	OR.5 The system shall transmit to the main interface.
6	6 Display input blank	OR.6 The system shall display the interface for inputting locations.
7	7 Store location	OR.7 The system shall store the starting point information.
8	8 Validate location	OR.8 The system shall validate if the starting point is a valid location.
9	9 Store location	OR.9 The system shall store the ending point information.
10	10 Validate location	OR.10 The system shall validate if the ending point is a valid location.
11	11 Preparing calculaltio	OR.11 The system shall prepare to calculate the exposure of each possible route from the starting point to the end point
12	12 Calculating exposur	OR.12 The system shall calculate the exposure of each possible route
13	13 Send request	OR.13 The system shall request geolocation access from the user.
14	14 Display pop-ups	OR.14 The system shall display the geolocation access permission pop-ups.
15	15 Receive confirmation	OR.15 The system shall receive confirmation of granted access.
16	16 Activate	OR.16 The system shall activate its location-based features.
17	17 Integrate	OR.17 The system shall integrate real-time air pollution data (e.g., from AERMOD or other sources) to overlay pollution concentration on the map.
18	18 Alert	OR.18 The system shall send real-time alerts if the pollution levels on a chosen route exceed a certain threshold.
19	19 Visualize	OR.19 The system shall display multiple routes with respective pollution exposure levels, allowing users to visually compare options.
20	20 Comparison	OR.20 The system shall provide a basic side-by-side comparison of different routes, but the comparisons included, such as time, distance or pollution levels, have not yet to be fully defined.

Figure 13. Req. Table

9. **Originating & Derived Requirement Tables**

The Originating and Derived Requirement Tables formalized over 30 originating requirements, such as validating user locations and integrating air pollution data, along with all derived requirements linked to specific use cases. These tables were used to ensure traceability between high-level goals and system functionalities. By organizing these requirements, we learned about dependencies and constraints, such as the need for seamless API integration with data sources. It better defined functional and non-functional requirements, ensuring alignment with user needs. This tool prompted further design work by prioritizing critical requirements for initial implementation while deferring less urgent features to later development phases.

Index	Originating Requirements	Abstract Name	Issues	Resolution
OR.1	The system shall detect the tap input.	Detect		
OR.2	The system shall initiate app launch sequence.	Initiate		

OR.3	The system shall begin loading necessary resources.	Loading resources	What kinds of resources is needed? It needs to specify the detailed information.	The system shall begin loading necessary resources, INCLUDING Web Server, Database Server, backend framework, Mapping APIs, Geolocation API, Authentication Service, Caching System, etc.
OR.4	The system shall verify user credentials.	Verify user		
OR.5	The system shall transit to the main interface.	Transit		
OR.6	The system shall display the interface for inputting locations.	Display input blank		
OR.7	The system shall store the starting point information.	Store location		
OR.8	The system shall validate if the starting point is a valid location.	Validate location	Issue: How about if the input is not a valid location? It need to provide the solution for this scenario.	The system shall VALIDATE if the starting point is a valid location or PROMPT the user that this is an unavailable location.
OR.9	The system shall store the ending point information.	Store location		
OR.10	The system shall validate if the ending point is a valid location.	Validate location		
OR.11	The system shall prepare to calculate the exposure of each possible route from the starting point to the end point	Preparing calculation		
OR.12	The system shall calculate the exposure of each possible route	Calculating exposure		
OR.13	The system shall request geolocation access from the user.	Send request		
OR.14	The system shall display the geolocation access permission pop-ups.	Display pop-ups		
OR.15	The system shall receive confirmation of granted access.	Receive confirmation		
OR.16	The system shall activate its location-based features.	Activate		
OR.17	The app shall integrate real-time air pollution data (e.g., from AERMOD or other sources) to overlay pollution concentration on the map.	Integrate		
OR.18	The system shall send real-time alerts if the pollution levels on a chosen route exceed a certain threshold.	Alert	The format of the alert is not well defined, is it a pop-up, a push or something else? Also, the consequence of the alert is not yet fully defined.	The system shall send PUSH ALERT detailing specific pollutant levels and their severity when air quality on a user's chosen route exceeds personalized thresholds.
OR.19	The system shall display multiple routes with respective pollution exposure levels, allowing	Visualize		

	users to visually compare options.			
OR.20	The system shall provide a basic side-by-side comparison of different routes, but the comparisons included, such as time, distance or pollution levels, have not yet to be fully defined.	Comparison		
OR.21	The system shall allow users to create and manage profiles that store personal preferences such as route preferences (e.g., prioritizing low pollution, time, or distance).	Set preferences		
OR.22	The system shall allow users to input health-related information (e.g., respiratory conditions).	Customization		
OR.23	The system shall incorporate air pollution forecasting for future time periods.	Forecasting		
OR.24	The system shall provide users with analytics on their route history, including cumulative pollution exposure, etc.	Loading history		
OR.25	The system shall allow users to input multiple stops along their route, calculating the healthiest path that passes through all chosen points.	Enable stops on route		
OR.26	The system shall allow users to share their healthiest routes with friends or the public, encouraging the adoption of pollution-aware routes within communities.	Sharing		
OR.27	The system shall synchronize user data across multiple devices.	Synchronize information		
OR.28	The system shall provide voice-guided navigation for the healthiest routes, giving users hands-free interaction during their journey.	Voice guiding		
OR.29	The system shall include a settings page where users can choose preferred language.	Choosing language		
OR.30	The system shall allow users to toggle basic map layers such as traffic, satellite view, or a simplified map.	Setting map view		

Table 2. Originating Requirements

Index	Derived Functional Requirement	Function Name	Source OR
DR.1	The system shall initiate the app launch sequence.	Initiate App Launch	**OR.2**
DR.2	The system shall load essential resources for air pollution data and routing.	Load Resources	**OR.3**
DR.3	The system shall display fields for inputting a starting and ending location.	Display Input Fields	**OR.6**
DR.4	The system shall validate the starting and ending locations.	Validate Locations	**OR.8**
DR.5	The system shall store the validated locations in memory for routing.	Store Locations	**OR.7**
DR.6	The system shall calculate the pollution exposure for each possible route.	Calculate Route Exposure	**OR.12**
DR.7	The system shall prompt the user for geolocation access.	Request Geolocation Access	**OR.13**
DR.8	The system shall display multiple route options with respective pollution exposure levels.	Display Route Options	**OR.19**
DR.9	The system shall notify the user with a pop-up alert if pollution levels on a chosen route exceed a specified threshold.	High Pollution Alert	**OR.18**
DR.10	The system shall enable users to add multiple stops on a route.	Multi-Stop Routing	**OR.25**
DR.11	The system shall enable users to adjust route preferences (e.g., prioritize time, distance, or air quality).	Adjust Route Preferences	**OR.21**
DR.12	The system shall display a visual map overlay of pollution concentrations along the route in real time.	Pollution Map Overlay	**OR.17**
DR.13	The system shall provide a comparison summary for different routes, including pollution exposure and other metrics.	Route Comparison Summary	**OR.20**
DR.14	The system shall allow users to toggle between different map views, including traffic, satellite, and pollution overlay.	Toggle Map Views	**OR.30**
DR.15	The system shall offer voice guidance for navigation on the selected route.	Voice Guidance	**OR.28**
DR.16	The system shall display pollution exposure statistics and analytics based on the user's cumulative route history.	Show Exposure Analytics	**OR.24**
DR.17	The system shall enable users to share their selected healthy routesacross different apps.	Share Routes	**OR.26**
DR.18	The system shall synchronize user preferences and route data across all devices linked to the user's account.	Sync Across Devices	**OR.27**
DR.19	The system shall allow users to manage profiles with customizable preferences.	Manage User Profiles	**OR.21**
DR.20	The system shall display options for users to input health-related information, such as respiratory conditions.	Input Health Information	**OR.22**
DR.21	The system shall include options in the settings menu for users to select their preferred language.	Set Language Preferences	**OR.29**

Figure 14. Derived Requirements

10. **SysML Requirements Table**

The SysML Requirements Table was used to map requirements derived specifically from the SysML Activity Diagram, ensuring that behavioral workflows aligned with system expectations. This table focused on critical requirements such as real-time geolocation accuracy, pollution data integration, and the display of comparative route metrics. Through this process, we better defined the technical dependencies for achieving these behaviors, such as the precision of pollution overlay algorithms. The table also highlighted gaps in meeting derived requirements, which led to further design work, including exploring more robust APIs for pollution data and designing fallback mechanisms for scenarios where real-time data is unavailable.

