项目文档

# Functional Requirement

1. Functional Requirements  
  
1.1 Vehicle Registration Function   
 Function ID: FR-01   
 Description: Administrators can register new vehicles, update existing registrations, or remove outdated vehicle records. This function ensures data integrity by validating input and checking for duplicate entries in the Data Source.   
 Input: Vehicle details (e.g., make, model, license plate, vehicle identification number (VIN), and registration status).   
 Output: Updated or newly added vehicle data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect the new or modified vehicle location.   
  
1.2 Manage Vehicle Data Function   
 Function ID: FR-02   
 Description: Administrators can add, update, or remove vehicle data from the system. The function ensures that all changes are validated and stored in the Data Cache and Data Archive.   
 Input: Vehicle data (e.g., VIN, license plate, location, speed, or registration status) and an action (Add, Update, Remove).   
 Output: Updated vehicle data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the current state of vehicle data.   
  
1.3 Read Vehicle Information Function   
 Function ID: FR-03   
 Description: Administrators can retrieve and view detailed information about a specific vehicle, including its location and status. The retrieved data is stored in the Data Cache for quick access.   
 Input: Vehicle identifier (e.g., VIN or license plate).   
 Output: Retrieved vehicle data displayed in the User Interface. The Map Display and Icon Layer are updated to show the vehicle's location and current status.   
  
1.4 Delete Vehicle Record Function   
 Function ID: FR-04   
 Description: Administrators can delete a vehicle record from the system. The function ensures that the deletion is validated and that the Data Cache and Data Archive are updated accordingly.   
 Input: Vehicle identifier (e.g., VIN or license plate).   
 Output: Deleted vehicle data removed from the Data Cache and Data Archive. The Map Display and Icon Layer are updated to remove the deleted vehicle’s representation.   
  
1.5 Traffic Data Collection Function   
 Function ID: FR-05   
 Description: The system collects real-time traffic data from various Data Sources and stores it in the Data Cache and Data Archive for immediate and long-term use.   
 Input: Traffic data (e.g., vehicle speed, traffic volume, and occupancy) from external Data Sources.   
 Output: Collected and validated traffic data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect current traffic conditions.   
  
1.6 Analyze Traffic Metrics Function   
 Function ID: FR-06   
 Description: The system analyzes traffic data to derive key metrics such as congestion levels, travel times, and queue lengths using predefined Algorithms.   
 Input: Traffic data (e.g., vehicle speed, traffic volume, and occupancy) and analysis parameters (e.g., geographic area and time frame).   
 Output: Traffic metrics (e.g., congestion level, travel time, queue length) stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect the analysis results.   
  
1.7 Update Traffic Status Function   
 Function ID: FR-07   
 Description: The system updates the traffic status based on the latest data collected from the Data Source. This function ensures that the updated status is stored and visualized on the Map Display.   
 Input: Traffic data (e.g., vehicle speed, congestion level, and incident status) and update parameters (e.g., geographic area and time frame).   
 Output: Updated traffic status stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the current traffic conditions.   
  
1.8 View Traffic Congestion Function   
 Function ID: FR-08   
 Description: The system retrieves and displays current traffic congestion levels, based on vehicle speed and density, using the Michigan Geographic Framework for spatial representation.   
 Input: Geographic area and time frame for congestion analysis.   
 Output: Traffic congestion level visualization on the Map Display. The results are stored in the Data Cache for quick access in future sessions.   
  
1.9 Weather Data Integration Function   
 Function ID: FR-09   
 Description: The system integrates real-time weather data with traffic and road condition data to improve incident detection and road surface condition inference.   
 Input: Weather data (e.g., temperature, precipitation, visibility) and integration parameters (e.g., geographic area and time frame).   
 Output: Integrated weather and traffic data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect weather-related traffic impacts.   
  
1.10 Process Weather Observations Function   
 Function ID: FR-10   
 Description: The system processes and normalizes weather observation data, aligning it with traffic and road condition data for analysis and visualization.   
 Input: Weather observation data (e.g., temperature, wind speed, and precipitation) and processing parameters (e.g., geographic area and time frame).   
 Output: Processed weather data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect the weather conditions.   
  
1.11 Modify Weather Record Function   
 Function ID: FR-11   
 Description: Administrators can modify weather records in the system, including updating temperature, precipitation, or visibility data. The modified data is validated and stored for use in analysis.   
 Input: Weather record identifier (e.g., location and timestamp) and updated weather data (e.g., temperature, precipitation, visibility).   
 Output: Updated weather data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the new weather conditions.   
  
1.12 Road Condition Monitoring Function   
 Function ID: FR-12   
 Description: The system monitors road conditions and visualizes them on the Map Display using the Michigan Geographic Framework.   
 Input: Road condition data (e.g., surface condition, visibility, and friction) and monitoring parameters (e.g., geographic area and time frame).   
 Output: Road condition visualization on the Map Display. The processed data is stored in the Data Cache and Data Archive.   
  
1.13 Infer Road Surface Status Function   
 Function ID: FR-13   
 Description: The system infers the current road surface status by analyzing Road Condition Data, Traffic Data, and Weather Data. This function is used to determine conditions such as icy, wet, or dry road surfaces.   
 Input: Road condition data, traffic data, and weather data.   
 Output: Inferred road surface status stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect the inferred road conditions.   
  
1.14 Update Road Condition Function   
 Function ID: FR-14   
 Description: Administrators can update road condition records in the system, including surface condition, visibility, or road closure status. The updated data is validated and stored for analysis and visualization.   
 Input: Road segment identifier (e.g., road name or segment ID) and updated road condition data (e.g., surface condition, visibility, or closure status).   
 Output: Updated road condition data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the changes.   
  
