SOLAR ROUTE PLANNER

Tools Report

Janhavi Gaikwad (jg2569)
Systems Engineering, M-Eng'25

Special Thanks

I sincerely thank Professor Prof. David Schneider for his guidance, Prof. Wesley Hewett for SysML support, Yanbei Liu for insights that improved our assignments, and Mr. Yishuo Jiang, our project head, for leadership throughout the project.

Table of Contents

Figure Index

Table Index

Project Background

Drivers in urban environments are often exposed to varying levels of solar radiation during their commutes, which can lead to health conditions, discomfort, increased fatigue, and reduced overall driving efficiency. Conventional navigation systems prioritize the shortest or fastest routes but do not account for environmental factors such as sunlight exposure. The goal of this project is to develop a solar aware route planner for vehicles that minimizes sun exposure while maintaining efficient travel times. The system integrates real time solar and weather data from external data centers, calculate shadow parameters, and applies optimization techniques to generate personalized driving routes. Using systems engineering principles, the project defines functional and structural requirements, identifies potential failure modes, and ensures that the final solution is reliable, efficient, and aligned with driver comfort and safety needs.

The following report documents the design and development of solar route planner using systems engineering methodologies. It details the all system engineering tools used across project life cycle from understanding customer needs through tools like customer affinity process, functional and structural modeling using activity diagrams, FFBD, IDEF0, QFD. The report also covers risk analysis and testing of product using FMEA, VCRM etc. Overall, it documents the use of different systems engineering tools for the design and development of user centric product.

Tools report

1. Customer Value Proposition

The Customer Value Proposition is a systems engineering tool that defines how a product meets customer needs, and how those needs will be addressed. It links critical customer requirements to essential design features, ensuring that the system remains user focused throughout the design process. The CVP translates high priority customer needs, derived from competitor analysis and project goals in QFD, into actionable value propositions for the product.

Using the CVP, we were able to identify the most critical requirements for the solar route planner and understand how these requirements could be met through the systems features. It provided a holistic view of the system deliverables to the user. Insights from the CVP laid the foundation for subsequent tools, such as the annotated concept sketch and the tests, by clearly highlighting which features and functions were essential to deliver maximum value to the customer hence deciding which requirements from originating requirements to focus upon.

2. Annotated Concept Sketch, Storyboarding

The Annotated Concept Sketch is a systems engineering tool that provides a high-level visual representation of a systems structure, key functions, and critical components in a single diagram. It offers a holistic view of how structural and functional elements interact with each other and with the external environment. For the solar route planner project, the annotated concept sketch combined both structural and functional components, illustrating how data centers collect weather and solar data, how the model processes this information to calculate efficient routes, and how the system guides the user along the chosen path.

For this project, functional annotated concept rough sketch is created after defining initial originating requirements and later refined into the structural and functional combined sketch after CVP definition.

Figure 1: Structural and Functional Annotated sketch

Using this tool, we were able to understand the key structural components of the system, understand their interactions, and determine which data is transmitted between subsystems. It helped us better define the system from users' perspective, which helps to inform about the system to wider non-technical audience. The insights from the annotated concept sketch also guided further design work, including refining data requirements in the context diagram and generating additional use cases for the system, ensuring that all major interactions and dependencies were clearly captured for subsequent development.

3. Customer Affinity Process Results

The Customer Affinity Process is a systems engineering tool used at the beginning of the project to understand customer needs, expectations, and pain points. It involves gathering and analyzing customer input to identify key areas for improvement and desired features in the system. For the solar route planner project, this process helped to capture user preferences and expectations on early stages, guiding the initial definition of system requirements.

Using this tool, we were able to identify the main pain points of users, as well as additional features they expected in the system. For example, aesthetic user interface is highly recommended feature as per the results

Group Lvl 1 Titles (Group)	Sum # of Comments in Each Lvl 1 Group	Group Lvl 2 Titles (Group Summary)	Sum # of Comments in Each Lvl 2 Group	Group Lvl 3 Titles (Area)	Sum # of Comments in Each Lvl 3 Group
Route simplicity	3	Design	23	Functionality	43
Compact Route	3				
Clear Directions	3				
Asthetic and feel	14				
Reliabilioty	4	Accuracy	9		
Non-fragile data	5				
Safe navigation	10	Safety	12		
Safety alerts	2				
Single user	3	User Interaction	11	Feature	32
Multi user	8				
Compatibility	4	User Convinience	10		
Lightweight app	2				
Affordable	4				
Ease of use	5	Functionality	11		
Flexible options	6				
User skill support	8	Learning	11	Impact	35
Knowledge access	3				
Creative route option	6	Exploration	13		
Imaginative paths	3				
Standard path	4				
User engagemnet	7	Engagement	10		
Focused recommendation	2				
History log	1				
Total	**110**		**110**		**110**

Table 1: Customer affinity process result

This allowed us to better define the initial functional requirements and potential enhancements to make the system more appealing and user focused. Insights from the Customer Affinity Process also helped in forming the initial originating requirements and use cases for the system, providing a strong foundation for subsequent design and development work for user-oriented system.

4. Context Diagram

A context diagram is used to represent the system as a single entity and shows how it interacts with external system, user and data sources. It focuses only on the system boundary and information exchanged across it. It doesn't talk about the internal details of the system. In this project the context diagram established holistic view of how route planner model interacts with the surrounding environment.

In our project, the context diagram was developed after obtaining results from the customer affinity process and after identifying the initial set of requirements. At this stage it is used to refine the understanding of data transfer of systems with external systems like data server. By creating this diagram, it became clear that which component was under our control and which were outside the system control. This helps us better define system scope, additional use cases to have a robust system. For example, although system depends upon the sun position data from the datacenter, the data center itself is not part of system as shown in figure below.

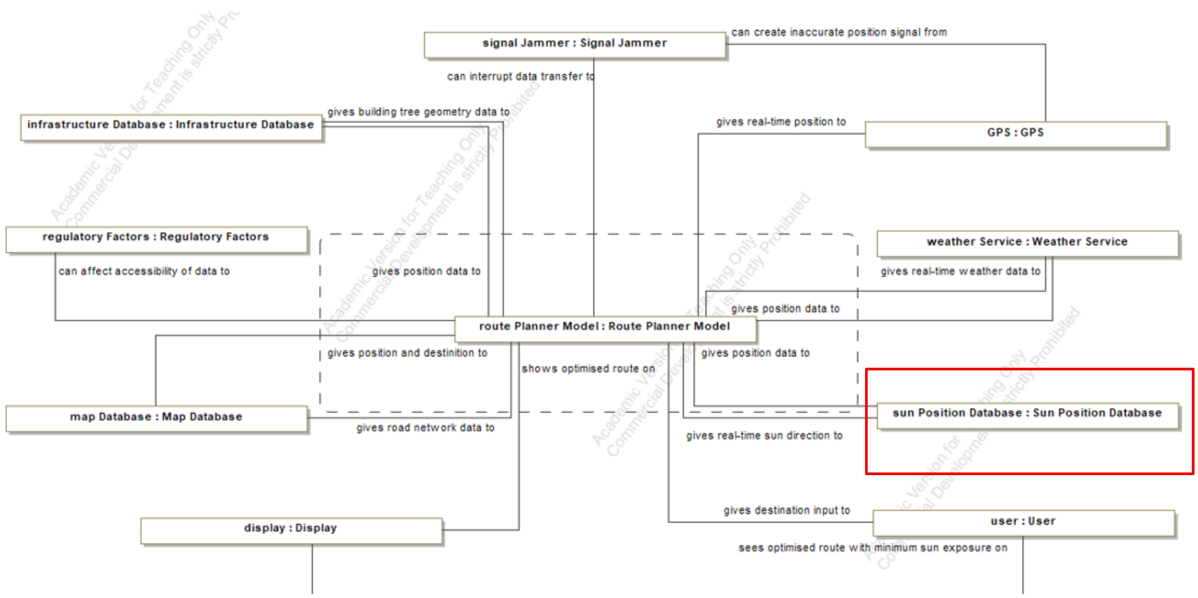


Figure 2 : Context Diagram

This leads to additional requirements focused on requirements related to managing failure scenarios like data center outage or data unavailability.

5. Use Cases & User Story

Use cases are the list of ways users will interact with the system. It is written during behavioral modelling of the system. In our workflow it is defined after context diagram where we enumerate specific scenarios the system must support user interaction and rank them by priority as Low, medium, high. While user stories capture how a given use case is implemented from stakeholders' perspective. User stories are defined during the implementation stage of the system documenting the stakeholder conversation and design actions taken to fulfill the requirement.

User cases let us understand the different user scenarios under which the solar route planner must operate. Additionally, it tells which use cases are more important. While the user stories make us understand how those use cases are implemented in our system by documenting stakeholder expectations and outcome of the implementation as shown in confirmation section of user story below

User story 1

❖ **Card**

- The system shall provide a planned route in the absence of real-time data, ensuring uninterrupted travel and reducing sun-related health risks. The provided route shall achieve at least 98% accuracy compared to a route planned using real-time data.Offline data for route planning should be stored automatically.

❖ **Conversation**

- Available data storage capacity in model for offline data is 2TB.
- They would like frequency of updation for infrastructure and route data to be once in a month.
- They would like to store weather and sunposition forcast data of 7 days.
- Maintaining accuracy with respect to relatime-data model is top priority, storage space for data is secondary priority

❖ **Confirmation**

- Automatic oflline data updating system is designed which can store 7 days of sun position and weather forcast data. System updates infrasturcture and route data once a month.
- Model can successfuly plan the route with more than 98% of accuracy with respect to the real-time data model.

Figure 3: User Story

User stories let us better define the originating requirements to make those requirements more user focused. It also helped us define the final implementation result for each use case in the later stages. The next natural step after defining use cases was to explore interactions between different use cases using use case diagrams and to convert key use case cases into use behavioral diagrams to refine operation level requirements.