#	Name	Text
1	1 Detect	OR.1 The system shall detect the tap input.
2	2 Initiate	OR.2 The system shall initiate app launch sequence.
3	3 Loading resources	OR.3 The system shall begin loading necessary resources.
4	4 Verify user	OR.4 The system shall verify user credentials.
5	5 Transite	OR.5 The system shall transmit to the main interface.
6	6 Display input blank	OR.6 The system shall display the interface for inputting locations.
7	7 Store location	OR.7 The system shall store the starting point information.
8	8 Validate location	OR.8 The system shall validate if the starting point is a valid location.
9	9 Store location	OR.9 The system shall store the ending point information.
10	10 Validate location	OR.10 The system shall validate if the ending point is a valid location.
11	11 Preparing calculaltio	OR.11 The system shall prepare to calculate the exposure of each possible route from the starting point to the end point
12	12 Calculating exposur	OR.12 The system shall calculate the exposure of each possible route
13	13 Send request	OR.13 The system shall request geolocation access from the user.
14	14 Display pop-ups	OR.14 The system shall display the geolocation access permission pop-ups.
15	15 Receive confirmation	OR.15 The system shall receive confirmation of granted access.
16	16 Activate	OR.16 The system shall activate its location-based features.
17	17 Integrate	OR.17 The system shall integrate real-time air pollution data (e.g., from AERMOD or other sources) to overlay pollution concentration on the map.
18	18 Alert	OR.18 The system shall send real-time alerts if the pollution levels on a chosen route exceed a certain threshold.
19	19 Visualize	OR.19 The system shall display multiple routes with respective pollution exposure levels, allowing users to visually compare options.
20	20 Comparison	OR.20 The system shall provide a basic side-by-side comparison of different routes, but the comparisons included, such as time, distance or pollution levels, have not yet to be fully defined.

Figure 15. SysML Requirements Table

11. **Concept Fragment Generation** Enough to support your one combination table

Concept Fragment Generation provided the foundation for brainstorming and exploring various potential design solutions for subsystems like real-time adaptation, connectivity, and user notifications. For example, one fragment focused on caching air pollution data to enable offline functionality, while another considered predictive algorithms for route suggestions based on historical trends. This tool helped us learn about innovative ways to tackle challenges such as intermittent connectivity or delayed updates. It better defined the design of each subsystem by narrowing down feasible approaches. This process led to further work on refining solutions into actionable components, eventually integrating them into the morphology box.

Constant Name	Est. Date	Value	Units	Source	OR 1	OR 2	OR 3	OR 4	OR 5	OR 6	OR 7	OR 8	OR 9	OR 10	OR 11	OR 12	OR 13	OR 14	OR 15	OR 16	OR 17	OR 18	OR 19	OR 20
MAX_ROUTE_DISTANCE	10/22/24	100	km	City Planning Dept.											X	X							X	X
MIN_POLLUTION_ALERT	10/15/24	150	AQI	EPA Guidelines												X					X	X		
GPS_ACCURACY	10/8/24	10	m	Device Spec						X	X	X	X	X			X	X	X	X				
MAX_CALCULATION_TIME	10/15/24	5	sec	UX Requirements												X							X	X
DATA_UPDATE_INTERVAL	11/5/24	300	sec	API Documentation		X															X	X		
SESSION_TIMEOUT	11/19/24	1800	sec	Security Policy			X																	
MIN_ROUTE_DIFFERENCE	10/22/24	0.5	km	UX Focus Group																			X	X

Figure 16. Concept Fragments

12. **Combination Tables / Morphology Boxes**

The Combination Tables (or Morphology Boxes) tool systematically explored solutions for the Healthy Route App by combining air pollution data sources—such as AERMOD layers, mobile sensors, fixed monitoring stations, and satellite data—with route suggestion methods like exposure minimization algorithms, traffic assignment solvers, multi-criteria path optimization, and machine learning models. Key combinations, such as AERMOD with exposure minimization algorithms or mobile sensors with traffic assignment solvers, balanced data accuracy, computational efficiency, and user experience. This process guided design refinements, ensuring effective integration of diverse data sources with robust route optimization methods.

Concept combination table:	
The system shall fetch air pollution data.	**The system shall suggeted the healthiest route.**
AERMOD concentration layer	Exposure minimization algorithm
Mobile air quality sensors	Traffic assignment solver
Fixed monitoring stations	Multi-criteria path optimization
Satellite-based air quality data	Machine learning prediction model
Combine concepts:	
The system shall fetch air pollution data.	**The system shall suggeted the healthiest route.**
AERMOD concentration layer and exposure minimization algorithm	
Mobile air quality sensors and traffic assignment solver	
Fixed monitoring stations and multi-criteria path optimization	
Satellite-based air quality data and machine learning prediction model	
AERMOD concentration layer and machine learning prediction model	
Mobile air quality sensors and exposure minimization algorithm	
NOTES:	
All concepts in the left table can be combined with concepts in the right table to create comprehensive solutions for the Healthy Route App.	

Figure 17. Combination Table

13. **Functional Flow Block Diagram**

The Functional Flow Block Diagram (FFBD) outlined the sequence and dependencies of key system operations, with a minimum of ten blocks covering activities like app initialization, route calculation, and user notification. It was used to ensure that all functions logically flowed and were interconnected without bottlenecks. The system then concurrently retrieves air pollution data (e.g., from AERMOD layers) and real-time traffic information to generate possible routes. For each route, pollution exposure is calculated, and exposures are compared iteratively to identify the healthiest option. The route with the lowest exposure is saved and displayed on the map, enabling the user to transition seamlessly to navigation mode. This structured flow ensures the app delivers accurate, efficient, and user-friendly route suggestions. This diagram can better define system timing requirements and error-handling pathways as well.

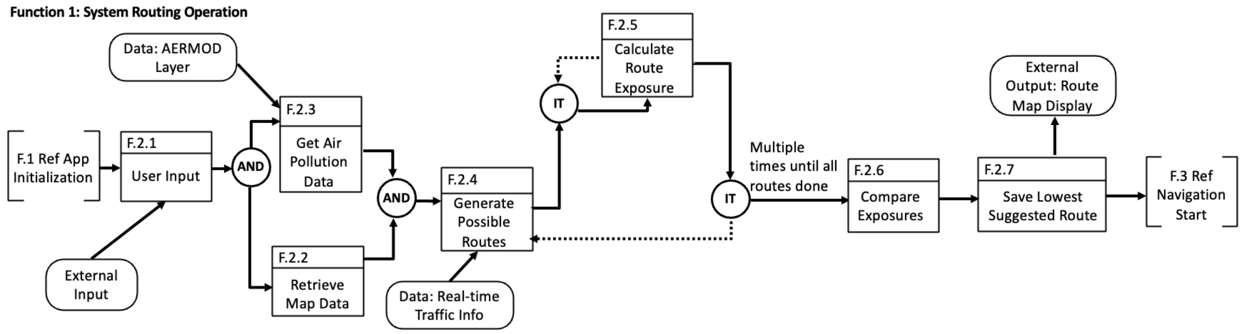


Figure 18. FFBD

14. **Activity Diagram**

The activity diagram for the Healthy Route App visually represents the user interaction and system activities necessary to generate the healthiest route. It starts with the user opening the app. It progresses through sequential activities, including entering starting and ending points, verifying inputs, querying external APIs for air quality and mapping data, processing the fetched data, and calculating potential routes. Once the exposure levels for each route are generated, the app displays the suggested routes, allowing the user to view and select their preferred option. This diagram highlights decision points, such as input validation, and illustrates the app's reliance on external data sources and processing workflows.