1.15 Incident Detection Function   
 Function ID: FR-15   
 Description: The system detects traffic incidents such as accidents, breakdowns, or road closures by analyzing Traffic Data, Road Condition Data, and Weather Data.   
 Input: Traffic data, road condition data, and weather data.   
 Output: Detected incident data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect incident locations.   
  
1.16 Manage Incident Details Function   
 Function ID: FR-16   
 Description: Administrators can add, update, or remove incident details in the system. The function ensures that changes are validated and stored for use in alerts and visualization.   
 Input: Incident details (e.g., location, type, severity, and time of occurrence) and an action (Add, Update, Remove).   
 Output: Updated or newly added incident data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the new or modified incident status.   
  
1.17 Delete Incident Record Function   
 Function ID: FR-17   
 Description: Administrators can delete an incident record from the system. The function ensures that the deletion is validated and that the Data Cache and Data Archive are updated accordingly.   
 Input: Incident identifier (e.g., location and timestamp).   
 Output: Deleted incident data removed from the Data Cache and Data Archive. The Map Display and Icon Layer are updated to remove the incident’s visual representation.   
  
1.18 View Incident Information Function   
 Function ID: FR-18   
 Description: Administrators can retrieve and view detailed information about a specific incident, including its location, type, severity, and status. The retrieved data is stored in the Data Cache for quick access.   
 Input: Incident identifier (e.g., location, type, or timestamp).   
 Output: Retrieved incident data displayed in the User Interface. The Map Display and Icon Layer are updated to reflect the incident’s location and details.   
  
1.19 Data Caching Management Function   
 Function ID: FR-19   
 Description: Administrators can manage the Data Cache by adding, updating, or removing data. The function ensures that caching strategies are optimized and that the Data Cache and Data Archive are synchronized.   
 Input: Data type (e.g., vehicle, road, or incident data) and an action (Add, Update, Remove).   
 Output: Updated Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect any changes in data representation.   
  
1.20 Archive Historical Data Function   
 Function ID: FR-20   
 Description: Administrators can archive historical data from the Data Cache to the Data Archive for long-term storage. The function ensures that the data is validated and formatted for archival.   
 Input: Data type (e.g., traffic, weather, or incident data) and a time range for archiving.   
 Output: Archived data stored in the Data Archive. The Data Cache is cleared or updated as needed. The Map Display and Icon Layer remain unchanged unless affected by the data.   
  
1.21 Retrieve Archived Data Function   
 Function ID: FR-21   
 Description: Administrators can retrieve historical data from the Data Archive and store it in the Data Cache for quick access and visualization.   
 Input: Data type (e.g., vehicle, road condition, or incident data) and an identifier (e.g., vehicle ID or incident ID).   
 Output: Retrieved data stored in the Data Cache. The Map Display and Icon Layer are updated to reflect the archived data as needed.   
  
1.22 Data Quality Validation Function   
 Function ID: FR-22   
 Description: The system validates incoming data (e.g., traffic, weather, or road condition data) to ensure consistency, completeness, and accuracy. Invalid or corrupted data is flagged or corrected.   
 Input: Data type (e.g., traffic data, weather data, or road condition data) and a time range for validation.   
 Output: Valid data stored in the Data Cache and Data Archive. Invalid data is flagged and reported to the Administrator. The Map Display and Icon Layer are updated if the validation affects the data visualization.   
  
1.23 Modify Data Quality Rule Function   
 Function ID: FR-23   
 Description: Administrators can modify existing data quality rules to adapt to changing data standards or system requirements. The function ensures that the new rule is validated and stored.   
 Input: Rule identifier (e.g., rule name or ID) and updated rule parameters (e.g., validation thresholds, data formats, or error tolerances).   
 Output: Updated data quality rule stored in the Data Cache and Data Archive. The Algorithm is updated to reflect the new rule.   
  
1.24 Administrator Login Function   
 Function ID: FR-24   
 Description: Administrators can log in to the system using valid credentials. The function ensures that login attempts are validated and that session information is stored in the Data Cache.   
 Input: Username and password.   
 Output: Session information stored in the Data Cache. The Administrator is granted access to the system and receives a confirmation Alert.   
  
1.25 Access User Interface Function   
 Function ID: FR-25   
 Description: The system provides access to the User Interface after successful login. The function ensures that the session is recorded and that the Administrator is informed of the access.   
 Input: Valid login session.   
 Output: Access granted to the User Interface. Session details are stored in the Data Cache. An Alert is generated to confirm the access.   
  
1.26 Browse Traffic Information Function   
 Function ID: FR-26   
 Description: Administrators can browse and view real-time or historical traffic information. The function ensures that the data is retrieved, validated, and visualized on the Map Display.   
 Input: Geographic area and time frame for traffic data.   
 Output: Traffic information retrieved and displayed in the User Interface. The Map Display and Icon Layer are updated to reflect current traffic conditions.   
  
1.27 Display Map Overview Function   
 Function ID: FR-27   
 Description: The system displays a comprehensive map overview that integrates traffic, road condition, and weather data. The function ensures that the map is updated and that the data is formatted for visualization.   
 Input: Traffic data, road condition data, and weather data.   
 Output: Updated Map Display with visual representation of traffic, road, and weather conditions. The Data Cache is updated with the latest data for quick reference.   
  
1.28 Toggle Icon Layer Function   
 Function ID: FR-28   
 Description: Administrators can toggle the visibility of the Icon Layer to display or hide specific data visualizations on the Map Display.   
 Input: Action (Enable or Disable) and optional data layer to toggle.   
 Output: Updated visibility status of the Icon Layer stored in the Data Cache. The Map Display is refreshed to reflect the new visibility state.   
  