6. SysML Use Case Diagram

Use case diagram was created after defining the use case list and assigning priorities to each use case. It provides visual representation of how different use cases interact with each other and with user from functional point of view. Unlike use cases where every use case is listed in isolation, the diagram help us elaborate how one case can trigger each other. Through this diagram, we learned how major operational scenarios are connected and how system behaves in different scenarios from user perspective. The diagram better helps to define the expected system behavior from the view of functional priorities. For example, the diagram shows system attempts to use real-time data by default(includes) and switches to offline data(extended) when real time data is unavailable, clearly representing both interaction and priority.

Figure 4 : Use Case Diagram- Use case interaction

This exercise led to further design work by giving additional functional requirements. Also, it helped identify user-critical requirements that must be satisfied for reliable systems operation.

7. Use Case Behavioral Diagrams

A use case behavioral diagram is used to elaborate high level use case by breaking it down into a functional flow across different stakeholders. While use cases capture what the system should do at high level, the UCBD translates that into sequence of action items and decision-making points.

The UCBD was created after defining the use case to convert high level requirements into precise and quantifiable requirements necessary for systems operation. This tool helped us understand how system should act internally and how it should interact with external subsystems to achieve given objective. It helped better define originating requirements by reducing ambiguity. For example, the use case “user wants to add an intermediate stop during the journey” was initially broad requirement. Through UCBD, this was broken down into specific requirements such as the number of stops user can add, how stops are selected on map and how they are integrated into route planning as shown in figure below.

User wants to add an intermediate stop during journey		
Initial Conditions		
1. The sytem is in navigation mode		
Operator (Driver)	**System (Route planner)**	**User display**
The driver inputs the intermediate stop		
	The system shall be able to accept MAX_STOPS stops	
	The system shall be able to locate stops on map	
		The user display will show marked stops on map window
The driver will provide acceptance		
	The system shall connect with remote data centre to access realtime infrastructure data	
	The system shall connect with remote data centre to access realtime route data	
	The system shall plan route from the data with minimum Online_Sun_Exposure_Limit	

Table 2: UCBD

These other requirements are generally ignored or handled in a less collaborative way which can lead to integration problems at later stages. Having this precise requirement helped reduce uncertainty. This work further led to the creation of the Operational description tree (ODT), where the interactions and interfaces identified for given UCBD.

8. SysML Activity Diagram

A SysML activity diagram is essentially the same as use case behavioral diagram but expressed in SysML environment. It represents functional flow of the system, showing sequence of actions, decisions and interactions between the system and critical external actor like user display and data server as shown in figure

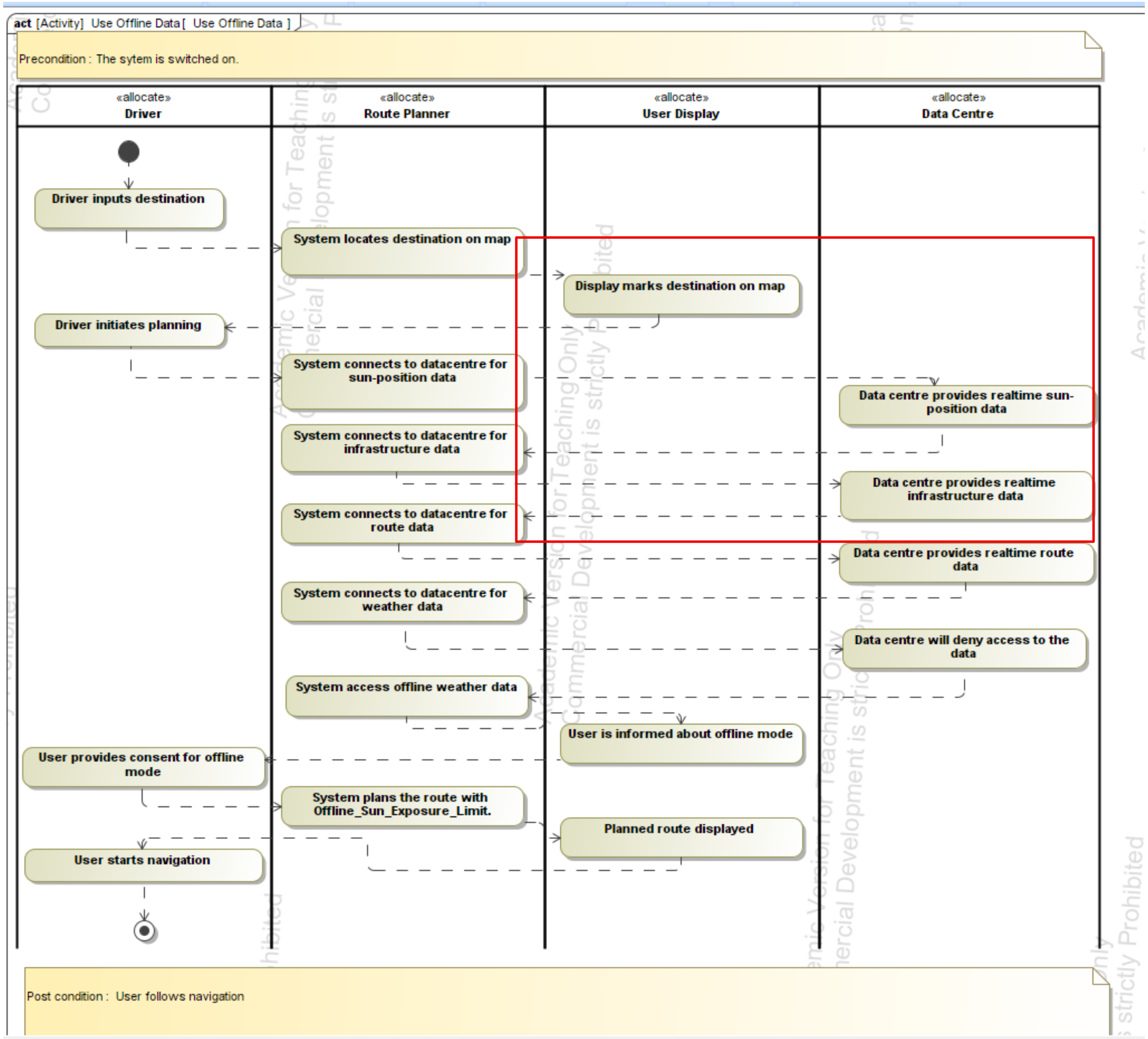


Figure 5: Activity Diagram

The activity diagram was used after defining use case diagram in SysML environment to further quantify high level requirements and define structured visual form. The overall learning and the definition is like UCBD diagram. However, the advantage of defining activity diagrams in SysML environment is that it can be integrated into the activity diagram model, allowing access with a single click in a more systematic and effortless manner. However, a limitation is that the activity diagram must be paired with the corresponding requirements diagram to have well-phrased system requirements in same diagram, which adds extra step.

9. Originating & Derived Requirement Tables

Originating and derived requirement tables are structured tools used to capture, organize and track all requirements of the system. The originating requirements represent needs gathered from stakeholders, customers or regulatory sources. Additionally, many detailed OR are captured during systems engineering activities. For some requirements derived requirements are defined for given originating requirement. Each requirement is assigned to unique ID to ensure traceability across project lifecycle. These tables provide a single consolidated view of all system requirements.

These tables are used and defined throughout the project lifecycle to understand customer needs, guide conceptual design and support functional and behavioral modelling until the development stage. It helps us understand which requirements of system were intended to fulfill, even if some were later deprioritized. They helped me better define system by consolidating all functional attributes of system at one place and tracing them to different systems of engineering operations. Having this consolidated list also clarified scope of the system and guided further design work including detailed analysis, risk assessment and verification planning.

10. SysML Requirements Dia/Table

The SysML requirements table is essentially the originating requirements expressed in the SysML environment, capturing all stakeholders and system needs in structured traceable format. While the SysML requirements diagram is a graphical presentation of the same requirements. It is generally used complimentary to SysML activity diagram to trace requirements obtained from activity diagram to originating requirement as shown in figure

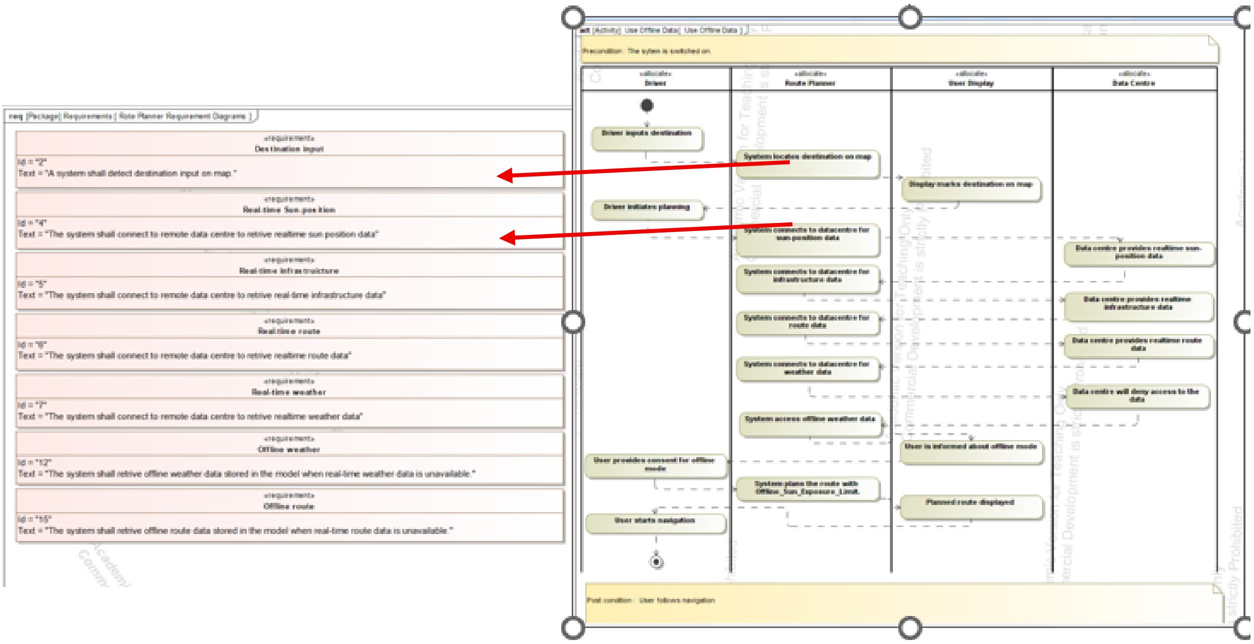


Figure 6: Associated Requirements Diagram for Activity Diagram

This tool was used to map the requirements derived from activity diagram, helping me trace that requirement to originating requirements. Although overall learning definition and future work was same as originating requirement. Defining requirements in the SysML environment allows easy handling of those requirements to be converted into graphical representation for other purposes while seamlessly maintaining traceability.