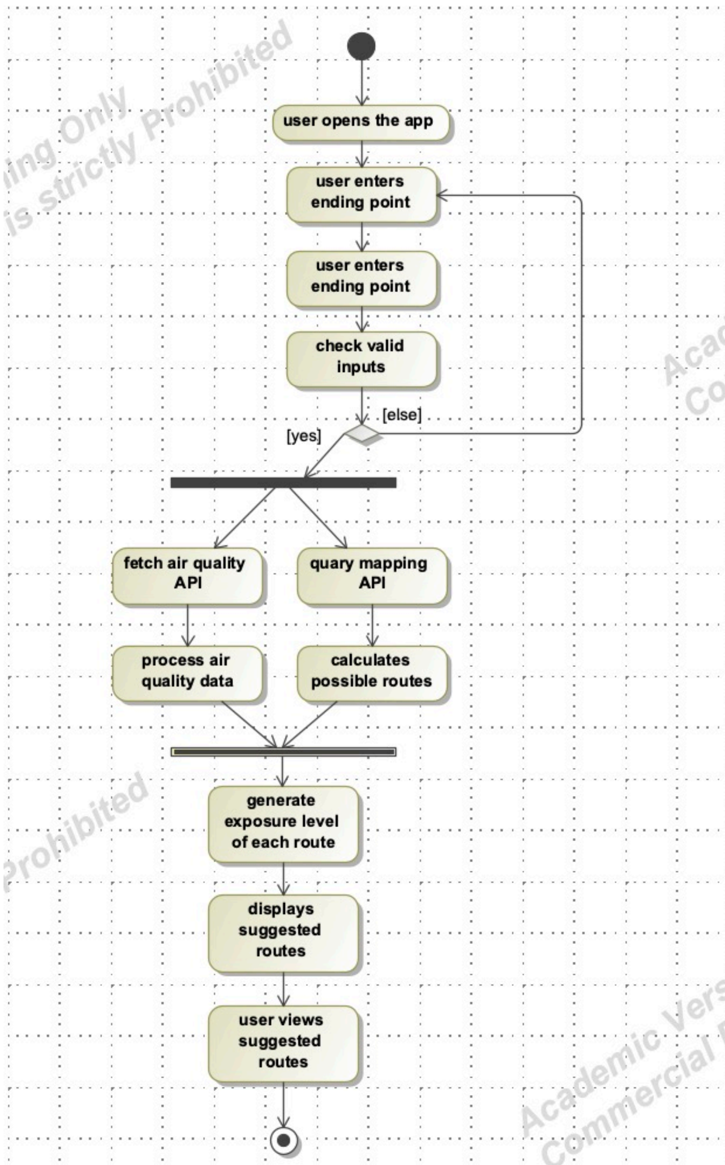


Figure 19. Activity Diagram

15. **Decision Matrixes**

The Decision Matrix was used to evaluate design alternatives against predefined criteria, including accuracy, user experience, scalability, and cost. For example, different algorithms for pollution data interpolation were scored based on performance and resource efficiency. This process revealed the strengths and weaknesses of each option, leading to a better definition of trade-offs in subsystem implementation. It prompted further design work by prioritizing high-scoring solutions, such as machine learning-based predictions for air quality, while flagging lower-priority features for future enhancements.

Criteria	Values: Option A	Values: Option B	Values: Option C	Min Value	Max Value	Normalized Values: Option A	Normalized Values: Option B	Normalized Values: Option C	Weight	Final Scores: Option A	Final Scores: Option B	Final Scores: Option C
Accuracy of Route Calculation	8	7	9			0.88	0.78	1.00	0.15	0.13	0.12	0.15
Make a Simple User Interface	9	6	5			1.00	0.67	0.56	0.15	0.15	0.10	0.08
Air Pollution Data Integration	7	9	8			0.78	1.00	0.88	0.15	0.12	0.15	0.13
Traffic Data Integration	8	6	9			0.88	0.67	1.00	0.1	0.09	0.07	0.10
Customization (5 is best)	5	3	4			1.00	0.60	0.80	0.05	0.05	0.03	0.04
Real-time Adjustments	6	7	5			0.86	1.00	0.71	0.1	0.09	0.10	0.07
Quick Input Response (Second)	0.1 second	0.5 second	0.7 second			0.14	0.71	1.00	0.2	0.03	0.14	0.20
Correct GPS location	8	7	9			0.88	0.78	1.00	0.1	0.09	0.08	0.10
						6.42	6.21	6.95	1	0.74	0.78	0.88

Notes

Definition of Criteria

Accuracy of Route Calculation: Measures how well the app calculates the healthiest route based on pollution data.

Make a Simple User Interface: Evaluates the ease with which users can input starting and ending points.

Air Pollution Data Integration: How well the app incorporates and updates pollution data in real-time.

Traffic Data Integration: The effectiveness of including traffic data to assess air pollution exposure.

Customization: The app's ability to allow users to adjust route preferences (e.g., avoiding highways or heavily polluted areas).

Real-time Adjustments: The app's capability to alter routes based on changing conditions like weather or sudden pollution spikes.

Quick Input Response: Focuses on how quickly the app processes the input and provides a route.

Correct GPS location: Measures how accurately the app calculates the geolocation based on the user's current location.

Customization Performance Measurement Scale

Scale Measure for Customization

Score	Scale Condition
5	Users can fully customize route preferences (e.g., pollution threshold, traffic conditions, weather impacts) and receive real-time updates. The app offers personalized suggestions and allows detailed route exclusions (e.g., specific streets or neighborhoods). All settings can be saved and applied automatically.
4	Users can customize most route preferences, with real-time updates available for the majority of options. Users can exclude general areas (e.g., highways) but may not exclude specific streets. Some settings can be saved for future use.
3	Users can customize the key route preferences (e.g., avoid highways or heavily polluted areas), but real-time updates are limited to a few options. Customization is available at a higher level but lacks fine detail like excluding specific streets. Settings must be manually re-entered each time.
2	Customization options are limited to broad route preferences (e.g., pollution or shortest path), and real-time updates are only available for traffic, not air quality or weather. Users cannot exclude specific areas, and settings cannot be saved.
1	Users can only choose between preset route preferences (e.g., healthiest route or shortest route) with no real-time updates and minimal control over customization. No options to exclude areas or adjust based on specific conditions.

Graph

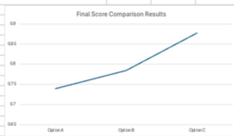


Figure 20. Decision Matrix

16. **Goal – Question – Metric GQM**

The GQM Framework established nine goals, each supported by three specific questions and metrics, to assess the app's success. Goals included reducing user exposure to air pollution, ensuring the accuracy of air quality data, making GPS positioning accurate during use, making the route navigation visually clear, integrating smoothly with other health apps, easy customization for user preferences, enhancing user satisfaction and retention, improve app performance and stability, and allowing users to easily change transportation methods. This process helped better define performance benchmarks for the system. It led to the following development process by identifying gaps in meeting specific metrics, such as implementing additional validation steps for pollution data sources to improve reliability.

Goals	Questions	Ideal Metric	Approximate Metric	Data Collection Method
Reduce user exposure to air pollution	Does the app consistently suggest routes with lower air pollution levels?	Percentage reduction in average air pollution exposure compared to shortest route	Difference in calculated air quality index (AQI) between suggested route and shortest route	Log route suggestions and compare AQI data
	How often do users save the suggested healthier route?	Percentage of times users select the suggested healthier route	Ratio of healthier routes selected to total routes suggested	App background usage analytics
Ensure accuracy of air quality data	How closely does our air quality data match official measurements?	Correlation coefficient between app data and official monitoring station data	Percentage of times app data is within 10% of official data	Regular comparison with official air quality data sources
	How quickly does the app update air quality information?	Average time between environmental data updates	Time stamp difference between current time and last data update	System logs of data update times
Make GPS positioning accurate during use				
Make the route navigation visually clear				
Integrate smoothly with other health apps	Can users connect their fitness trackers to the app?	Number of successful integrations with popular fitness tracking apps	Percentage of users who have linked a fitness device	Integration logs and user account data
	How does route choice correlate with other health indicators?	Correlation between route AQI and user-reported health symptoms	Changes in average step count or heart rate on healthier routes	Integrated fitness data and user health logs
Easy customization for user preferences	How well does the app adapt to individual user preferences?	Percentage of users who report the app suggestions match their preferences	Number of times users adjust suggested routes	User preference settings and route modification logs
	Do users with different health threholds receive appropriate route suggestions?	Accuracy of route recommendations based on user health profiles	User satisfaction ratings from those with declared health conditions	Health profile data and preference from user settings
Enhance user satisfaction and retention				
Improve app performance and stability				
Allow users to easily change transportation methods				

Figure 21. Goal-Question-Metric GQM

17. **Analytical Hierarchy Process (applied to Goals)** All goals, 3-5 at the top level

The Analytical Hierarchy Process prioritized the nine GQM goals by comparing them across criteria like user impact, technical feasibility, and cost. For instance, improving route accuracy was ranked higher than providing multi-stop features in the initial rollout. This process helped clarify which goals were most critical to address early in development. It better defined resource allocation for system enhancements, prompting further design work to focus on the highest-priority goals, such as real-time data processing and user notification systems.

1	Make the app calculate air pollution exposure accurately		Make the app generate route options correctly		Make a user friendly interface in the app	Other (agile integrating, easily fixed, user preference, sharing function)			
	0.2		0.3		0.2	0.3			
2						Make the app easy to customize (user preference, sharing function)		Make the app technical feasible (agile integrating, easily fixed)	
						0.6		0.4	
3	Make the app fetch air pollution data accurately.	Make the app access the user's geolocation accurately.	Make the app find each possible routes quickly anfd correctly.	Make the app integrate air pollution level with each path efficiently.	Make the app have a user friendly interface in route navigating.	Make the app easy to set personal preference.	Make the app correctly link to the sharing apps.	Make tha app easy to be the fixed timely.	Make the app agile in integrating data APIs.
4	0.4	0.6	0.5	0.5	1	0.6	0.4	0.3	0.7
5	0.08	0.12	0.15	0.15	0.2	0.108	0.072	0.036	0.084

Figure 22. Analytical Hierarchy Process

18. **QFD (House of Quality)**

The Quality Function Deployment (QFD), or House of Quality, linked eight to twelve customer objectives, such as providing alternative route options and ensuring ease of use, with ten to fifteen engineering characteristics like data accuracy and UI clarity. Five competitors, like BreezoMeter and Google Map, as existing mapping apps, were analyzed to benchmark these characteristics. This tool revealed areas where the app could surpass competitors, such as real-time air quality integration. It better defined critical technical features and their alignment with user needs. Additional software development efforts focused on enhancing engineering features that were rated highly in importance yet were underperforming at present.