1.29 Declutter Map View Function   
 Function ID: FR-29   
 Description: Administrators can declutter the Map Display by hiding or prioritizing specific data layers to improve visual clarity.   
 Input: Data layers to hide or show.   
 Output: Updated Map Display with decluttered view. The new decluttering settings are stored in the Data Cache for future use.   
  
1.30 Add Data Source Function   
 Function ID: FR-30   
 Description: Administrators can add a new Data Source to the system, including its name, type, connection parameters, and data format. The function ensures that the Data Source is validated and stored.   
 Input: Data Source details (e.g., name, type, connection parameters, and data format).   
 Output: New Data Source metadata stored in the Data Cache and Data Archive. An Alert is generated to confirm the addition.   
  
1.31 Configure Algorithm Function   
 Function ID: FR-31   
 Description: Administrators can configure the parameters of an Algorithm to customize its behavior for data processing and analysis.   
 Input: Algorithm identifier (e.g., name or ID) and new configuration parameters (e.g., thresholds, weights, or data formats).   
 Output: Updated Algorithm configuration stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed if the configuration affects spatial data visualization.   
  
1.32 Update Algorithm Function   
 Function ID: FR-32   
 Description: Administrators can update an Algorithm with new settings or versions. The function ensures that the update is validated and that the Algorithm is synchronized with the Data Cache and Data Archive.   
 Input: Algorithm identifier (e.g., name or ID) and updated algorithm parameters or version.   
 Output: Updated Algorithm stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed if the update affects data visualization.   
  
1.33 Delete Algorithm Function   
 Function ID: FR-33   
 Description: Administrators can delete an Algorithm from the system. The function ensures that the deletion is validated and that the Data Cache and Data Archive are updated.   
 Input: Algorithm identifier (e.g., name or version).   
 Output: Algorithm removed from the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed if the deletion affects data visualization.   
  
1.34 Format Data Output Function   
 Function ID: FR-34   
 Description: The system formats data into specified Output Formats (e.g., SAE J2354, TMDD, or custom format) for external use.   
 Input: Data type (e.g., traffic, weather, or incident data) and Output Format (e.g., SAE J2354 or TMDD).   
 Output: Formatted data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated if the formatting affects spatial data.   
  
1.35 Modify Output Format Function   
 Function ID: FR-35   
 Description: Administrators can modify the Output Format configuration to align data with new or updated standards.   
 Input: Output Format identifier (e.g., format name or ID) and new format parameters (e.g., mapping rules or field definitions).   
 Output: Updated Output Format configuration stored in the Data Cache and Data Archive. The Algorithm is updated to reflect the new format.   
  
1.36 Publish Data to SAE J2354 Function   
 Function ID: FR-36   
 Description: The system publishes data in SAE J2354 format to external systems.   
 Input: Data type (e.g., traffic, road condition, or weather data) to be published.   
 Output: Data in SAE J2354 format published to the SAE J2354 interface. The Data Cache and Data Archive are updated to reflect the publication.   
  
1.37 Publish Data to TMDD Function   
 Function ID: FR-37   
 Description: The system publishes data in TMDD format to external systems.   
 Input: Data type (e.g., traffic, road condition, or weather data) to be published.   
 Output: Data in TMDD format published to the TMDD interface. The Data Cache and Data Archive are updated to reflect the publication.   
  
1.38 Generate Alert Function   
 Function ID: FR-38   
 Description: The system generates alerts based on specified criteria, such as traffic congestion, weather events, or asset conditions.   
 Input: Alert criteria (e.g., type, severity, geographic area, or time frame).   
 Output: Generated Alert stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated if the alert involves spatial data.   
  
1.39 Modify Alert Settings Function   
 Function ID: FR-39   
 Description: Administrators can modify alert settings, such as severity thresholds, notification frequency, or data types that trigger alerts.   
 Input: Alert identifier (e.g., name or ID) and new alert parameters (e.g., severity thresholds or notification channels).   
 Output: Updated alert settings stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed if the alert settings affect visualization.   
  
1.40 View Alert History Function   
 Function ID: FR-40   
 Description: Administrators can view historical alerts based on time range, severity, or alert type. The function ensures that the data is retrieved and displayed.   
 Input: Time range or filter criteria for alert history.   
 Output: Alert history retrieved and displayed in the User Interface. The Data Cache is updated with the retrieved data. The Map Display and Icon Layer are updated if the alerts involve spatial information.   
  
1.41 View AlgorithmExecutionLog Function   
 Function ID: FR-41   
 Description: Administrators can view the Algorithm Execution Log to track the execution history of Algorithms, including input parameters, execution time, and status.   
 Input: Time range, algorithm name, or execution status.   
 Output: Algorithm Execution Log retrieved and displayed in the User Interface. The Data Cache is updated with the retrieved log entries. The Map Display is updated if the logs involve spatial data execution.   
  
1.42 Format DataFormatMapping Function   
 Function ID: FR-42   
 Description: Administrators can define or modify how data fields are mapped to standardized formats (e.g., SAE J2354 or TMDD).   
 Input: Data type (e.g., traffic, weather, or incident data) and mapping rules (e.g., field definitions or format specifications).   
 Output: Updated DataFormatMapping stored in the Data Cache and Data Archive. The Algorithm is updated to use the new mapping rules.   
  
1.43 View Incident History Function   
 Function ID: FR-43   
 Description: Administrators can retrieve and view historical incident records based on time range, geographic area, or incident type.   
 Input: Time range, geographic area, or incident type.   
 Output: Incident history retrieved and displayed in the User Interface. The Data Cache is updated with the retrieved data. The Map Display and Icon Layer are refreshed to show the locations of historical incidents.   
  