11. Concept Fragment Generation

Concept fragment generation is a systems engineering tool used to explore multiple ways of fulfilling given requirement. Once high-level use cases are identified and converted into functional originating requirements, concept fragments serve as a brainstorming tool to systematically consider different potential solutions.

Using concept fragments, we could understand all possible ways to satisfy a specific requirement and why some solutions were not considered. For example, for concept fragment table I could see different ways to update cloud coverage data in system, like updating at fixed interval or triggering updates by certain event. While we can also see some approaches were removed due to implementation complexity as shown in figure

Concept fragments	
The system shall update cloud coverage data.	**The system shall recalculate the planned route based on updated cloud coverage data.**
Periodic data update at fixed interval	Machine learning based route prediction (trained on sun exposure data)
~~Event-driven data update (e.g., sudden cloud changes)~~	Multi-objective function optimization (exposure vs time vs safety)
~~Push notification from weather data center~~	~~Predictive route simulation (using forecast data)~~
Hybrid update (periodic + event-triggered)	Adaptive recalculation (trigger only if updated data crosses threshold)
~~Machine learning-based data update (predict next cloud coverage)~~	~~Constraint-based route recalculation (consider user priorities & safety metrics)~~
Waypoint-triggered update	~~Real-time optimization using current GPS & environmental data~~
~~Crowdsourced, smart city sensors data input~~	~~Segment-level recalculation (recompute only affected portion of route)~~
~~Adaptive interval update (update frequency changes based on environment variability)~~	~~Scenario-based planning (simulate multiple route options)~~
Sensor-triggered update (onboard light sensor, weather sensor)	Solar weighted shortest distance algorithm(djkstra algorithm)
~~Adaptive machine learning retraining (periodically improve prediction model)~~	
Notes	
Other data update methods have not been considered due to implementation complexity.	
Alternative route calculation methods were excluded because of lower accuracy and higher memory requirements.	
Other route verification approaches were not selected due to computational intensity.	

Table 3: Concept Fragments

This gave a clear view of most feasible and optimal solution for critical requirements and helped identify potential risk associated with other approaches. Once solutions for all requirements were identified, they were combined in a concept combination table to create cohesive solution across related requirements, ensuring the overall system design is both practical and well integrated.

12. Combination Tables / Morphology Boxes

Combination tables are used in systems engineering to merge solutions identified in concept fragments for interdependent requirements into a single, cohesive system. After generating concept fragments for individual requirements, combination tables help systematically explore which solutions can work together, ensuring compatibility and identifying most feasible overall system design. This tool allows engineers to visualize interaction between different solutions to make informed decisions before finalizing systems architecture.

Using combination table, we understand for verification of updated routes is related to recalculation of new routes. Also, which solution of route calculation can be paired with verification methodology. For example, when using a multi objective function optimization algorithm, the threshold for calculation is already updated, so verifying new route efficiency can rely on these parameters without requiring additional user input. In contrast, a machine learning based sun prediction model using only sun exposure data requires user confirmation which basically accounts for time required to reach to destination as shown in figure.

The system shall recalculate the planned route based on update	The system shall verify benefit of new route
Multi-objective function optimization (exposure vs time vs safety)	Threshold-based comparison (exposure < current exposure, time increase < 0.2×current time → suggest route)
Machine learning based route prediction (trained on sun exposure data)	User confirmation (ask user before suggesting route)
Solar weighted shortest distance algorithm(djkstra algorithm)	User confirmation (ask user before suggesting route)
Adaptive recalculation (trigger only if updated data crosses threshold)	Weighted cost comparison (α×exposure + β×time > threshold → suggest route)

Table 4: Combination Table

This tool helped us better define the optimal combination of solutions to fulfill related requirements and to finalize the desired option from available options we used decision matrix. Once the solution was finalized it allowed to create system architecture to visualize how these solutions integrate with total system.

13. Functional Flow Block Diagram

A functional flow block diagram ius a systems engineering tool used to represent the sequence and interaction of function within a system. It is used for functional modeling of systems and provides a detailed view of how high-level system operations are carried out through different functions, showing the logical flow, decision points and dependencies between them. FFBD helps translate rough sketches or high-level use case ideas into structured representation of systems operation, making it easier to analyze functional requirement as an integrated system.

The FFBD was used once the high-level system functions were visualized and critical functional requirements were finalized. Using this diagram, we could understand which requirements are involved in each mode of operation and how system operates between them. For example, the system attempts to use real time data by default but switches to offline data if real-time data is unavailable as shown in figure

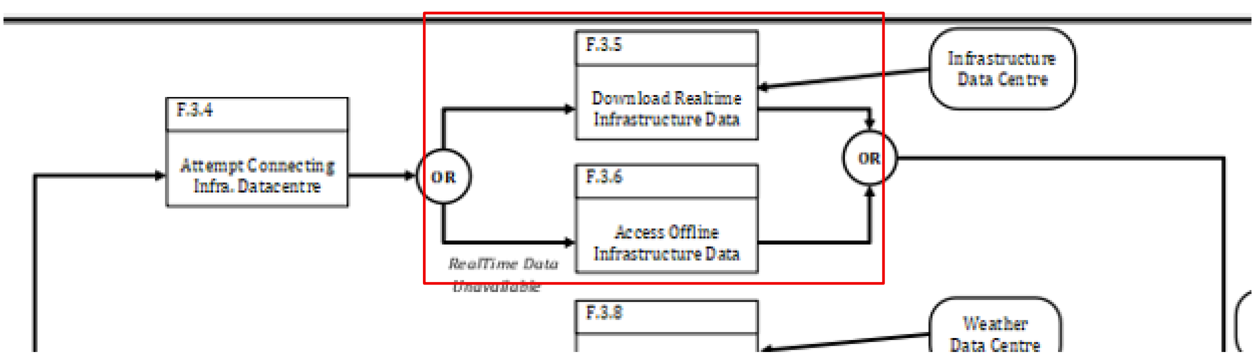


Figure 7 : Decision Making in FFBD

The diagram also helped define decision making points, parallelly operating functions and essential functions needed for high level operations. While FFBD clarifies functional flow, it doesn't provide details about how this function will be implemented on subsystems level. This leads to further design work including concept fragment generation and combination table, to identify detailed operations within each functional block and ensure cohesive system design.

14. IDEF0

IDEF0 is a systems engineering tool used to model the major functions of a system and the data required for each function. While the function flow block diagram defines the sequence of functions and their interaction and decision-making points, IDEF0 in system architecture definition focus on understanding the inputs, outputs, controls and mechanism for each high-level function. It provides hierarchical view of function operations and the associated data flow, helping to understand how information moves through system.

IDEF0 was used to analyze the data flow within the route planner model for example, the route planner model has level 2 functions such as initialize model and plan route as shown in figure

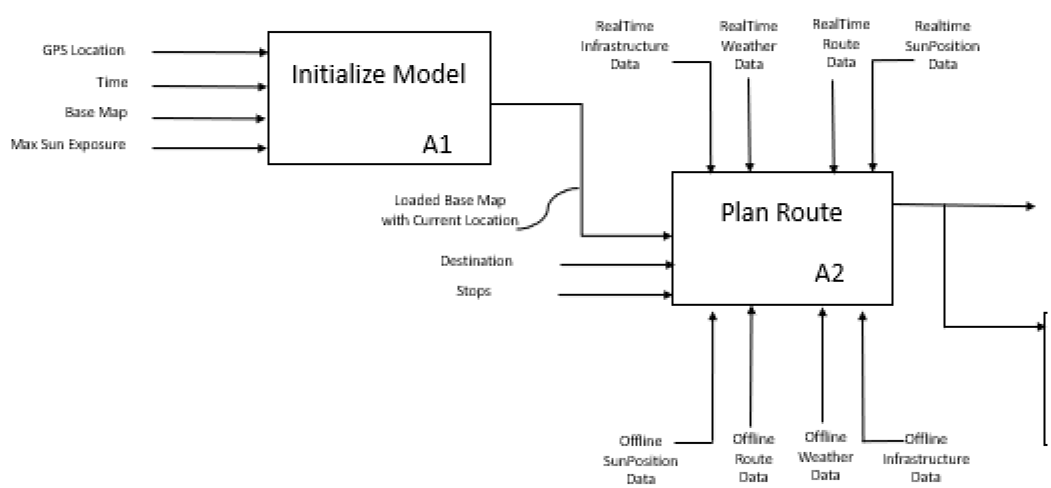


Figure 8 : IDEF0 Level-2

The initialized model function creates a loaded base map with the current location, destination and stops, which is then used by plan route function to calculate the final route. Using IDEF0 we could understand which data is essential for each high-level function, which subsystem generates data and where it is consumed next. This tool also allowed us to define the system by breaking down functional architecture into a hierarchy and identifying inputs and outputs for each operation at different levels.

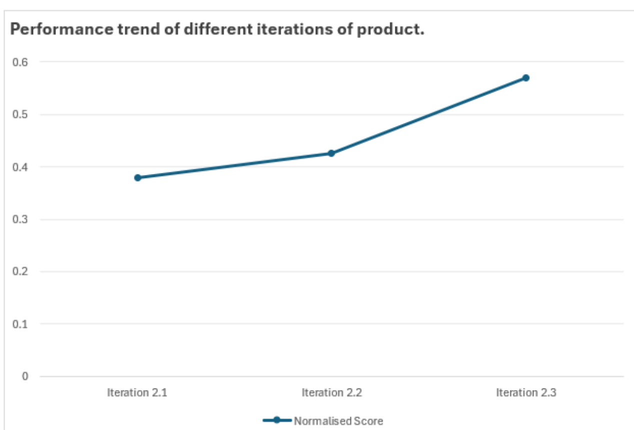
The information from IDEF0 guided further design work, including subsystem allocation to create a more cohesive system. Additionally, this data can be used further for interface definition in ODT

15. Decision Matrixes

A decision matrix is a tool used in systems engineering to evaluate and compare multiple options against a set of defined criteria to make a decision. It helps prioritize alternative based on quantitative or qualitative performance measures, ensuring that decisions are objective and traceable. The decision matrix can be used at any stage of project life cycle to select most suitable solution amongst the competing options.