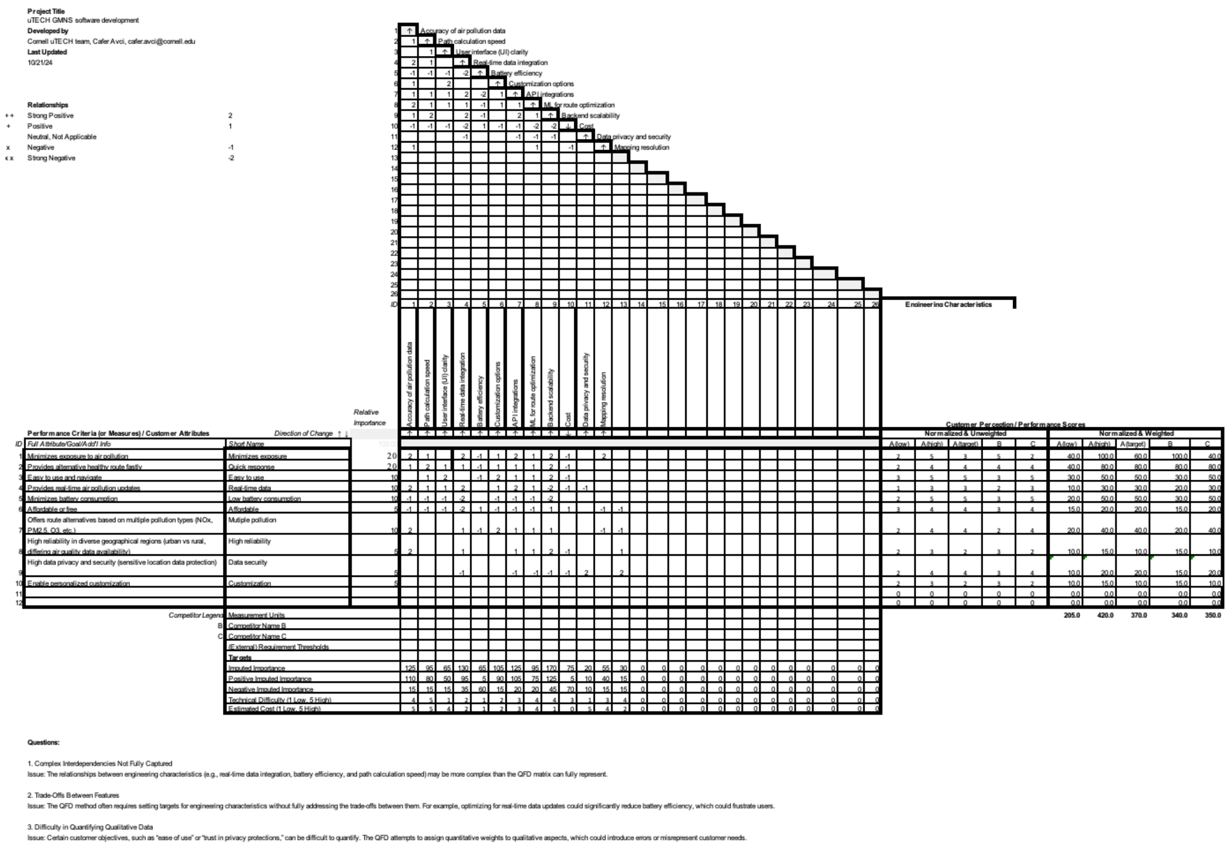


Figure 23. QFD (House of Quality)

Competitor 1: BreezoMeter, a popular app that provides real-time air quality information based on location.
Relavent Links: https://www.breezometer.com/air-quality-map/
Competitor 2: AirCare, provides air quality information and allows users to plan routes based on air quality.
Relavent Links: 1. https://getaircare.com/ 2. https://www.facebook.com/aircaretheapp/
Competitor 3: Google Maps (Waze), offering realtime driving directions based on live traffic updates.
Relavent Links: : 1. https://www.waze.com/live-map/ 2. https://en.wikipedia.org/wiki/Waze
Competitor 4: Plumelabs, provides air pollution forecasts and real-time data on air quality for various cities worldwide.
Relavent Links: https://air.plumelabs.com/en/
Competitor 5: Citymapper, a public transit app and mapping service[4][5] which displays transport options, usually with live timing, between any two locations in a supported city
Relavent Links: 1. https://citymapper.com/?lang=en 2. https://en.wikipedia.org/wiki/Citymapper

Figure 24. Competitors

19. **Sub-System Definition & Allocation**

The Subsystem Definition and allocation process distributes all system requirements across subsystems, such as the Route Calculation Subsystem and the User Notification Subsystem. By ensuring no requirements were overlooked, we learned about the interdependencies between subsystems, such as data retrieval needing real-time adaptation for dynamic route updates. The responsibilities and boundaries of these structured subsystems encourage the subsequent phases to enhance interfaces and guarantee smooth integration among the subsystems.

Air Pollution Data Retrieval Subsystem	Data Validation and Quality Control Subsystem	Route Calculation Subsystem	User Interface (UI) Subsystem	Connectivity and Offline Mode Subsystem	Real-Time Adaptation Subsystem	User Privacy and Security Subsystem	System Health Monitoring Subsystem	User Notification Subsystem
The app shall integrate real-time air pollution data (e.g., from AERMOD or other sources) to overlay pollution concentration on the map.	The system shall begin loading necessary resources.	The system shall prepare to calculate the exposure of each possible route from the starting point to the end point	The system shall detect the tap input.	The system shall provide users with analytics on their route history, including cumulative pollution exposure, etc.	The system shall request geolocation access from the user.	The system shall verify user credentials.	The system shall begin loading necessary resources.	The system shall send real-time alerts if the pollution levels on a chosen route exceed a certain threshold.
Data Source Selection	*Data Consistency Checks*	*Optimization Algorithms*	*Interactive Map Display*	*Data Caching*	The system shall display the geolocation access permission pop-ups.	*Data Anonymization*	*Error Logging and Reporting*	*Real-Time Alerts*
API Integration	The system shall store the starting point information.	The system shall calculate the exposure of each possible route	The system shall initiate app launch sequence.	The system shall synchronize user data across multiple devices.	*Environmental Data Integration*	The system shall allow users to create and manage profiles that store personal preferences such as route preferences (e.g., prioritizing low pollution, time, or distance).	*Automatic Restart Mechanism*	The system shall allow users to share their healthiest routes with friends or the public, encouraging the adoption of pollution-aware routes within communities.
The system shall incorporate air pollution forecasting for future time periods.	*Fallback Data Strategy*	*Weighted Pollution Factors*	The system shall transite to the main interface.	*Reduced Functionality Model*	The system shall receive confirmation of granted access.	*User Data Control*	*Error Logging and Reporting*	*Route Update Notifications*
Data Frequency and Prediction Control	The system shall validate if the starting point is a valid location.	The system shall provide a basic side-by-side comparison of different routes, but the comparisons included, such as time, distance or pollution levels, have not yet to be fully defined.	The system shall display the interface for inputting locations.	*Network Status Monitoring*	The system shall activate its location-based features.	The system shall allow users to input health-related information (e.g., respiratory conditions).		*Feedback Collection*
	The system shall store the ending point information.	The system shall allow users to input multiple stops along their route, calculating the healthiest path that passes through all chosen	*User Input Fields*		*Dynamic Route Adjustment*	*Secure Data Transmission*		
	The system shall validate if the ending point is a valid location.	*Adaptive Route Options*	The system shall display multiple routes with respective pollution exposure levels, allowing users to visually compare options.		The system shall provide voice-guided navigation for the healthiest routes, giving users hands-free interaction during their journey.			
	Data Filtering and Validating		The system shall include a settings page where users can choose preferenced language.		*Notification and Navigation Interface*			
			The system shall allow users to toggle basic map layers such as traffic, satellite view, or a simplified map.					
			Display and Notification Panel					

Figure 25. Sub-System Definition & Allocation

20. **ODT**

The ODT outlined operational states for at least four use cases, such as initializing the app, inputting starting and ending points, and selecting routes. It included columns for state definitions and transitions, helping clarify how the system reacts to various inputs and environmental changes. This table helped us better define system flexibility and robustness, guiding further work on handling edge cases like incomplete data or user interruptions.