1.44 Manage DataRule Function   
 Function ID: FR-44   
 Description: Administrators can manage Data Rules, including adding, updating, or removing rules for data validation and consistency.   
 Input: DataRule identifier (e.g., rule name or ID) and new or modified rule parameters (e.g., validation thresholds or error tolerances).   
 Output: Updated DataRule stored in the Data Cache and Data Archive. The Algorithm is updated to use the new or modified rule in its processing.   
  
1.45 Configure Map Layer Setting Function   
 Function ID: FR-45   
 Description: Administrators can configure the visibility, styling, and priority of map layers in the Map Display.   
 Input: Layer settings (e.g., visibility status, color, or priority).   
 Output: Updated Map Layer Settings stored in the Data Cache. The Map Display and Icon Layer are refreshed to reflect the new settings.   
  
1.46 Michigan Geographic Framework Integration Function   
 Function ID: FR-46   
 Description: The system integrates data with the Michigan Geographic Framework to ensure accurate spatial representation on the Map Display.   
 Input: Data types (e.g., traffic, road condition, or weather data) to be integrated with geographic data.   
 Output: Data aligned with the Michigan Geographic Framework and stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect the geographic context.   
  
1.47 Modify Geo-referencing Settings Function   
 Function ID: FR-47   
 Description: Administrators can modify the geo-referencing parameters of the Michigan Geographic Framework, such as coordinate system or projection.   
 Input: Geo-referencing parameters (e.g., coordinate system, projection, or geographic scale).   
 Output: Updated geo-referencing settings stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the new configuration.   
  
1.48 Travel Demand Analysis Function   
 Function ID: FR-48   
 Description: The system analyzes traffic and weather data to calculate travel demand in specific geographic areas and time frames.   
 Input: Geographic area and time frame for analysis.   
 Output: Travel demand data stored in the Data Cache and Data Archive. The Map Display is updated to show the travel demand distribution.   
  
1.49 Update Travel Demand Model Function   
 Function ID: FR-49   
 Description: Administrators can update the travel demand model with new or modified parameters (e.g., population growth or traffic influence factors).   
 Input: Travel demand model parameters (e.g., population growth, travel patterns, or traffic influence factors).   
 Output: Updated travel demand model stored in the Data Cache and Data Archive. The Map Display is refreshed to reflect the new demand distribution.   
  
1.50 Asset Condition Monitoring Function   
 Function ID: FR-50   
 Description: The system monitors asset conditions, such as infrastructure status, and visualizes them on the Map Display using the Icon Layer.   
 Input: Asset type and geographic area for monitoring.   
 Output: Asset condition data stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated to reflect asset status.   
  
1.51 Update Asset Status Function   
 Function ID: FR-51   
 Description: Administrators can update the status of an asset (e.g., operational, under maintenance, or decommissioned) in the system.   
 Input: Asset identifier (e.g., asset ID or location) and updated status.   
 Output: Updated asset status stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the new asset condition.   
  
1.52 Manage AssetStatus Function   
 Function ID: FR-52   
 Description: Administrators can add, update, or remove asset status records. The function ensures that changes are validated and stored in the Data Cache and Data Archive.   
 Input: Asset identifier (e.g., asset ID or location) and an action (Add, Update, Remove).   
 Output: Updated or newly added asset status stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the new or modified status.   
  
1.53 Delete Icon Layer Function   
 Function ID: FR-53   
 Description: Administrators can delete an Icon Layer from the Map Display. The function ensures that the deletion is validated and that the Data Cache is updated.   
 Input: Icon Layer identifier (e.g., layer name or ID).   
 Output: Icon Layer removed from the Map Display. The Data Cache is updated to reflect the removal.   
  
1.54 Manage Traveler Record Function   
 Function ID: FR-54   
 Description: Administrators can manage traveler records, including adding, updating, or removing information. The function ensures that all changes are validated and stored in the Data Cache and Data Archive.   
 Input: Traveler identifier (e.g., name or ID) and an action (Add, Update, Remove).   
 Output: Updated or newly added traveler record stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed if the traveler is associated with location data.   
  
1.55 MI Drive Presentation Support Function   
 Function ID: FR-55   
 Description: The system supports the presentation of traveler information for MI Drive, using the selected Output Format and Presentation Method.   
 Input: Traveler information type (e.g., traffic conditions, road closures, or weather impacts).   
 Output: Traveler information formatted and displayed in the selected Output Format. The Map Display is updated to reflect the presentation.   
  
1.56 View Traveler Information Function   
 Function ID: FR-56   
 Description: Administrators can retrieve and view traveler information, such as traffic, road conditions, or weather impacts. The function ensures that the data is validated and displayed.   
 Input: Geographic area and time frame for the query.   
 Output: Traveler information retrieved and displayed in the User Interface. The Map Display and Icon Layer are refreshed to reflect the data.   
  
1.57 Update Traveler Notification Function   
 Function ID: FR-57   
 Description: Administrators can update traveler notifications, including message text, severity, or display duration. The function ensures that the update is validated and stored.   
 Input: Notification identifier (e.g., location or message ID) and updated notification details.   
 Output: Updated notification stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the new notification.   
  
1.58 Delete Traveler Record Function   
 Function ID: FR-58   
 Description: Administrators can delete a traveler record from the system. The function ensures that the deletion is validated and that the Data Cache and Data Archive are updated.   
 Input: Traveler identifier (e.g., user ID or traveler profile).   
 Output: Traveler data removed from the Data Cache and Data Archive. The Map Display and Icon Layer are updated to remove the traveler's representation.   
  