In this project, we got the different combination option from morphology table representing different iteration in diagram. To select desired combination from option, we evaluated the iterations against performance criteria like accuracy, response time, battery impact. By evaluating iterations against performance criteria, we could safely choose configuration who achieved optimal results across all requirements. For example, iteration 2.3 was chosen because it outperformed the other two in overall performance. As shown in figure

Plot the results by Final Score and look for a *kink* or *drop-off* in the scores.



Iteration	*Iteration 2.1*	*Iteration 2.2*	*Iteration 2.3*
Normalised Score	0.38	0.43	0.57

Figure 9 : Comparative performance obtained after Decision Matrix

The decision matrix also highlights areas where the system perform well and where improvements could have highest impact. This analysis helped me finalize the concept fragment for best performing system and led to further design work such as identifying engineering characteristics and defining structural requirements for a chosen solution.

16. Goal – Question – Metric GQM

The Goal Question Metric is a tool in systems engineering used to systematically quantify performance goals of a system. It starts with performance goals for given system and associated questions to better understand those goals which help define the quantitative value for given performance goals. Even though GQM does not directly provide values for given goals, it creates structured methodology to gather questions and data to evaluate those values so that unrealistic goals are not set during primary phase of project.

In this project GQM was used after defining functional requirements to quantify system performance goals. For example, to achieve the goal of ‘minimizing sun exposure during the journey, it was necessary to understand points like how much sun exposure is harmless and what is the shadow coverage generally available in urban landscapes as mentioned in snippet.

Sr.No	Goals	Questions	Ideal Metric	Approximate Metric	Data collection method
1	Make model to reduce sun exposure as much as possible	What is medically recommended duration of daily sun exposure for different user profiles? (e.g skin type, UV index)	Recommended sun exposure (in minutes) for different skin types for standard UV index.	General exposure time range across moderate UV levels for average skin types.	Literature review from health agencies like WHO, dermatology associations.
		What is average drive time per user?	Total travel during the day	Travel time for just the major drives like going to office college.	For few cities and rural areas contact major institutions like offices and colleges to know from how long those people travel everyday to reach there and use google maps to know the time to tarvel'
		What proportion of urban and rural area is typically under natural or artificial shade?	percentage of shade available at different locations for specific times of the day.	Percentage of shade available for given season of the year and within specific time zone	Analysis in digital twin for refernce cities and rural areas ensuring at least one urban and one rural model across different demographics.
2	Make response time of model as low as possible	What is the acceptable system response time for generating route plan for this application context?	Route planning time recorded when user exits specifically because of the delay in the route planning	Route planning time recorded when user exits midway for any reason during route planning	System logging of route generation times and user exit data for every interation.

Table 5 : GQM

GQM helped us understand efficient ways to collect data such as using average sun exposure rather than gathering data for every possible scenario. Using this tool, we could define the data collection method for all goal related questions. Once data was obtained, it allowed us to quantitatively define the practical performance goals.

17. Analytical Hierarchy Process

The Analytical Hierarchy Process is a decision-making tool used to determine the relative importance of different goals in a system. Since all goals created in GQM do not create same level of impact, AHP helps to assign all goals into high level objectives and use those objectives to assign weightage. This is useful tool in complex system where different goals like performance, reliability and safety compete with each other

In this project, AHP was used to understand the relative importance of different system goals. By using this tool, we understood which goals contribute most to overall system effectiveness. For example, reduction of sun exposure emerged as the most important goal with highest weightage of

0.182, indicating its dominant influence on system performance compared to other goals. As shown in figure

1	**Optimise the model performance**				**Improve the model reliability**			**Enhance the user experience**			**Ensure model safety and security**
2	0.4				0.2			0.3			0.1
3	**Enhance the computational performance of model**		**Enhance the functional poerformance of model**								
4	0.3		0.7								
5	Optimize CPU/GPU usage	Reduce the battery impact of model	Make model to reduce sun exposure as much as possible	Reduced the increase in travel time for model	Maintain the prediction accuracy of model in offline mode	Maintain the prediction accuracy of model in online mode	Make model to realtime adapt with changes in cloud coverage	Make response time of model as low as possible	Make model easy to use	Make model easy to install	Ensure privacy and data security of user
6	0.4	0.6									
7			0.65	0.35							
8					0.35	0.4	0.25				
9								0.6	0.3	0.1	
10											1
11	0.048	0.072	0.182	0.098	0.07	0.08	0.05	0.18	0.09	0.03	0.1

Table 6 : AHP

This helped us better define which goals the system must focus on to deliver maximum impact, in our case significant portion of system performance was depended upon reducing sun exposure and minimizing response time. The resulting weights were used in the decision matrix further to do performance comparison for different concept fragments to select one with better performance.

18. QFD (House of Quality)

Quality Function Deployment, also known as house of quality, is a systems engineering tool used to translate customer needs and system goals into engineering characteristics. It provides a structured way to link goals with technical design of system. QFD also enables comparison of our system with other competitors in the market.

In this project, QFD was used to assess the system from a customer perspective and compare it against competitors like Google Maps, SunQuest etc., This helped us to understand how our system was performing against the key competitors in the critical goal segments like travel time and sun exposure reduction. Additionally, it shows how these goals are linked to specific engineering characteristics. For example, reducing sun exposure is strongly related to offline storage capacity, since higher storage allows more detailed weather data to be stored, improving route prediction accuracy in offline mode.

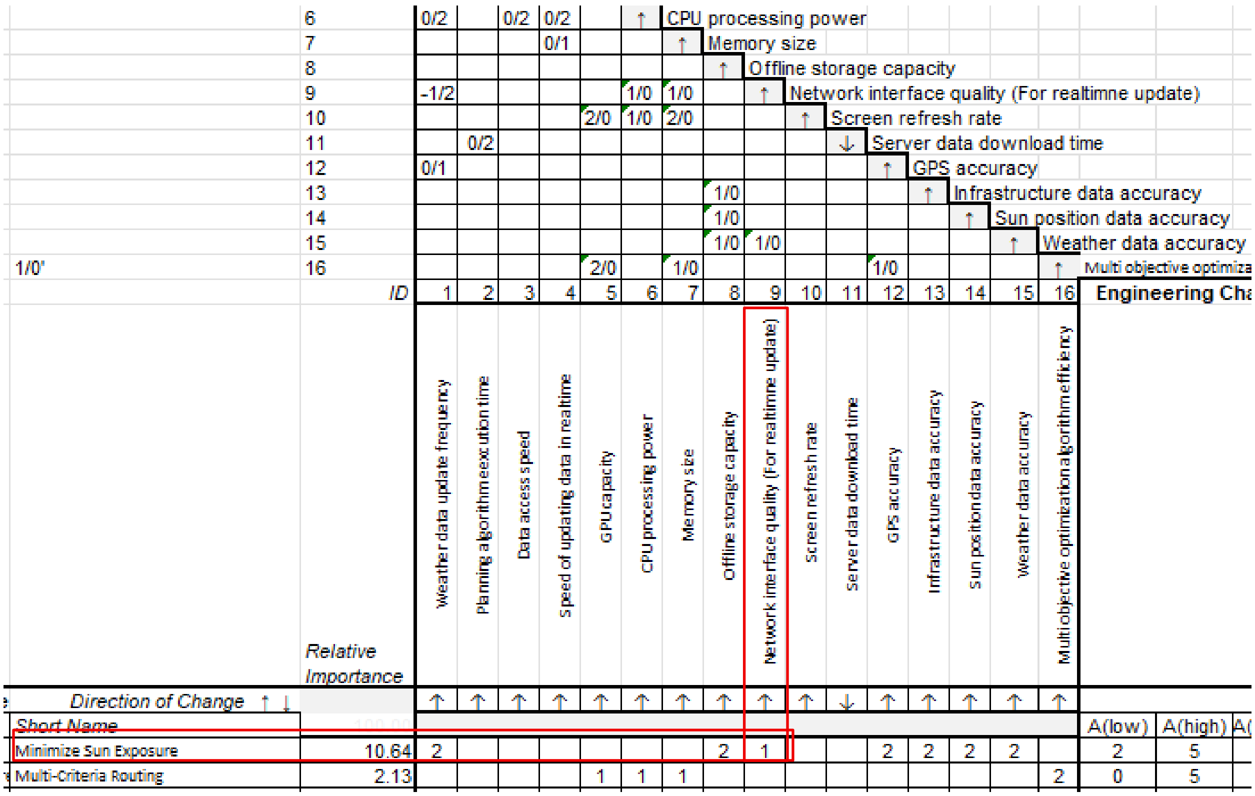


Figure 10 : Goal & Engineering characteristic in QFD

QFD helped to better define how system goals map to engineering requirements like GPS accuracy, memory size and network interface quality, highlighting which engineering characteristics are most important for system performance. This analysis further led to design work such as defining a clear CVP based on goals and competitor data and using identified engineering characteristics as inputs for failure mode detection and risk assessment associated with design choices.

19. Sub-System Definition & Allocation

Subsystem definition and allocation is a systems engineering activity used to decompose the overall system into smaller, manageable subsystems. This is usually done after all originating requirements are defined and functional modeling tools such as FFBD and IDEF0 are created. The objective is to group related requirements in a way that minimizes interfaces and maximize cohesion across different subsystems.

In this project, subsystem definition and allocation helped to understand which subsystem collectively forms the more cohesive system. For example, for subsystem definition instead of creating separate subsystems for online and offline data handling, a single data management subsystem was defined. This definition helped to reduce the number of interface interactions that could happen during data management operations.