Operator (User)	User Interface (UI) Subsystem	Data Validation and Quality Control Subsystem	Air Pollution Data Retrieval Subsystem	Route Calculation Subsystem	Real-Time Adaptation Subsystem	Routing	State	Timing Target
The user opens the navigation app.							Initializing	
	The system shall display the interface for inputting locations.							
The user types in a starting point.							Input, Awaiting validation	
	The system shall update to show the entered starting point.							
		The system shall validate if the starting point is a valid location.						
			The system shall request air quality data for the starting point.				Starting point set	
The user types in an ending point.							Input, Awaiting validation	
	The system shall update to show the entered ending point.							
		The system shall validate if the ending point is a valid location.						
			The system shall request air quality data for the ending point.				Ending point set	
User requests route calculation.							Calculating	
	The system shall display a series of routing notification.							
			The system shall retrieve air pollution data for areas along potential routes.					
				The system shall calculate routes based on lowest exposure to pollution.			Routes calculated	Few seconds
Info. Event ("Routes calculated")								
	The system shall update to display suggested routes.							
					The system shall evaluate traffic and pollution data in real-time, if available.		Routes displayed	
User selects a route.								
	The system shall display the selected route on the map.							
			The system shall confirm relevant pollution data for the chosen route.					
				The system shall store route details.				
					The system shall monitor the route for changes in pollution or traffic conditions.		Route actived	

Figure 26. ODT

21. State Diagram & Matrix

The State Diagram and Matrix outlined the various states of the Healthy Route App, including "Initialization," "Validating," "Data Processing," and "Route Calculating." It also specified the transitions triggered by user actions or system events. By covering at least four use cases, the diagram illustrated how the app dynamically shifts between these states. For example, it transitions from "Input, Awaiting Validation" to "Starting Point Set" when a user enters a starting point. This tool clarified the app's responsiveness to user inputs and system conditions.

Through an analysis of the state transitions, we identified areas that required optimization, such as ensuring smooth transitions when users switch between online and offline modes. This led to further improvements, including refining error-handling mechanisms for state transition failures.

State	Initializing	Input, Awaiting validation	Starting point set	Input, Awaiting validation	Ending point set	Calculating	Routes calculated	Routes displayed	Route actived
Initializing		App resources are loaded							
Input, Awaiting validation	App startup complete		Use entered starting point						
Starting point set				Starting point validated					
Input, Awaiting validation					Use entered ending point				
Ending point set						Ending point validated			
Calculating							Route calculation initiated		
Routes calculated								User selects a route	
Routes displayed									Route guidance activated, real-time monitoring begins
Route actived	Route navigation completed								

Figure 27. State Diagram & Matrix

22. **Interface Matrix**

- Basic or Adv. for 1 Subsystem covering the ODT's 4 Use Case min

The Interface Matrix captured the interactions between key subsystems, such as the User Interface (UI) Subsystem, Route Calculation Subsystem, and Air Pollution Data Retrieval Subsystem. It outlined data flow and control signals, ensuring that all subsystems communicated effectively. For example, it documented how the Route Calculation Subsystem retrieves pollution data and sends results to the UI. This tool helped better define inter-subsystem dependencies and revealed potential bottlenecks, such as data latency. As a result, additional work focused on streamlining these interfaces, such as optimizing API calls for faster data retrieval.

Cornell Cup USA presented by Intel

Sample Interface Trace Matrix Excerpt : User Interface (UI) Subsystem Sheet

User Interface (UI) Subsystem	Air Pollution Data Retrieval Subsystem	Data Validation and Quality Control Subsystem	Route Calculation Subsystem	Value	Units	Estimate?	Last Updated	Last Updated By	Interface Champ.	Est. Update Due Date	Actual Due Date	Row #	User Interface (UI) Subsystem
	Provided to			PM2.5	μg/m³		10/25/24	He	He	10/25/24	10/25/24	5	Real-time pollution data (PM2.5, NO2 levels, etc.)
			Provided to				10/26/24	He	He	10/26/24	10/26/24	6	Optimized routes with pollution data
		Provided to		PM2.5	μg/m³		10/26/24	He	He	10/26/24	10/26/24	7	Validated pollution data
							10/24/24			10/24/24	10/24/24	8	Timing Requirement
	Provided to			1	second		10/24/24	Yuwen	Meg	10/24/24	10/24/24	9	Continuous data updates
		Provided to		~300	ms		10/24/24	Yuwen	Meg	10/24/24	10/24/24	10	Fast validation for UI display
			Provided to	~1	second		10/24/24	Yuwen	Meg	10/24/24	10/24/24	11	Real-time response to user queries
							10/13/24			10/13/24	10/13/24	12	Control Signals
		Provided to					10/13/24	Meg	Yuwen	10/13/24	10/13/24	13	User-initiated data request
			Provided to				10/13/24	Meg	Yuwen	10/13/24	10/13/24	14	Initiates route calculation based on user input
	Provided to			~30	second/period		10/13/24	Meg	Yuwen	10/13/24	10/13/24	15	Periodic system-initiated data pull

Figure 28. Interface Matrix 1

Cornell Cup USA presented by Intel

Sample Interface Trace Matrix Excerpt : Air Pollution Data Retrieval Subsystem Sheet

User Interface (UI) Subsystem	Air Pollution Data Retrieval Subsystem	Data Validation and Quality Control Subsystem	Route Calculation Subsystem	Value	Units	Estimate?	Last Updated	Last Updated By	Interface Champ.	Est. Update Due Date	Actual Due Date	Row #	Air Pollution Data Retrieval Subsystem
Provided to				5000	km		10/25/24	He	Yuwen	10/25/24	10/25/24	5	User location and route preferences
			Provided to	500	ms		10/24/24	He	Yuwen	10/24/24	10/24/24	6	Initiates data pull based on calculation requirements
			Provided to	PM2.5	μg/m³		10/24/24	He	Yuwen	10/24/24	10/24/24	7	Pollution data with location-specific weights
		Provided to		500	ms		10/24/24	He	Yuwen	10/24/24	10/24/24	8	Automated upon data update in retrieval subsystem
		Provided to		PM2.5	μg/m³		10/24/24	Yingfei	He	10/24/24	10/24/24	9	Pollutant data
Provided to				~200	ms		10/14/24	Yingfei	He	10/14/24	10/14/24	10	Real-time data transfer
Provided to							10/24/24	Yingfei	He	10/24/24	10/24/24	11	User-initiated requests

Figure 29. Interface Matrix 2

Cornell Cup USA presented by Intel

Sample Interface Trace Matrix Excerpt : Data Validation and Quality Control Subsystem Sheet

User Interface (UI) Subsystem	Air Pollution Data Retrieval Subsystem	Data Validation and Quality Control Subsystem	Route Calculation Subsystem	Value	Units	Estimate?	Last Updated	Last Updated By	Interface Champ.	Est. Update Due Date	Actual Due Date	Row #	Data Validation and Quality Control Subsystem
Provided to				PM2.5	μg/m³		10/25/24	Yingfei	Meg	10/25/24	10/25/24	5	User-selected pollutant layers for validation.
Provided to				200	ms		10/24/24	Yingfei	Meg	10/24/24	10/24/24	6	Immediate validation request
Provided to							10/24/24	Yingfei	Meg	10/24/24	10/24/24	7	User-initiated input validation
	Provided to			200	ms		10/24/24	He	Meg	10/24/24	10/24/24	8	Immediate transfer to validation subsystem
	Provided to			error codes	int		10/24/24	Yuwen	Yuwen	10/24/24	10/24/24	9	Invalid data flagged
	Provided to						10/14/24	Yingfei	Yuwen	10/14/24	10/14/24	10	Automatic push
			Provided to	200	ms		10/24/24	Yingfei	Yuwen	10/24/24	10/24/24	11	Initiates validation based on user selection of pollutants
			Provided to	<300	ms		10/13/24	Yingfei	Yuwen	10/13/24	10/13/24	12	Fast transfer

Figure 30. Interface Matrix 3

Cornell Cup USA presented by Intel

Sample Interface Trace Matrix Excerpt : Route Calculation Subsystem Sheet

User Interface (UI) Subsystem	Air Pollution Data Retrieval Subsystem	Data Validation and Quality Control Subsystem	Route Calculation Subsystem	Value	Units	Estimate?	Last Updated	Last Updated By	Interface Champ.	Est. Update Due Date	Actual Due Date	Row #	Route Calculation Subsystem
Provided to				String	Lat/Long		10/25/24	Yuwen	Meg	10/25/24	10/25/24	5	Start/end points, route type, and avoidance areas.
							10/26/24	Yuwen	Meg	10/26/24	10/26/24	6	Timing Requiremen
Provided to				≤ 1	second		10/26/24	Yuwen	He	10/26/24	10/26/24	7	Near-instant transfer
	Provided to			200	ms		10/24/24	Yuwen	He	10/24/24	10/24/24	8	Real-time data retrieval
		Provided to		<200	ms		10/24/24	Yingfei	He	10/24/24	10/24/24	9	Quick data transfer
		Provided to					10/24/24	Yingfei	He	10/24/24	10/24/24	10	Incomplete data flagged, user notified
		Provided to					10/24/24	Yingfei	He	10/24/24	10/24/24	11	Quality-checked data for pollution-aware routing
	Provided to			200	ms		10/13/24	Yingfei	He	10/13/24	10/13/24	12	Pollution data with location-based weights
	Provided to			error codes	int		10/13/24	Yingfei	Meg	10/13/24	10/13/24	13	Missing data prompts re-request from data sourc

Figure 31. Interface Matrix 4

23. **Sequence Diagram (SysML)**

The SysML Sequence Diagram detailed interactions between at least three participants, including two subsystems and a user. For example, it showed how the User sends a route request, the UI Subsystem processes the request, and the Route Calculation Subsystem fetches and calculates pollution exposure. This tool illustrated the timing and order of messages exchanged between components. It better defined the temporal dependencies in the system, revealing areas where delays could occur, such as during data fetching. This led to further work on optimizing communication protocols and reducing latency in subsystem interactions.