1.59 Manage Vehicle Registration Function   
 Function ID: FR-59   
 Description: Administrators can manage vehicle registration records, including adding, updating, or removing vehicle data. The function ensures that all changes are validated and stored in the Data Cache and Data Archive.   
 Input: Vehicle details (e.g., make, model, license plate, VIN) and an action (Add, Update, Remove).   
 Output: Updated or newly added vehicle registration stored in the Data Cache and Data Archive. The Map Display and Icon Layer are refreshed to reflect the changes.

# External Description

2. External Interfaces   
  
2.1 User Interface   
Description: The system provides a graphical User Interface (UI) for administrators to interact with and manage traffic, weather, and road condition data. The UI allows for data input, retrieval, visualization, and configuration.   
Interaction Method:   
- Administrators log in via the UI using valid credentials (FR-24).   
- The UI displays real-time and historical data, including traffic congestion, road conditions, weather impacts, and incident details (FR-08, FR-13, FR-15, FR-18, FR-26, FR-27, FR-28, FR-29).   
- Administrators can toggle the visibility of the Icon Layer and declutter the Map Display for better data visualization (FR-28, FR-29).   
- The UI also provides access to data management functions, such as adding, updating, or removing vehicle, incident, asset, or traveler records (FR-01, FR-04, FR-16, FR-17, FR-51, FR-58).   
- Alert settings, data quality rules, and algorithm configurations can be modified through the UI (FR-39, FR-44, FR-31).   
- The UI displays formatted data outputs and provides options for selecting output formats such as SAE J2354 or TMDD (FR-34, FR-35).   
- Traveler information, including traffic and weather conditions, can be viewed and managed through the UI (FR-56, FR-57).   
  
2.2 Hardware Interface   
Description: The system interacts with external hardware devices to collect real-time traffic and weather data. These devices may include sensors, cameras, radar systems, and other IoT-enabled infrastructure.   
Interaction Method:   
- Traffic data is collected from hardware devices such as vehicle sensors, traffic cameras, and radar systems. Inputs include vehicle speed, traffic volume, and occupancy (FR-05, FR-07, FR-48).   
- Weather data is collected from hardware devices such as weather stations, temperature sensors, and precipitation detectors. Inputs include temperature, wind speed, and visibility (FR-09, FR-10, FR-13).   
- Road condition data is collected from hardware devices that monitor road surface conditions, such as friction sensors and visibility meters. Inputs include surface condition, visibility, and friction (FR-12, FR-13).   
- Asset condition data is collected from hardware devices monitoring infrastructure, such as road signs, bridges, or tunnels. Inputs include asset status, maintenance needs, or decommissioning status (FR-50, FR-51).   
- These hardware devices communicate with the system via predefined communication protocols (e.g., MQTT, HTTP, or TCP/IP).   
  
2.3 Software Interface   
Description: The system interacts with various software components and databases to store, retrieve, and process data. These software interfaces include the Data Cache, Data Archive, Algorithm Engine, and external data formatting standards.   
Interaction Method:   
- Data Cache:   
 - Stores real-time data for quick access and visualization.   
 - Receives updates from functions such as vehicle registration (FR-01), traffic data collection (FR-05), and alert generation (FR-38).   
 - Provides data to the Map Display and Icon Layer for visualization (FR-03, FR-10, FR-13, FR-18, FR-22, FR-28, FR-29).   
  
- Data Archive:   
 - Stores historical data for long-term analysis and retrieval.   
 - Receives archived data from the Data Caching Management and Archive Historical Data functions (FR-19, FR-20).   
 - Provides data to the Retrieve Archived Data function (FR-21) and is updated with new data from various functional operations (FR-01, FR-05, FR-13, FR-16, etc.).   
  
- Algorithm Engine:   
 - Processes data using predefined algorithms to generate traffic metrics, infer road surface status, calculate travel demand, and detect incidents.   
 - Receives data from the Data Cache and Data Archive (FR-06, FR-13, FR-15, FR-48).   
 - Updates its configuration based on administrator inputs (FR-31, FR-32, FR-33).   
  
- Data Quality Rules:   
 - Stored in the Data Cache and Data Archive.   
 - Used to validate incoming data and ensure consistency, completeness, and accuracy (FR-22, FR-23).   
  
- Output Formats (e.g., SAE J2354, TMDD):   
 - Define the structure and format of data for external systems.   
 - Updated by administrators to align with new standards or data types (FR-34, FR-35, FR-42).   
 - Used to publish data in standardized formats (FR-36, FR-37).   
  
2.4 Communication Interface   
Description: The system communicates with external systems and services to exchange data and generate alerts. These interfaces ensure that data is shared in real-time and that administrators are informed of system events.   
Interaction Method:   
- Map Display:   
 - Visualizes traffic, road condition, weather, and asset data using the Michigan Geographic Framework.   
 - Updates in real-time based on data from the Data Cache and Data Archive (FR-08, FR-13, FR-15, FR-27, FR-46, FR-47).   
  
- Icon Layer:   
 - Provides visual indicators for vehicle, incident, asset, and traveler data on the Map Display.   
 - Updated dynamically based on data changes in the Data Cache and Data Archive (FR-03, FR-16, FR-18, FR-28, FR-50, FR-54).   
  
- Alert Notification:   
 - Sends alerts to administrators and external systems via predefined channels (e.g., email, SMS, or in-system notifications).   
 - Generated based on traffic, weather, or asset conditions (FR-38, FR-39, FR-40).   
 - Alert details are stored in the Data Cache and Data Archive for future reference (FR-38, FR-40).   
  
- Data Source Integration:   
 - Connects with external data sources (e.g., traffic sensors, weather APIs, road condition databases).   
 - Ensures data synchronization and validation between the system and these sources (FR-05, FR-09, FR-10, FR-12).   
  