User interaction and model management system	Data management system	Route planning system	Navigation execution system
OR.2 A system shall detect destination input on map.	**OR.3 The system shall track its GPS location constantly.**	**OR.9 A system shall update the route based on cloud coverage information in weather data.**	**OR.3 The system shall track its GPS location constantly.**
Touch interaction	*Onboard GPS receiver*	**OR.11 The system shall provide an alternate route to avoid restricted area.**	*Onboard GPS receiver*
Search based input	*Combine GPS with internal measurement unit*	**OR.16 The system shall plan route with minimum Online_Sun_Exposure_Limit percentage sun exposure when using real-time data**	*Combine GPS with internal measurement unit*
OR.10 The system shall allow user to add upto MAX_STOPS stops	**OR.4 The system shall connect to remote data centre to retrive realtime sun position data**	**OR.17 The system shall plan route with minimum Offline_Sun_Exposure_Limit percentage sun exposure when using offline data**	**OR.26 The system shall update navigation window with new route.**
OR.18 The system shall detect stops on the map.	*send a query to server with GPS location and time to retrive all relevant data(azimuth, elevation,maybe irradiance)*	**OR.19 The system shall plan an alternate route with minimum Online_Sun_Exposure_Limit percentage sun exposure when planning alternate route due to restricted areas.**	**OR.20 A system shall allow user to pause navigation**
Touch interaction	*Retrive only essential parameters (sun elevation and azimuth)*	**OR.23 The system shall recalculate the planned route based on updated cloud coverage data.**	**OR.36 The system shall provide voice-guide navigation to the user.**
Search based input	**OR.5 The system shall connect to remote data centre to retrive real-time infrastructure data**	*Machine learning based route prediction (trained on sun exposure data)*	
OR.21 A system shall allow user to change destination during	*Maintain local cahe of infrastructure and upodate deltas*	*Multi-objective function optimization (exposure vs time vs safety)*	
Interrupt and recalculate	*Request essential information relevant to current start and destination*	*Adaptive recalculation (trigger only if updated data crosses*	
OR.28 The system shall allow the user to set priorities for trade-offs between solar exposure and travel time.	**OR.6 The system shall connect to remote data centre to retrive realtime route data**	*Solar weighted shortest distance algorithm(djkstra algorithm)*	
OR.29 The system shall allow the user to modify default limits for sun exposure.	*Request complete route data for a city*	**OR.25 The system shall incorporate safety metrics, including traffic congestion, accident hotspots, and road conditions, in route planning.**	
OR.30 The system shall warn the user if the configured sun exposure limit exceeds recommended healthy levels.	*Maintain local route plan and request only changes(due to traffic, closure)*	**OR.24 The system shall verify benefit of new route.**	
OR.27 The system shall record and display sun exposure avoided for each trip.	**OR.7 The system shall connect to remote data centre to retrive realtime weather data**	*Threshold-based comparison (exposure < current exposure, time increase < 0.2×current time → suggest route)*	
OR.31 The system shall power on upon user command.	*Request only the weather parameter needed.(sunlight,cloud coverage, precipitation)*	*Weighted cost comparison (α×exposure + β×time > threshold → suggest route)*	
OR.32 The system shall power off upon user command.	*Maintain catched copy of local data request realtime update for immediate area.*	*User confirmation (ask user before suggesting route)*	
	OR.3 The system shall track its GPS location constantly.		
	Onboard GPS receiver		
	Combine GPS with internal measurement unit		
	OR.22 The system shall update cloud coverage data.		
	Periodic data update at fixed interval		

Table 7 : Subsystem Matrix

This tool helped us define the responsibilities of each subsystem and which requirements are part of which subsystem operations. Once subsystems and their corresponding requirements were finalized , it naturally led to further design work like identifying interfaces between the different subsystems using Operational Description Tree (ODT) and further detailing those interfaces using interface matrix.

20. ODT

The Operational Description Template is a system engineering tool used to describe how subsystems interact during different operational scenarios. It captures the flow of information, decisions and constraints (e.g. Time) across subsystem for different use cases. ODT focuses on defining interfaces and operational behavior across subsystems.

In this project, ODT was used after defining all subsystems to clearly identify the interfaces between them. It helped us understand which subsystem needs to share what information, what kind of events triggers action and where key decisions are being made. For example, during the use case of route recalculation, the data management subsystem decides whether to recalculate the route or not based on the updated weather data. It also has constraint to recalculate route within 10 sec.

Operator (Driver)	Model management and UI	Route planner	Data Management	Navigation	Data centre	State	Timing Target
The driver follows path suggested by navigation system						Online, Moving, Standby,Navigation	
			The system shall update cloud coverage data.			Online, Moving, Data Acquisation,Navigation	
			Info even("Current GPS Location")				
					Data centre will provide updated cloud data. **-Available** -Unavailable(Inform user)		
					Info event("Data is available")		
			The system shall determine whether to recalculate the route based on CHANGE_WEATHER % . **-Recalculate** -Do not calculate				10s
			Infor event ("Recalculate the route")				
		The system shall recalculate the planned route based on updated cloud coverage data.				Online, Moving, Calculate ,Navigation	
		The system shall verify benefit of new route. **-Efficient route** -Inefficient route (Do not suggest user)					
		Info Event("New route is more efficient")					

Table 8 : ODT

Additionally, it defines the overall system state reflective of different subsystem level operations. This tool provided holistic view of subsystem behavior across different interfaces for different use cases, including decision, constraints, states and information exchange. This led to further design work such as creating state diagrams to analyze system state transition.

21. State Diagram

A state diagram is a behavioral modeling tool used to represent the different states of a system and the transition between those states under various operating conditions. It shows entry and exits criteria for each state and identify which states are required for certain operations. State diagrams are typically created after system states have been identified through tools like ODT and activity diagrams.

In this project, the state diagram was used to understand how the system transitions between states during different operations and to identify the essential states needed for specific functions. Using this tool, we learned how system moves across online offline, data acquisition and calculation states. For example, the diagram shows that calculation state can be entered from both online or offline data mode but for online to recalculate it needs to go through data acquisition as system has to fetch data from server.

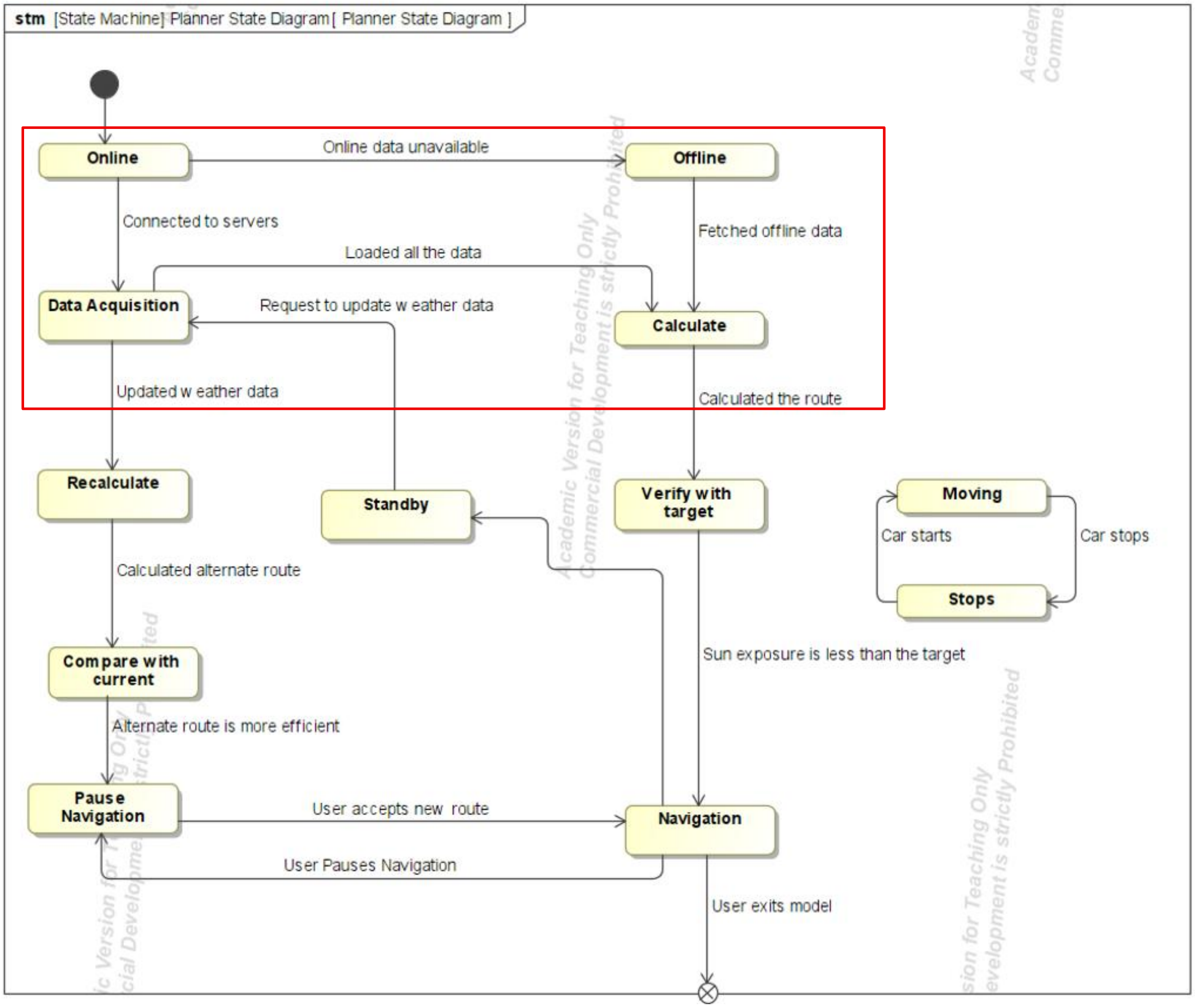


Figure 11 : State Diagram

This diagram helped to define state dependencies as well as entry and exit conditions for each state. The clarity obtained from the state diagram was used in interface matrix to define the interface interactions better.