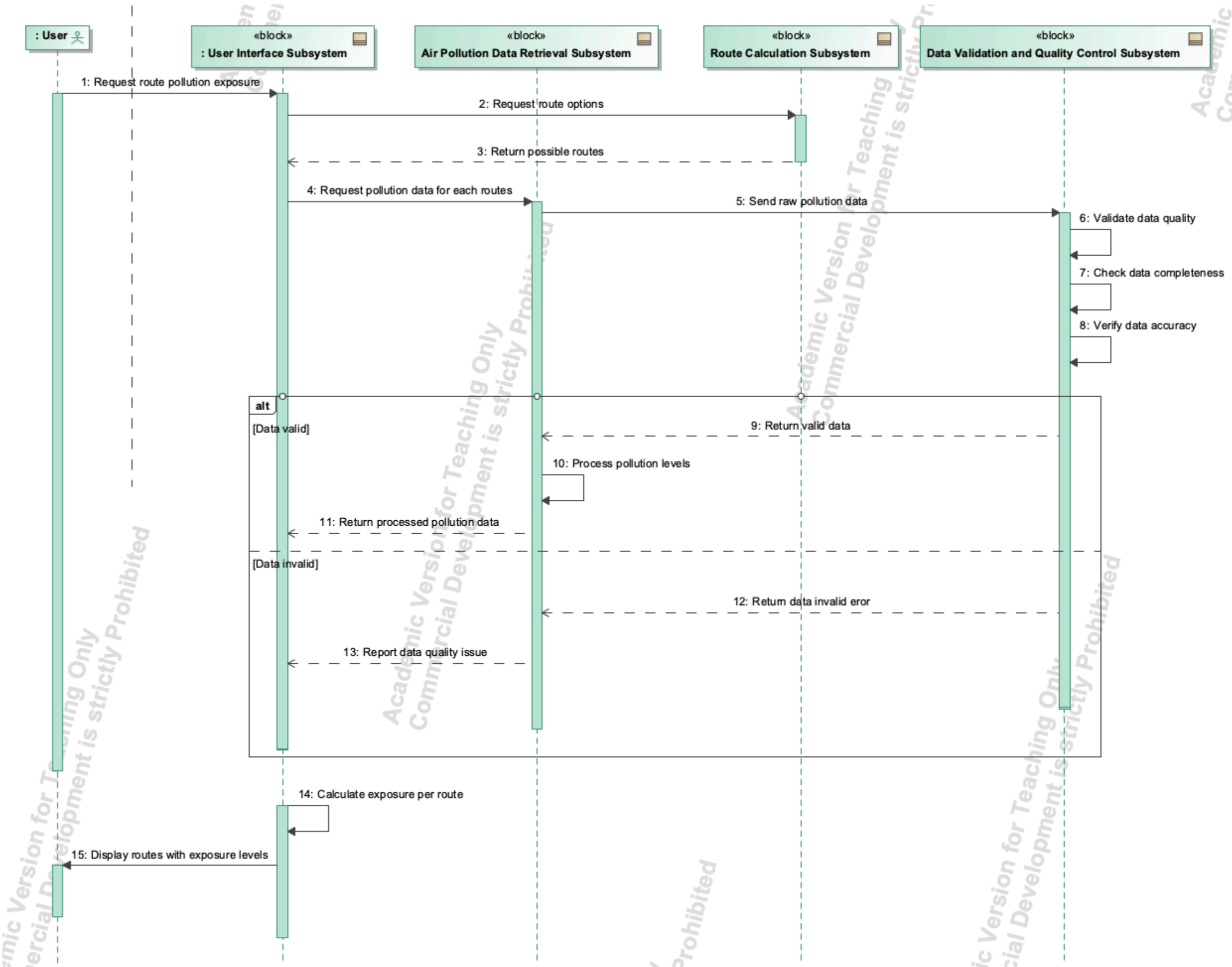


Figure 32. Sequence Diagram (SysML)

24. Behavioral Test Plan

The Behavioral Test Plan outlined three test procedures derived from originating requirements, such as validating the accuracy of pollution exposure calculations, testing the app's response to invalid user inputs, and verifying geolocation permissions. By systematically testing these behaviors, we learned about the system's reliability and responsiveness under various conditions. This tool helped better define the success criteria for core functionalities and prompted further design iterations, such as improving error messages for invalid inputs and refining geolocation access workflows.

Test #	Test Method	Test Facilities	Entry Condition	Exit Condition
TP.1	Test Procedure: Verify that single tap input is detected on all UI elements, such as buttons and list items.	Real device lab, UI testing tool.	User taps a UI element.	Tap is registered, and the corresponding action is executed.
TP.2	Test Procedure: Test that all required assets are loaded before the main interface is accessible.	Performance monitoring software.	App starts loading assets.	All assets are successfully loaded.
TP.3	Test Procedure:Verify correct credentials allow access.	Authentication backend server.	User inputs valid credentials.	User gains access to the main interface.

Figure 33. Behavioral Test Plan

25. **Test Methodologies for Non-Behavioral Requirements**

The Test Methodologies for Non-Behavioral Requirements outlined procedures for verifying non-functional aspects, such as system integration, historical data analysis, and mapping scalability. For example, load testing was used to measure how the app handles simultaneous user requests, while penetration testing assessed data protection measures. These methodologies helped us identify potential vulnerabilities, such as response delays during high traffic. They better defined the technical limits of the system, leading to further design work on optimizing server-side processes and enhancing encryption protocols.

Non-behaviral	Req. #	Requirement	Abstract Name	Test #	Test Method	Verification Method (A,I,D,T)	Test Facilities	Entry Condition	Exit Condition
	OR.30	The system shall allow users to toggle basic map layers such as traffic, satellite view, or a simplified map.	Toggle map layer	TP.4	Test switching between different map layers	Demonstration (D)	Layer toggle controls	User changes map layers.	Layer displayed changes correctly.
	OR.17	The app shall integrate real-time air pollution data.	Integration	TP.5	Verify data updates in real-time from selected sources.	Test (T).	Data feed simulation.	Real-time data feed available.	Data displayed correctly.
	OR.24	The system shall provide analytics on route history, including cumulative pollution exposure.	History analytics	TP.6	Test generation of cumulative exposure reports	Test (T)	Data analysis software	Historical route data is available.	Report is generated showing cumulative pollution exposure.

Figure 34. Test Methodologies for Non-Behavioral Requirements

26. **Verification Cross Reference Matrix (VCRM)**

The VCRM mapped test cases to requirements, ensuring comprehensive coverage of both behavioral and non-behavioral requirements. For instance, tests validating route optimization accuracy were cross-referenced with originating requirements for pollution exposure calculation.

This tool helped us track which requirements were fully validated and where gaps remained. By better defining the relationship between requirements and tests, it guided the following efforts to refine unverified requirements, such as implementing additional test cases for edge scenarios.

test (row) is used to verify originating requirement (column)	OR.1	OR.3	OR.4	OR.17	OR.24	OR.30
TP.1	x					
TP.2		x				
TP.3			x			
TP.4						x
TP.5				x		
TP.6					x	

Figure 35. Verification Cross Reference Matrix (VCRM)

27. **Severity Rating System & Likelihood Rating System**

The Severity and Likelihood Rating Systems established criteria for assessing the impact and probability of potential risks, such as data breaches or system crashes. Severity ratings categorized risks based on user impact, while likelihood ratings considered factors like historical data and system vulnerabilities. This tool helped prioritize risks that needed immediate mitigation, such as ensuring data security during server maintenance. It better defined the criticality of various risks and prompted further work on implementing fallback systems and redundancies to address high-priority risks.

Severity	
5	Cost exceeds 20% of the total budget, requires more than 4 weeks to address, and causes a major reduction in performance or poses a significant risk to user safety.
4	Cost is 10-20% of the total budget, requires 2-4 weeks to address, performance is adversely affected by 15-20%, and poses a potential but not immediate risk to user safety.
3	Cost is 5-10% of the total budget, requires 1-2 weeks to address, performance is adversely affected by 10-15%, and does not directly pose a risk to user safety.
2	Cost is 1-5% of the total budget, requires 2-5 days to address, performance is minimally affected (5-10%), and poses no harm to user safety.
1	Cost is less than 1% of the total budget, requires less than 2 days to address, performance is affected by less than 5%, and poses no risk to user safety.
Likelihood	
5	Occurs more than once a day or in greater than 20% of operations during normal use.
4	Occurs several times a week or in 10-20% of operations.
3	Occurs no more than once a week, affecting 5-10% of operations.
2	Occurs about once a month or in 1-5% of operations.
1	Occurs less than once every three months, affecting less than 1% of operations.