- Algorithm Execution Log:   
 - Stores and retrieves logs of algorithm execution, including input parameters, execution time, and status.   
 - Used to monitor system performance and troubleshoot issues (FR-41).   
  
- Traveler Information Presentation:   
 - Supports data presentation for external platforms such as MI Drive.   
 - Formats and pushes traveler information (e.g., traffic conditions, road closures, weather impacts) using the selected Output Format (FR-55, FR-56).   
  
- Data Publication:   
 - Publishes data in SAE J2354 and TMDD formats to external systems for standardized data sharing.   
 - Updates the Data Cache and Data Archive with publication records (FR-36, FR-37).   
  
- Data Formatting and Mapping:   
 - Defines how data fields are mapped to standardized formats.   
 - Updated by administrators to reflect new or modified field definitions (FR-42).   
  
By clearly defining these external interfaces, the system ensures seamless data flow, real-time visualization, and effective communication with external systems and users.

# Use Case

Use Case Name: Vehicle Registration   
Use Case ID: UC-01   
Actors: Administrator, User Interface, Data Source, Map Display, Icon Layer, Data Quality Check, Travel Demand, TMDD Format, SAE J2354 Format, Alert, Michigan Geographic Framework, Asset Condition, Traveler Information, Road Condition Data, Traffic Metric Data, Traffic Data, Weather Data, Data Cache, Data Archive   
Preconditions: The system must be connected to the Data Source for vehicle information. The Administrator has access to the User Interface. The Map Display and Icon Layer are active and available for visualization.   
Postconditions: The vehicle is successfully registered in the system. The relevant data is stored in the Data Cache and Data Archive. The Map Display updates to reflect the new vehicle information.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Vehicle Registration" option.   
3. The system prompts the Administrator to input the vehicle details (e.g., make, model, license plate, and vehicle identification number).   
4. The Administrator enters the required vehicle information.   
5. The system validates the input data using the Data Quality Check module.   
6. The system connects to the Data Source to verify if the vehicle already exists in the database.   
7. If the vehicle is not already registered, the system proceeds to register the vehicle.   
8. The system stores the new vehicle data in the Data Cache and Data Archive.   
9. The Map Display updates to show the new vehicle's location using the Michigan Geographic Framework.   
10. An Icon Layer is added to the Map Display to represent the registered vehicle.   
11. The system generates an Alert if the registration is successful.   
12. The Administrator confirms the successful registration and closes the process.   
  
Alternative Flow:   
1. If the vehicle is already registered, the system displays an Alert indicating the duplication.   
2. The Administrator can choose to update the existing vehicle information or cancel the registration process.   
3. If the Administrator chooses to update, the system allows the input of new or modified vehicle details.   
4. The system validates the updated data and stores it in the Data Cache and Data Archive.   
5. The Map Display and Icon Layer are updated to reflect the changes.   
6. If the Administrator chooses to cancel, the system returns to the main menu without making any changes.  
  
Use Case Name: Manage Vehicle Data   
Use Case ID: UC-02   
Actors: Administrator, User Interface, Data Cache, Data Archive, Map Display, Icon Layer, Data Quality Check, Vehicle Data, Traffic Data, Weather Data, Asset Condition, Michigan Geographic Framework, Traveler Information, Presentation Method, Output Format, Algorithm, SAE J2354 Format, TMDD Format, Alert, Travel Demand, Data Source   
Preconditions: The system is connected to the Data Source for vehicle and related data. The Administrator has access to the User Interface. The Data Cache and Data Archive are ready to store or update vehicle data. The Map Display and Icon Layer are active for data visualization.   
Postconditions: The vehicle data is successfully managed (added, updated, or removed). The Data Cache and Data Archive are updated accordingly. The Map Display reflects the current vehicle data status.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Manage Vehicle Data" option.   
3. The system displays a list of available actions: "Add Vehicle Data," "Update Vehicle Data," or "Remove Vehicle Data."   
4. The Administrator selects "Add Vehicle Data."   
5. The system prompts the Administrator to input the vehicle details (e.g., make, model, license plate, and vehicle identification number).   
6. The Administrator enters the required vehicle data.   
7. The system validates the input data using the Data Quality Check module.   
8. The system connects to the Data Source to verify if the vehicle already exists in the database.   
9. If the vehicle is not already registered, the system proceeds to store the new data.   
10. The system saves the new vehicle data in the Data Cache and Data Archive.   
11. The Map Display updates to show the new vehicle's location using the Michigan Geographic Framework.   
12. An Icon Layer is added to the Map Display to represent the new vehicle.   
13. The system generates an Alert confirming the successful addition of the vehicle data.   
14. The Administrator confirms the action and closes the process.   
  
Alternative Flow:   
1. If the vehicle is already registered and the Administrator selects "Add Vehicle Data," the system displays an Alert indicating the duplication.   
2. The Administrator can choose to update the existing vehicle data or cancel the operation.   
3. If the Administrator chooses to update, the system allows the input of new or modified vehicle details.   
4. The system validates the updated data and stores it in the Data Cache and Data Archive.   
5. The Map Display and Icon Layer are updated to reflect the changes.   
6. If the Administrator chooses to cancel, the system returns to the main menu without making any changes.   
7. If the Administrator selects "Remove Vehicle Data," the system prompts for the vehicle identification number.   
8. The Administrator enters the vehicle identification number.   
9. The system verifies the existence of the vehicle in the database and removes the associated data.   
10. The Data Cache and Data Archive are updated to reflect the removal.   
11. The Map Display and Icon Layer are updated to remove the vehicle representation.   
12. The system generates an Alert confirming the successful removal of the vehicle data.  
  