22. Interface Matrix

An interface matrix is a detailed systems engineering document used to capture and manage interface information between different subsystems. While ODT and sequence diagram is used to define functional interface, the interface matrix helps to define the structural interface across different subsystems. It is developed after ODT definition to clearly document by what amount the design choices in one subsystem can constrain the other.

In this project, the interface matrix was used to understand the constraint and associated data needed from other subsystem teams. By referring to this document I could identify interface limitations that directly impact the system behavior. For example, the geographical radius of real-time data that the route planning system can use us constrained by data managements systems data storage capacity. Hence the geographical data storage radius has been set to 50km for our case

Interface Matrix Excerpt : Data Management Subsystem

Data Management	User interaction and model management	Route planning system	Navigation execution system	Value	Units	Estimate?
	Received from				Degrees/minutes/seco	
	Received from			5	N/A	
	Received from				Degrees/minutes/seco	
		Provided to		500	KB	
		Provided to		50	km	
		Provided to		20	MB	
		Provided to		500	ms/s	
		Provided to		100	ms/s	
		Provided to		70	km	
				1	GB	
	Received from			15	days	
	Received from			1	month	
	Received from			30	days	
	Received from			3	months	
	Received from			10	min	
					boolen	
				10		
		Received From		10%		
				8	GB	

Table 9 :Structural Constrained by Subsystems

The interface matrix also helped track updated interface values, due data for providing data to other subsystems and person responsible for each interface. This helps to define design constrained on our different subsystems. This information further supported risk identification and test cases definition for given subsystem making it as strong input for test and risk planning.

23. Sequence Diagram (SysML)

A sequence diagram is used to represent the functional interaction between different subsystems for different use cases. While the interface matrix captures the detailed structural interface information, the sequence diagram focuses on the data exchange, triggers between different subsystems. It is created after the subsystem definitions and allocation to describe how subsystems collaborate to fulfill system level functions

Using the sequence diagram, I was able to understand how different subsystems interacted over time to accomplish given use case and how long each subsystem remains active during that use case. It also talks about what kind of data acts as a trigger for operation in given subsystem for example, model management sends start and end location of route to data management to fetch a data between those locations and it activates the data management system as shown in figure

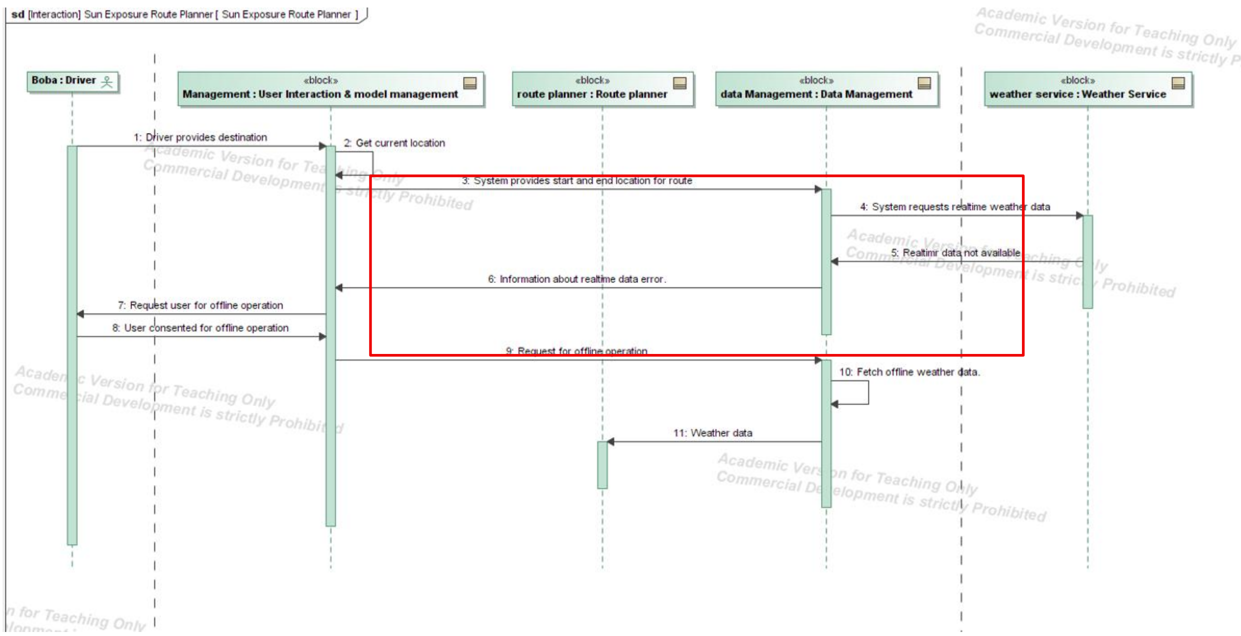


Figure 12 : Input Data to Data Management Subsystem

Additionally, it helped to understand the dependency chain , showing how a failure in one subsystem can impact other. These insights from sequence diagram supported further design work such as failure mode analysis allowing to identify vulnerable subsystem interaction and improve subsystem robustness by addressing potential functional failures.

24. Behavioral Test Plan

A Behavioral Test Plan is a systems engineering tool used to define and organize tests for verifying the functional behavior of a system. It is generally created once subsystems are allocated and functional interfaces are defined. The test plan provides a structured approach to ensure that all functional requirements are tested, specifying the methods, facilities, entry and exit conditions, and expected outcomes for each test.

Using the behavioral test plan, we were able to identify which tests need to be performed to validate the functional behavior of the system. It also helped define details such as test methods, required facilities, and the entry and exit criteria for each test. This allowed me to better understand the scope and conditions of testing, including when tests should be performed during development and the parameters that must be satisfied to pass a test. For example, all 4 tests can be performed once prototype for a model is ready as shown in figure

Test Number	Test Method	Test facilities	Entry Condition	Exit Condition
TP.1	Test procedure: Model plans route with online data	Model prototype, Good internet connection	Prototype for online planner subsystem and data management subsystem is ready and integrated in digital twin.	Online model suggest a route with accurate estimate of sun-exposure reduction and travel time with 90% of accuracy
TP.2	Test procedure: Model plans route with offline data	Model prototype , Offline data	Prototype for offline planner subsystem and offline data management subsystem is readyand integrated in digital twin.	Offline model suggest a route with accurate estimate of sun-exposure reduction and travel time with 87% of accuracy
TP.3	Test Procedure : Model replans route during navigation as per updated weather condition	Model prototype, Good internet connection	Prototype for replanner subsystem and data management subsystem is ready and integrated in digital twin.	In scenarios where rerouting options were available, the planner provided route suggestion in 96% of cases.
TP. 4	Test Procedure :Model reduces sun exposure	Model prototype	Prototype for online planner subsystem and data management subsystem is ready and integrated in digital twin.	The model achieved 23% of sun exposure reduction relative to popular route in 90% of cases.

Table 10 : Behavioral Test Plan

Once the test plan was established, it could be traced back to individual functional requirements using the Verification and Validation Cross Reference Matrix, ensuring that every requirement is thoroughly verified before the final system release.

25. Test Methodologies for Non-Behavioral Requirements

Test Methodologies for Non-Behavioral Requirements are systems engineering tools used to verify the structural or non-functional aspects of a system, such as performance, engineering characteristics and interface compliance. These methodologies define how structural requirements will be validated using appropriate verification methods, such as analysis, inspection, demonstration, or physical testing. Similar to a behavioral test plan, it also captures entry and exit conditions, required facilities, and expected outcomes for each test.

Using this tool, we were able to identify which tests are necessary to satisfy structural requirements and determine the appropriate verification method for each requirement. It also provided details on test facilities, the stage of development when tests can be performed, and the criteria that must be met to pass a test. Hence giving us intuition about which tests I can perform at earlier stages for example for TP.5 and TP.6 only test model in computer is enough so it can be performed at the early stages of project life cycle.

Requirement number	Requirement	Abstract name	Test number	Test method	Verification method (A,I,D,T)	Test Facilities	Entry Condition
OR.38	The system shall process and update rel time data with maximum latency of 500ms/s	Real time responsiveness	TP.5	Measure actual latency using test data	A	Test computer with model installed, Data centre, Internet connection	Real time data management subsystem is ready
OR.1	The system shall not store offline data more than 1GB	Offline storage	TP.6	Inspect total data stored for 15 days of weather forecast, 30 days sun position , infrastructure and route data of 50km radius for diffrent season of the year	I	Test computer with model installed, Data centre, Internet connection	Prototype for offline data management is ready
OR.40	The system shall operate without exceeding 80% of available system memory or CPU usage during route calculation	CPU/Memory usage	TP.7	Monitor CPU/Memory usage with performance profiling tool with route calculation under different conditions	T	Test device with model installed, Performance monitoring software	Complete system is ready
OR.3	The system shall track its GPS location with 1Hz.	Position update rate	TP.8	Observe GPS update and verify its frequency	D	Model installed in actual car(Outdoor testing environment)	GPS enabled, System is powered on
OR.42	The system shall shall maintain a minimum data synchronization speed of 3Mbps for download under good internet connectivity	Download speed	TP.9	Measure download speed during synchronisation	T	Model with network access, Network simulation environment	System connected to good internet. sync process is ready

Table 11 : Test methodologies for Non-Behavioral Requirements

It allowed me to better define how structural requirements are verified, ensuring clarity on test methods, conditions and timeline. Once these methodologies were established, the tests were incorporated into the Verification and Validation Cross Reference Matrix to ensure that all structural requirements are properly linked to their corresponding tests, completing the verification coverage for the system.

26. Verification Cross Reference Matrix (VCRM)

The Verification Cross Reference Matrix is a systems engineering tool used to ensure that every system requirement has corresponding test case. It provides a traceable link between requirements and their verification methods, ensuring that no requirement is left untested before the final system release. VCRM helps maintain a structured overview of the verification plan for both functional and non functional requirements.

Using the VCRM in this project, I could identify which test cases correspond to which requirements and highlight any requirements that still lacked planned tests, such as requirements 8, 10, and 11 in the current example.