Figure 36. Severity Rating System & Likelihood Rating System

28. **Risk Priority Number Table for your Criticality Rating System, and a Stoplight Graph**

The RPN Table combined severity and likelihood ratings to calculate a priority score for each risk, with a stoplight graph visually categorizing risks as high (red), medium (yellow), or low (green). For example, a risk of data inconsistency during offline mode was assigned a medium priority, while user data breaches were marked as high. This tool clarified which risks required immediate attention and which could be deferred. It led to further work on risk mitigation strategies, such as introducing data validation protocols and enhancing backup mechanisms.

Risk Priority Number (RPN) Definition Table

Likelihood						
5	5	10	15	20	25	
4	4	8	12	16	20	
3	3	6	9	12	15	
2	2	4	6	8	10	
1	1	2	3	4	5	
	1	2	3	4	5	**Severity**

Risk Criticality Ranges

17-25	High Risk
11-16	Medium High Risk
7-10	Medium Risk
4-6	Medium Low Risk
1-3	Low Risk

Figure 37. Risk Priority Number Table for your Criticality Rating System, and a Stoplight Graph

29. **Failure Mode and Effect Analysis (FMEA)**

The FMEA identified six potential failure modes, such as incorrect pollution data integration or navigation errors, and analyzed their effects on user experience. For instance, failure to fetch real-time pollution data was categorized as high-impact due to its direct effect on route accuracy. By quantifying the severity, occurrence, and detection of each failure mode, this tool highlighted vulnerabilities requiring immediate design changes. It better defined the need for robust error detection systems, prompting work on implementing fallback features, such as defaulting to cached data during outages.

Failure Mode #	Subsystem	Failure Mode	Failure Effects	Possible Cause	Failure Effects Severity	Occurrence Likelihood	Risk Priority Number (RPN) (=severity * occurrence)	Risk Criticality (Corrective Action Priority)		Corrective Action	Failure Effects Severity	Occurrence Likelihood	Risk Priority Number (RPN)	Criticality (Corrective Action Priority)	Corrective Action Effort	Corrective Action	Failure Effects Severity	Occurrence Likelihood	Risk Priority Number (RPN)	Criticality (Corrective Action Priority)	Corrective Action Effort	Troubleshooting (optional)
F.1	Real-Time Adaptation	Failure in Real-Time Adaptation to Changing Conditions	Users may continue along unhealthy routes, negating the purpose of the app.	data update delays	4	5	20	HIGH		Develop caching and pre-fetching mechanisms to minimize data input delays and create data validation rules to ensure accuracy.	4	2	8	MEDIUM	150 hrs.							Check for data latency using network logs and timestamps to identify bottlenecks.
				processing lags or collapse	4	2	8	MEDIUM		Optimize algorithm efficiency through code refactoring and employ parallel processing where feasible.	4	1	4	MEDIUM LOW	120 hrs.							
				network connectivity issues	2	5	10	MEDIUM		Implement offline mode capabilities with automatic data synchronization when connectivity is restored.	4	1	4	MEDIUM LOW	120 hrs.							
F.2	Data Validation and Quality Control	Inaccurate User Location Detection	The app fails to accurately determine the user's current location, leading to calculating routes from incorrect starting points, reducing accuracy and user trust.	Incorrect GPS signal interference	4	4	16	MEDIUM HIGH		Utilize alternative positioning methods like Wi-Fi and cellular triangulation, and implement signal fallback strategies.	4	2	8	MEDIUM	120 hrs.							Test GPS signal strength and accuracy across various environments (indoors, urban areas, rural areas). Compare location data with known reference points to assess errors and inconsistencies.Verify the use of fallback mechanisms (e.g., Wi-Fi, cellular triangulation) when GPS is unavailable.
F.3	Route Calculation	Incorrect Route Calculation	Incorrect or suboptimal routes due to errors in the route calculation logic. Users may take routes with higher air pollution exposure than expected.	enerates incorrect or suboptimal routes due to errors in the route calculation logic.	4	4	16	MEDIUM HIGH		Conduct a comprehensive review and testing of algorithms with various datasets to identify and correct logic errors.	2	1	2	LOW	120 hrs.	Introduce A/B testing for algorithm iterations to identify the most accurate approach.	4	1	4	MEDIUM LOW	60 hrs.	Check data inputs (e.g., map data, pollution data) for accuracy and completeness. Conduct step-by-step logging of the route calculation process to identify logical errors. Test algorithms with different edge cases (e.g., complex paths, missing data) to ensure robust behavior.
				incorrect data inputs	2	4	8	MEDIUM		Establish strict data quality controls, including automated checks and verification processes.	2	1	2	LOW	30 Hrs.							
				computational errors in the calculation module	2	5	10	MEDIUM		Introduce error detection and correction mechanisms during calculations.	2	1	2	LOW	30 Hrs.							
F.4	User Notification	Inconsistent User Notifications	Users miss critical updates, impacting route decisions and health outcomes.	incorrect logic for triggering alerts	4	5	20	HIGH		Redesign the triggering logic for consistency and introduce logging to identify and correct discrepancies.	4	1	4	MEDIUM LOW	90 hrs.	Use mock user environments to simulate different conditions affecting notifications.	4	1	4	MEDIUM LOW	90 hrs.	Check for overlapping conditions in the notification logic that may cause conflicts or skipped triggers. Use logging to verify the delivery of notifications through different channels (e.g., push notifications, email).
				network connectivity issues	2	5	10	MEDIUM		Implement message queuing to ensure notifications are delivered once connectivity is restored.	4	1	4	MEDIUM LOW	90 hrs.	Develop hybrid communication protocols that switch between data sources (e.g., cellular, Wi-Fi) as needed.	5	2	10	MEDIUM	90 hrs.	
F.5	Data Validation and Quality Control	Ineffective Data Integration from Multiple Sources	Users may receive inaccurate route information based on conflicting or incomplete data.	data conflicts	4	4	16	MEDIUM HIGH		Introduce data normalization and transformation layers to align incoming data to a standardized format. Implement data completeness checks and alert systems for missing data, with protocols for fetching backups.	2	5	10	MEDIUM	120 hrs.							Check logs for communication errors, data parsing failures, or transmission delays. Verify the presence and consistency of required data fields in data feeds. Isolate and validate data from each source individually before integrating with the broader system.
				schema mismatches	2	4	8	MEDIUM		Develop retry logic and fallback methods for data transmission failures.	2	5	10	MEDIUM	120 hrs.							
F.6	Data Validation and Quality Control	Crash on Specific Devices or OS Versions	Users are unable to use the app on those devices, resulting in negative reviews and lost adoption.	compatibility issues	4	4	16	MEDIUM HIGH		Perform device-specific testing and optimize the codebase for compatibility using cross-platform development tools.	4	1	4	MEDIUM LOW	90 hrs.							Check for any unhandled exceptions in the code that could trigger application crashes. Confirm that all third-party libraries or dependencies are up-to-date and compatible with target platforms.
				memory constraints	4	2	8	MEDIUM		Conduct memory usage profiling and optimize resource allocation to reduce leaks and excessive consumption.	4	1	4	MEDIUM LOW	90 hrs.							

Likelihood	Current # of Risks Shown Stoplight Graph				
5	0	3	0	2	0
4	0	2	0	4	0
3	0	0	0	0	0
2	0	0	0	2	0
1	0	0	0	0	0
	1	2	3	4	5
					Severity

Likelihood	Current # of Risks Shown Stoplight Graph				
5	0	2	0	0	0
4	0	0	0	0	0
3	0	0	0	0	0
2	0	0	0	2	0
1	0	3	0	6	0
	1	2	3	4	5
					Severity

Likelihood	Current # of Risks Shown Stoplight Graph				
5	0	0	0	0	0
4	0	0	0	0	0
3	0	0	0	0	0
2	0	0	0	0	1
1	0	0	0	2	0
	1	2	3	4	5
					Severity

Figure 38. Failure Mode and Effect Analysis (FMEA)

30. **Event Tree or Fault Tree**

The Event Tree diagram illustrated potential pathways leading to critical failures, such as app crashes during route recalculations. At least three to five levels of depth were included, breaking down root causes like API timeouts or memory overflows. Equations quantified the probability of each failure, helping us assess overall system reliability. This tool clearly showed the high-risk components and dependencies, guiding further establishment of fault isolation techniques and preventive measures to reduce the likelihood of cascading failures.