Use Case Name: Read Vehicle Information   
Use Case ID: UC-03   
Actors: Administrator, User Interface, Data Source, Data Cache, Map Display, Icon Layer, Data Quality Check, Vehicle Data, Michigan Geographic Framework, Alert   
Preconditions: The system is connected to the Data Source for vehicle-related data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for visualization. The Data Cache is available for temporary storage.   
Postconditions: The vehicle information is retrieved and displayed. The Map Display updates to show the vehicle's location. The Data Cache contains the retrieved data for quick access.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Read Vehicle Information" option.   
3. The system prompts the Administrator to input the vehicle identification number (VIN) or license plate.   
4. The Administrator enters the required vehicle identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Source to retrieve the corresponding vehicle data.   
7. The system fetches the vehicle's location data and other relevant information.   
8. The retrieved data is stored in the Data Cache for quick reference.   
9. The Map Display is updated to show the vehicle's location using the Michigan Geographic Framework.   
10. An Icon Layer is added to the Map Display to represent the vehicle's current status.   
11. The system generates an Alert confirming the successful retrieval of the vehicle information.   
12. The Administrator reviews the displayed data and closes the process.   
  
Alternative Flow:   
1. If the vehicle identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and data retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying data.  
  
Use Case Name: Delete Vehicle Record   
Use Case ID: UC-04   
Actors: Administrator, User Interface, Data Cache, Data Archive, Map Display, Icon Layer, Data Source, Data Quality Check, Michigan Geographic Framework, Alert   
Preconditions: The system is connected to the Data Source for vehicle data. The Administrator has access to the User Interface. The Data Cache and Data Archive contain the vehicle record to be deleted. The Map Display and Icon Layer are active and reflect the current vehicle data.   
Postconditions: The vehicle record is successfully deleted from the system. The Data Cache and Data Archive are updated. The Map Display and Icon Layer are updated to remove the deleted vehicle's representation. An Alert is generated to confirm the deletion.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Delete Vehicle Record" option.   
3. The system prompts the Administrator to input the vehicle identification number (VIN) or license plate.   
4. The Administrator enters the required vehicle identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Source to confirm the existence of the vehicle record.   
7. If the record exists, the system proceeds to delete it.   
8. The system removes the vehicle data from the Data Cache and Data Archive.   
9. The Map Display is updated to remove the vehicle's location using the Michigan Geographic Framework.   
10. The Icon Layer is updated to remove the vehicle's representation.   
11. The system generates an Alert confirming the successful deletion of the vehicle record.   
12. The Administrator confirms the deletion and closes the process.   
  
Alternative Flow:   
1. If the vehicle identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and data deletion steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.  
  
Use Case Name: Traffic Data Collection   
Use Case ID: UC-05   
Actors: Administrator, User Interface, Data Source, Data Cache, Data Archive, Traffic Metric Data, Weather Data, Road Condition Data, Michigan Geographic Framework, Map Display, Icon Layer, Data Quality Check, Alert, Algorithm, Output Format, Presentation Method, Traveler Information, Travel Demand, SAE J2354 Format, TMDD Format   
Preconditions: The system is connected to the Data Source for real-time traffic, weather, and road condition data. The Administrator has access to the User Interface. The Data Cache and Data Archive are available for data storage. The Map Display and Icon Layer are active for visualization.   
Postconditions: Traffic data is successfully collected, processed, and stored. The Data Cache and Data Archive are updated with the latest data. The Map Display reflects the current traffic conditions. An Alert is generated to notify the Administrator of the successful collection.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Traffic Data Collection" option.   
3. The system initiates data collection from the Data Source, including Traffic Metric Data, Weather Data, and Road Condition Data.   
4. The system applies the Algorithm to process and normalize the collected data according to the SAE J2354 and TMDD formats.   
5. The system performs a Data Quality Check to ensure data accuracy and consistency.   
6. Validated data is stored in the Data Cache for immediate access and in the Data Archive for long-term storage.   
7. The Map Display is updated using the Michigan Geographic Framework to show the current traffic conditions.   
8. The Icon Layer is modified to reflect traffic incidents, congestion, or road closures.   
9. The system generates an Alert confirming the successful collection and display of traffic data.   
10. The Administrator reviews the data and closes the process.   
  
Alternative Flow:   
1. If the Data Source is unavailable or the data is invalid, the system displays an Alert indicating the failure.   
2. The Administrator can choose to retry the data collection or cancel the operation.   
3. If the Administrator selects to retry, the system reconnects to the Data Source and repeats the data collection process.   
4. If the Administrator cancels, the system returns to the main menu without updating the data.  
  
Use Case Name: Analyze Traffic Metrics   
Use Case ID: UC-06   
Actors: Administrator, User Interface, Traffic Metric Data, Algorithm, Output Format, Presentation Method, Map Display, Icon Layer, Data Cache, Data Archive, Michigan Geographic Framework, Alert, Traveler Information, Travel Demand, Data Source, Data Quality Check   
Preconditions: The system is connected to the Data Source for real-time and historical traffic data. The Administrator has access to the User Interface. The Data Cache and Data Archive are ready for data retrieval and processing. The Map Display and Icon Layer are active for visualization. The Algorithm is configured to analyze traffic metrics.   
Postconditions: Traffic metrics are analyzed and presented in the desired Output Format. The results are stored in the Data Cache and Data Archive. The Map Display is updated to reflect traffic conditions. The Administrator is notified of the analysis completion via an Alert.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Analyze Traffic Metrics" option.   
3. The system prompts the Administrator to specify the time range and geographic area for analysis.   
4. The Administrator provides the required parameters.   
5. The system retrieves relevant Traffic Metric Data from the Data Source.   
6. The system performs a Data Quality Check on the retrieved data.   
7. The Algorithm processes the data to calculate traffic metrics such as congestion levels, average speed, and incident frequency.   
8. The system formats the results using the selected Output Format and Presentation Method.   
9. The Map Display is updated to show traffic patterns and incidents using the Michigan Geographic Framework.   
10. The Icon Layer is adjusted to reflect changes in traffic conditions.   
11. The system stores the analysis results in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator that the analysis is complete.   
13. The Administrator reviews the analysis results and closes the process.   
  