	OR.1	OR.2	OR.3	OR.4	OR.5	OR.6	OR.7	OR.9	OR.12	OR.13	OR.14	OR.15	OR.16	OR.17	OR.22	OR.23	OR.24	OR.38	OR.39	OR.40	OR.41	OR.42	OR.43
TP.1				X	X	X	X						X										
TP.2									X	X	X	X		X									
TP.3								X							X	X	X						
TP.4																							X
TP.5																		X					
TP.6	X																						
TP.7																				X			
TP.8			X																				
TP.9																						X	

Table 12 : VCRM

This allowed us to better define which test procedures are executed for each requirement and ensure complete coverage. Once testing is completed, the data collected can be used to identify potential risks and failures in the system, assess their likelihood, and plan mitigation strategies using tools like FMEA. The VCRM therefore serves as a bridge between verification activities and risk management, supporting a robust and reliable system design.

27. Severity Rating System & Likelihood Rating System

The Severity Rating System and Likelihood Rating System are tools used in Failure Mode and Effects Analysis to quantitatively define the potential impact and probability of failures within a system. The severity rating defines how serious a failure would be if it occurs, while the likelihood rating defines how often a failure is expected to happen. These ratings provide a standardized way to assess the criticality of failures for a given system.

Using these tools, I was able to understand clear definitions for severity and likelihood levels specific to our system. It is important to define the subjective severity and likelihood of a system because different systems have different acceptance to failures. For example, since solar route planner is not a very critical system failure once in a 3 month is defined as very less likelihood with value 1

Severity	
5	Require more than 20 days to address Costs more than 15% of total budget Performce reduction of >50% System failure impacts usability of system Doesnt cause any health discomfort
4	Require more than 15 days to address Costs more than 10%-15% of total budget Performce reduction of 30-50% System failure requires frequent manual intervention More sun exposure than optimal
3	Require more than 10 days to address Costs more than 8%10% of total budget Performce reduction of 15-30% System failure needs manual inputs Noticable discomfort but safe
2	Require more than 5 days to address Costs more than 2%-8% of total budget Performce reduction of 5-15% System failure can be undone by workarounds Sunburn and heat discomfort
1	Require more than 1 day to address Costs more than 1%-2% of total budget Performce reduction of <5% System failure doesnt impact user convinience Significant sunburn

Likelihood	
5	Occurs greater than once a week, Occurs in greater than 10% of the operations
4	Occurs greater than once a week, Occurs in 5-10% of the operations
3	Occurs no more than once a week
2	Occurs between once every 2 weeks to once every 3 months
1	Occurs less than once every 3 months

Figure 13 : Severity & Likelihood Matrix

It allowed us to uniformly evaluate failure modes across the system and assign severity and likelihood values consistently. The ratings then feed directly into the FMEA process, helping prioritize which failure modes require attention and mitigation, and providing a structured basis for designing more reliable and safe system operations.

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

The Risk Priority Number table is a systems engineering tool used in FMEA to quantify the criticality of individual failure modes by multiplying their severity and likelihood ratings. This allows the team to identify which failure modes require immediate attention. Complementing this, the stoplight graph visually represents the distribution of failure causes across different criticality levels, providing a holistic view of the system risk profile at a glance.

Using this tool, we were able to classify failure causes based on their criticality and determine which ones needed urgent corrective action. For example, failure modes with an RPN between 10–25 were identified as high-risk and prioritized for mitigation.

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

16-25	High Risk
10-15	Medium High Risk
5-9	Medium Risk
3-4	Medium Low Risk
1-2	Low Risk

Figure 14 : RPN Table

After implementing corrective actions, the stoplight graph was updated to reassess the system, showing the impact of these mitigations. This approach allowed us to systematically define high impact failure causes and provided a clear method to classify and manage risks. The insights from the RPN table and stoplight graph were then used iteratively within the FMEA process to monitor the effectiveness of mitigation strategies and ensure continuous improvement in system reliability. Here as we can see changes in FEMA spotlight RPN diagram after the design changes.

Likelihood					
5	1	1	0	0	0
4	0	1	2	1	0
3	1	0	0	3	1
2	1	1	2	0	0
1	0	0	1	2	1
	1	2	3	4	5
			Severity		

→

Likelihood					
5	1	0	0	0	0
4	0	0	0	0	0
3	1	5	0	0	0
2	2	2	3	0	0
1	0	1	1	2	1
	1	2	3	4	5 Severity

Figure 15 : Change in RPN table after design iteration

29. Failure Mode and Effect Analysis (FMEA)

Failure Mode and Effect Analysis is a tool used to identify potential failure modes in a system to analyze their causes and effects, and prioritize them based on risk. It also tracks the implementation and impact of risk mitigation strategies, providing a structured approach to enhance system reliability and safety. FMEA integrates severity, likelihood, and detection measures to guide decision-making for reducing critical failures.

Using FMEA in this project, we could identify which failure mode is caused by which causes and associated criticality of each failure mode. Additionally, it helped determine which failure causes needed urgent attention and what corrective actions were effective in reducing associated risks. For example, for failure mode F.3 severity dropped from 4 to 2 after increasing the frequency of update for weather data

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(hrs)	
F.1	Route planner	Optimization failure	Suggesting inefficient route in terms of time and sun exposure. Increased sun exposure, Increased travel time	Extreme optimization parameters by user	5	3	15	MEDIUM HIGH		Put threashold on maximum optimization allowed	2	1	2	LOW	10	
				Incorrect algorithm implementation of multi objective optimization	5	1	5	MEDIUM								
F.2		Incorrect time required prediction	Reduced accuracy of prediction in online and offline mode, Increase travel time	Unexpected detours and blocked roads not accounted for	2	5	10	MEDIUM HIGH		Update the route for unexpected route with the frequency of 15min	2	2	4	MEDIUM LOW	8	
				GPS signal errors affecting speed estimates	2	4	8	MEDIUM		Incorporate dead reckoning system to update and calibrate GPS locally.	1	2	2	LOW	40	
				Traffic or road condition data is not update in real time	2	2	4	MEDIUM LOW								
F.3		Incorrect sun exposure prediction	Reduced accuracy of prediction in online and offline mode, Increase sun exposure.	Outdated weather or sun position data	4	3	12	MEDIUM HIGH		Increase the frequency of update for offline weather and sunposition data	2	3	6	MEDIUM	6	

Table 13 : FMEA

This tool allowed us to better define the risk criticality associated with each failure cause, track the impact of mitigation actions, and ensure that design changes effectively reduced system vulnerabilities. Insights from FMEA then guided further design work to minimize potential failures and improve overall system robustness.

30. Event Tree or Fault Tree

Event Tree or Fault Tree analysis is a systems engineering tool used to evaluate the probability of system level failures by showing the logical relationships between failures of individual subsystems to the overall system outcome. While FMEA focuses on detecting failures at the subsystem level, fault trees help identify which combinations of events or subsystem failures can lead to critical system level issues, providing a quantitative understanding of system reliability.
This tool allowed us to better define the causes and probabilities of such failures at the system level. Using this tool, I was able to identify which functionalities contributed most to the probability of system level failures. For example, the highest risk of suggesting a non optimal route was traced back to weak GPS signals using the fault tree.

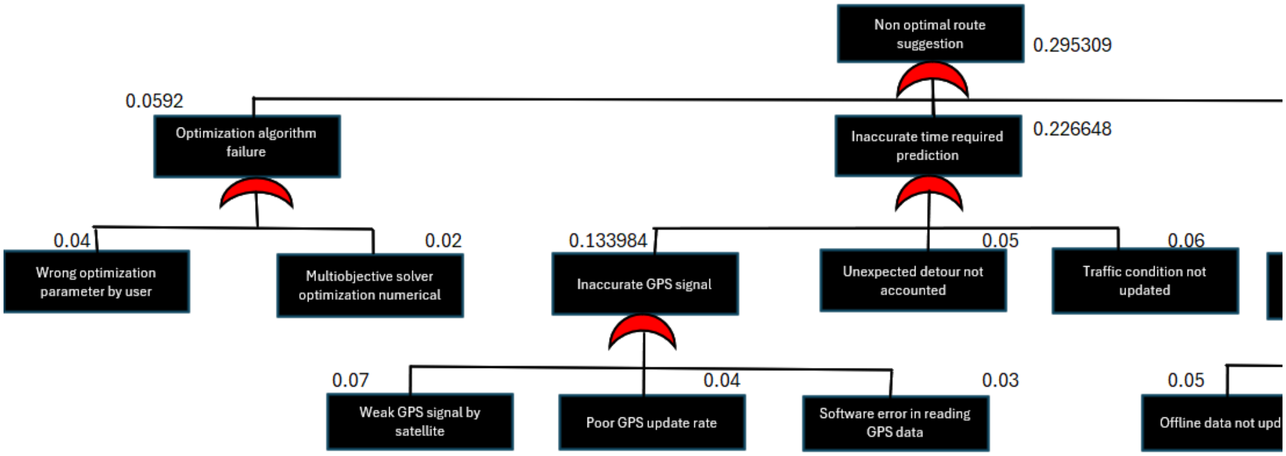


Figure 16 : Fault tree for Non-Optimal Route error

Based on these insights, further design work was implemented to improve GPS robustness by adding a dead reckoning system that uses internal sensors to verify GPS data. This reduced the probability of weak GPS from 0.7 to 0.2, lowering the chance of non optimal route suggestions to 0.24, thereby improving overall system reliability.