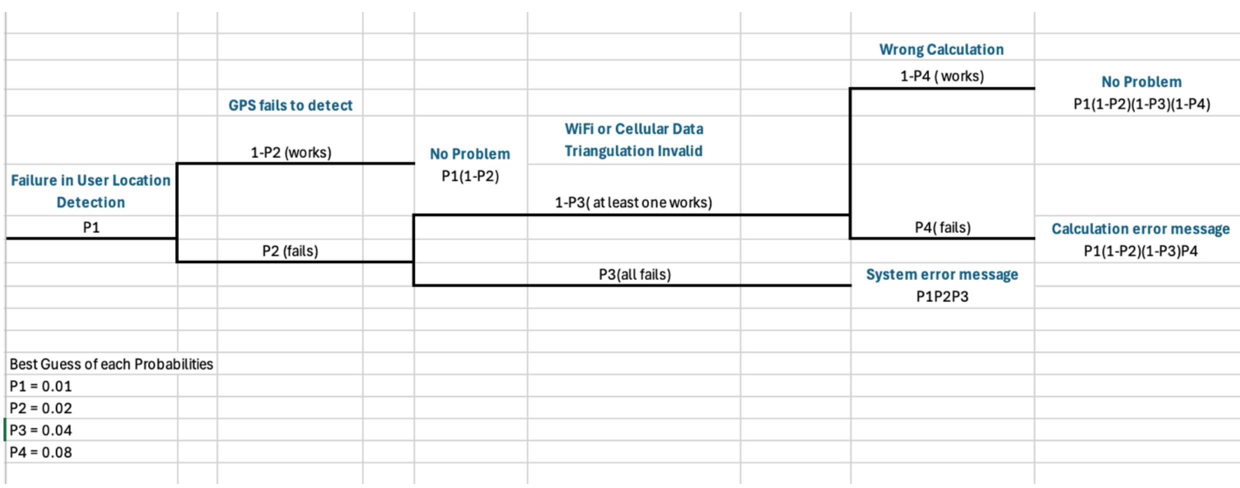


Figure 39. Event Tree

31. **SysML Parametric Dia.**

The SysML Parametric Diagram modeled three equations to calculate outputs like exposure level, health risk level, and the total route score. For example, one equation combined pollution concentration and route distance to calculate exposure levels. Another equation determined health risks for different users. This tool clarified how system variables interact and contributed to output calculations. By identifying dependencies and inconsistencies, it can better define algorithm requirements and guide additional efforts to optimize real-time performance calculations.

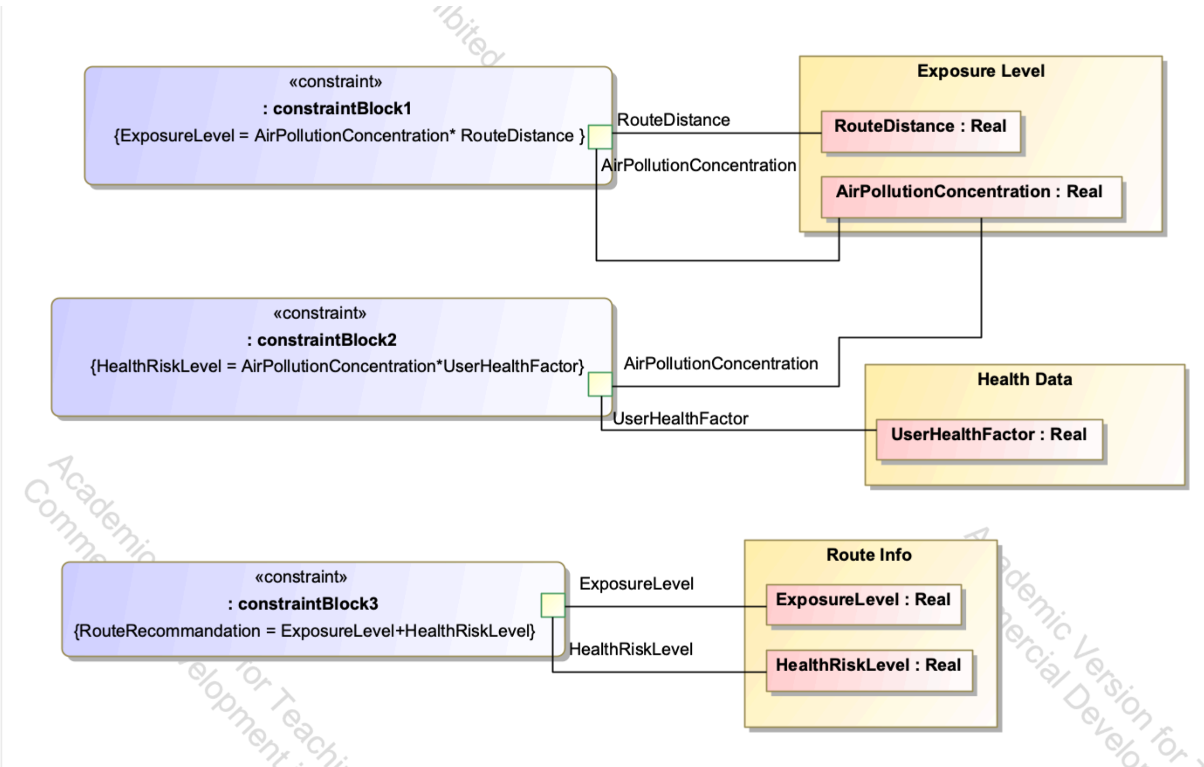


Figure 40. SysML Parametric Dia.

32. **Timeline Update or Extension (2-4 weeks ≈ 1 Sprint)**

The Timeline Update proposed the steps for our Healthy Route App, focusing on critical dependencies and deliverables informed by the systems engineering tools. Key tasks included refining the air pollution forecasting module, enhancing user notifications for high-pollution areas, and implementing more robust offline features. Dependencies were identified, such as completing subsystem integration before launching new features. This tool illustrated project priorities and guided resource allocation for future sprints, ensuring continued progress toward delivering a high-quality app.

Start Date	Completed	Due Date	Team Meml	Time Actua	Time Req (	Effort (1-7)	Dependent	ID	Sub-ID#s			Milestone	Task	SubTask 1	SubTask 2	SubTask 3	Deliverables
							*	A				Project Initiation and Planning					
							*		1				Define project scope and objectives				A detailed description of the project's objectives, scope, stakeholders, and high-level requirements.
									2				Identify key stakeholders				A list of all identified stakeholders, their roles, and their interests in the project.
									3				Create a detailed project plan and timeline				A timeline showing all major tasks, milestones, and dependencies.
									4				Establish development team and roles				A document with team members, their roles, and time commitments.
								B				Concept Refinement & Requirements Validation					
							*A.1, A.4		1				Conduct industry research				A comprehensive research report on current trends in mobile health apps, with the technology advancements in air quality monitoring
									2				Landscape Analysis				A comprehensive document summarizing findings on relavant competitors with the potential gaps
							A.1		3				Identify target users				
										1				Determine key use cases			List of possible use cases of the application when using in different scenarios
										2				Determine key user requirements			List of originating user requirements for the application
							A.2		4				Come up with marketing strategy ideas				A proposal for marketing plan to raise awareness about the importance of pollution exposure and its health impacts
								C				Data Explore & Acquisition					
							*B.2		1				Research traffic-related air pollution models (e.g., AERMOD)				Comprehensive documents summarizing findings on traffic-related air pollution models, with a focus on AERMOD.
									2				Acquire or set up access to real-time air quality data sources				A list of identified real-time air quality data sources, including their reliability, update frequency, and coverage areas.
									3				Study traffic assignment problem algorithms				A report on various traffic assignment problem algorithms, their pros and cons, and recommendations for the project.
							B.3.1		4				Investigate APIs for mapping and routing services				Collected documentation for relevant mapping and routing services APIs.
								D				System Design and Prototyping					
							*A,B,C		1				Initial System Design				FFBDs showing system processes and interactions
							*D.1		2				Interface Development				Completed Interface Matrix and documented interface challenges
							*D.2		3				Prototype Development				A low-fidelity UI prototype ready for testing
								E				Integration and Testing					
							*D.1, D.2		1				Integrate UI, route calculation, and data retrieval subsystems				Integrated prototype demonstrating key functionality
							*E.1		2				Develop behavioral test cases				Behavioral test cases with documented results
								F				Advanced Features Development					
							*D.1, D.2		1				Real-Time Adaptation				Working real-time adaptation features
							*F.1, F.2		2				User Privacy & Security				Completed privacy and security subsystems
							*F.1, F.2		3				Voice-Guided Navigation				App with fully functional voice-guided navigation feature
								G				Validation & Final Testing					
							*F.1, F.2, F.3		1				System Validation				A finalized testing report showing system readiness
							*G.1		2				Stakeholder Review				Stakeholder-approved app ready for deployment
TestOftenTestEarly							Next Steps After This Timeline:										

Figure 41. Timeline Update