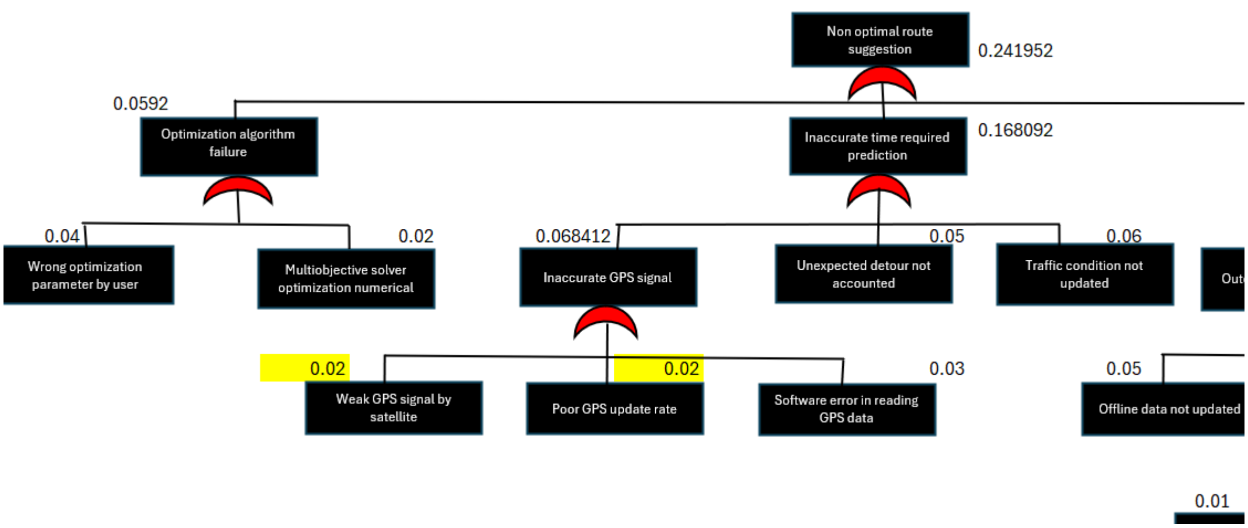


Figure 17 : Fault tree for Non-Optimal Route error after design change

31. SysML Parametric Dia.

SysML parametric diagram is a diagram used to model mathematical constraints and relationship between different system parameters. It is developed during the detailed definition of the system to check if system is behaving as expected or not

Using this tool, I understood how different parameters of systems are mathematically related to each other. It also helped me to understand whether given methodology will give me desired route efficiency value or not. For example. In the diagram what will be the different route efficiency values I can get for different conditions of sun exposure intensity (w_ij) and weightage combination for a different route segment (d_ij).

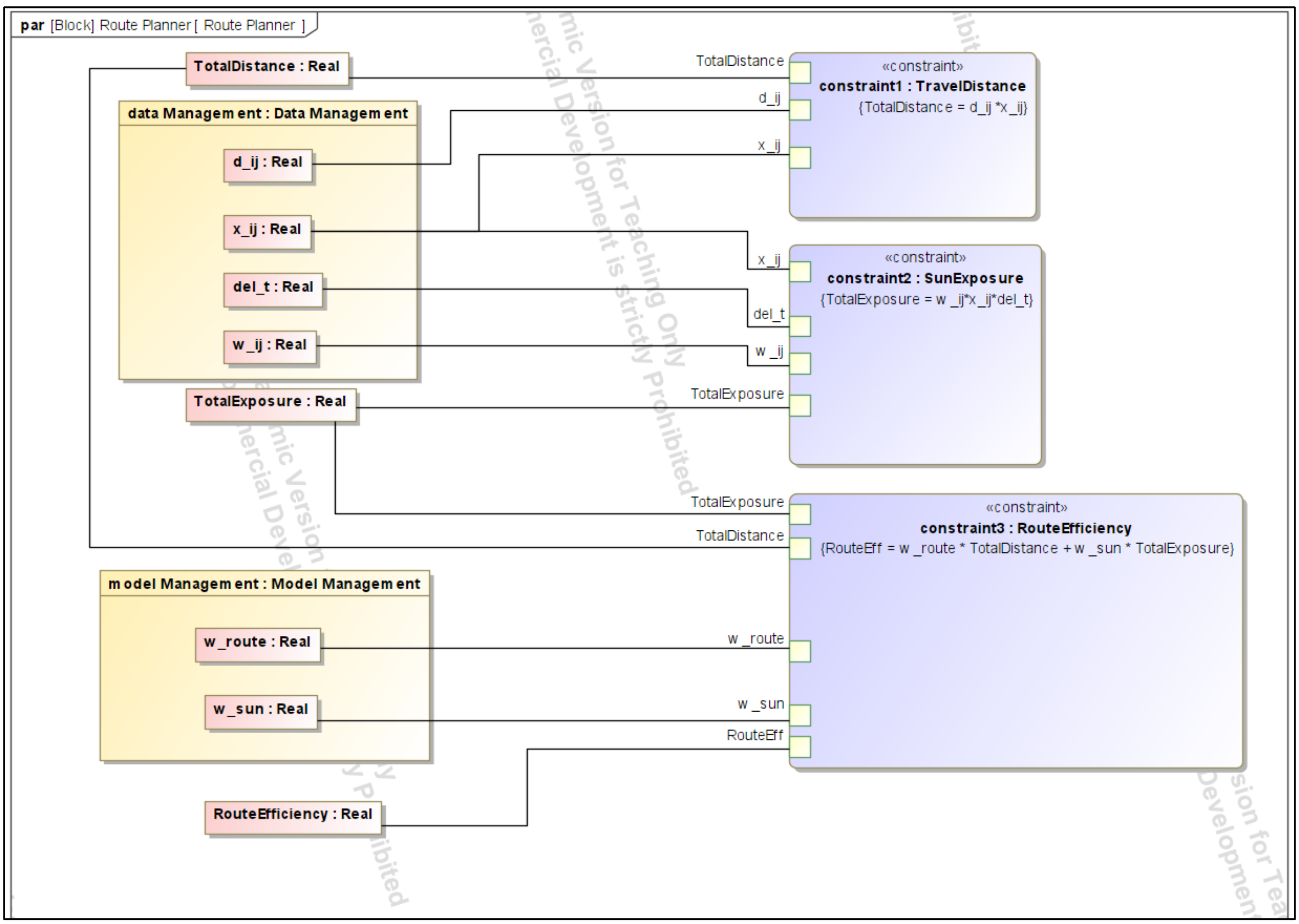


Figure 18 : Parametric Diagram

It helped me better define what should be the recommended weightage assigned to route distance needed to travel (w_route) and sun exposure (w_sun) experienced to get more comfortable experience to driver. It helped me finalize the optimum value of weightage I should be assigning to time and sun exposure which helped me to make standardized test for the functional behavior of the system.

32. Timeline Update or Extension

Timeline Update or Extension is a systems engineering tool used to organize project activities after all requirements are finalized and the systems functional and structural architecture is defined. It divides the work into manageable tasks. Further it assigns durations, and helps visualize the sequence of actions required for project completion. This tool ensures that all tasks are planned to meet project milestones and deadlines.

Using this tool, we were able to identify the action items necessary for project completion, understand the time required for each task, and track expected deliverables. As shown in figure,

Timeline					
Task		**Description**	**Start**	**End**	**Deliverables**
Phase 1 : Reserch and Requirements					
1.1	Literature review	Literature review	1-Aug	20-Aug	- Structured research document including equations, assumptions for modelling, solar trajec
1.2	Requirement Analysis	Define system requirements and success criteria	15-Aug	31-Aug	- System requirement document - Test criteria matrix
1.3	SysML Architecture	Develop SysML models for systems arcitecture	20-Aug	5-Sep	- SysMLarchietecture diagrams
Phase 2 : Shadow model development					
2.1	Shadow algorithm design	Develop mathematical model for solar position and shadow projection	10-Aug	30-Aug	- Solar position equation - Shadow projection formulas
2.2	Shadow model implementation	Development of shadow algorithm in C#	1-Sep	20-Sep	- Working shadow calculation code in unity interface - Unit test
2.3	Test 1	Testing of shadow model on arbitrary geometry	15-Sep	30-Sep	- Test results for cubes, cylinders -Accuracy validation report
Phase 3 : 3D New york city model acquisition					
3.1	3D model sourcing	Acquire NYC 3D building and road network model	20-Aug	10-Sep	- Raw 3D city data
3.2	Convert data in unity compatible format	Clean and import model in parts in unity	10-Sep	30-Sep	- Cleaned 3D model in parts (.fbx or .obj)
3.3	3D model import in unity	Assemble and import data in unity	1-Oct	20-Oct	- combined NYC building model along with stictched road in unity package format
3.4	Test 2	Geometry validation	15-Oct	31-Oct	- Validation report
Phase 4 : Integrated simulation environment					
4.1	Shadow model integartion	Integrate shadow algorithm with 3D Unity model	1-Nov	25-Nov	- Integrated C# algorithm for shadow calculation with 3D NYC model
4.2	Simulation Control	Develop method for time, date location control	10-Nov	30-Nov	- Method for simulation control - Method for result collection
4.3	Test 3	System Validation	25-Nov	8-Dec	- Test results for different conditions - Shadow accuracy validation
4.4	Semester 1 documentation	Compile results, lesson learned, semester report	9-Dec	15-Dec	- Semester 1 work report - Lesson learned document
Phase 5 : Digital twin data generation					
5.1	Route Network Definition	Define pedestrian route network (nodes, edges) on NYC model	20-Jan	3-Feb	- Network topology document - Coordinate mapping
5.2	Simulation Parameter Design	Define scenarios: weather conditions, time periods, seasons	27-Jan	12-Feb	- Weather condition definition - Scenario matrix - Sampling strategy
5.3	Data Collection	Run simulations to generate shadow exposure data for routes	8-Feb	28-Feb	- Shadow exposure dataset - Route metadata
5.4	Test 4	Data Quality Validation	25-Feb	8-Mar	- Data quality report - Cleaned dataset - Data validation test report

Table 14 : Timeline

It also helped us better define and organize tasks required to achieve the project goals and distribute responsibilities across the team efficiently. Beyond planning, the timeline serves as a tracking mechanism to monitor progress, adjust schedules as needed, and ensure that the project stays on track toward successful completion.

Closing Remarks

Through the application of systems engineering methodologies, the solar route planner project was able to address key stakeholders, customer pain points while balancing performance requirements and regulatory guidance. At each phase of design and development tools provided critical insight into the system behavior and constraints, interactions and potential failure , hence enabling creation of more robust and reliable system.

The project demonstrated the value of structured and methodological approach in systems engineering resulting in solution which is technically sound and user centric. This report can build strong foundation for future enhancements and potential real world implementations.