Alternative Flow:   
1. If the specified geographic area or time range is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the analysis.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and analysis steps.   
4. If the Administrator cancels, the system returns to the main menu without performing the analysis.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Update Traffic Status   
Use Case ID: UC-07   
Actors: Administrator, User Interface, Traffic Data, Algorithm, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Alert, Output Format, Presentation Method, Traveler Information, Data Cache, Data Archive   
Preconditions: The system is connected to the Data Source for real-time traffic updates. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for visualization. The Algorithm is configured to process and analyze traffic data. The Data Cache and Data Archive are available for storing updated traffic status.   
Postconditions: The traffic status is updated in the system. The Data Cache and Data Archive are updated with the latest traffic data. The Map Display reflects the current traffic conditions. An Alert is generated to confirm the update.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Update Traffic Status" option.   
3. The system prompts the Administrator to input the geographic area and time frame for the update.   
4. The Administrator provides the required parameters.   
5. The system retrieves the latest Traffic Data from the Data Source.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The Algorithm processes the data to update traffic metrics such as congestion, delays, and incidents.   
8. The system formats the updated traffic information using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show the latest traffic conditions.   
10. The Icon Layer is adjusted to reflect changes in traffic incidents or road conditions.   
11. The system stores the updated traffic status in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator of the successful update.   
13. The Administrator reviews the updated status and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the update.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and processing steps.   
4. If the Administrator cancels, the system returns to the main menu without updating the traffic status.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: View Traffic Congestion   
Use Case ID: UC-08   
Actors: Administrator, User Interface, Traffic Metric Data, Map Display, Icon Layer, Data Source, Data Quality Check, Michigan Geographic Framework, Alert, Traveler Information, Presentation Method, Output Format, Algorithm   
Preconditions: The system is connected to the Data Source for real-time traffic data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active and configured to visualize traffic conditions. The Algorithm is available to analyze and classify congestion levels.   
Postconditions: The current traffic congestion level is displayed to the Administrator. The Map Display and Icon Layer are updated to reflect congestion areas. The results are stored in the Data Cache. An Alert is generated to confirm the successful display of traffic congestion.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "View Traffic Congestion" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for the congestion analysis.   
4. The Administrator provides the required parameters.   
5. The system retrieves Traffic Metric Data from the Data Source based on the specified area and time.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The Algorithm processes the data to calculate congestion levels using relevant metrics such as vehicle speed and traffic density.   
8. The system formats the congestion data using the selected Output Format and Presentation Method.   
9. The Map Display is updated to show the current traffic congestion using the Michigan Geographic Framework.   
10. The Icon Layer is adjusted to highlight areas of high, medium, and low congestion.   
11. The system stores the congestion data in the Data Cache for quick access.   
12. The system generates an Alert to inform the Administrator that the traffic congestion visualization is complete.   
13. The Administrator reviews the congestion map and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and analysis steps.   
4. If the Administrator cancels, the system returns to the main menu without displaying traffic congestion.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Weather Data Integration   
Use Case ID: UC-09   
Actors: Administrator, User Interface, Weather Data, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Alert, Algorithm, Output Format, Presentation Method, Traveler Information, Traffic Metric Data, Road Condition Data, Traffic Data, Incident, Travel Demand   
Preconditions: The system is connected to the Data Source for real-time and historical weather data. The Administrator has access to the User Interface. The Data Cache and Data Archive are available for storing weather-related data. The Map Display and Icon Layer are active for visualization. The Algorithm is configured to process and integrate weather data with other traffic-related data.   
Postconditions: Weather data is successfully integrated into the system. The Data Cache and Data Archive are updated with the latest weather data. The Map Display reflects current weather conditions. An Alert is generated to confirm the integration.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Weather Data Integration" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for the weather data.   
4. The Administrator provides the required parameters.   
5. The system retrieves Weather Data from the Data Source based on the specified area and time.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The Algorithm processes the weather data and integrates it with Traffic Metric Data and Road Condition Data.   
8. The system formats the integrated data using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show the impact of weather conditions on traffic.   
10. The Icon Layer is adjusted to reflect weather-related incidents such as flooding, snow, or fog.   
11. The system stores the integrated weather data in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator that the weather data integration is complete.   
13. The Administrator reviews the updated map and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and integration steps.   
4. If the Administrator cancels, the system returns to the main menu without updating the data.   
5. If the Data Source is unreachable or returns corrupted weather data, the system generates an Alert and halts the process.  
  
Use Case Name: Process Weather Observations   
Use Case ID: UC-10   
Actors: Administrator, User Interface, Weather Data, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Algorithm, Alert, Traveler Information, Traffic Metric Data, Road Condition Data   
Preconditions: The system is connected to the Data Source for real-time weather observations. The Administrator has access to the User Interface. The Data Cache and Data Archive are available for storing processed weather data. The Map Display and Icon Layer are active for visualizing weather impacts. The Algorithm is configured to process and normalize weather observations.   
Postconditions: Weather observations are successfully processed and stored. The Data Cache and Data Archive are updated with the latest weather data. The Map Display reflects the current weather conditions. An Alert is generated to confirm the processing is complete.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Process Weather Observations" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for the weather data.   
4. The Administrator provides the required parameters.   
5. The system retrieves Weather Data from the Data Source based on the specified area and time.   
6. The system performs a Data Quality Check to ensure the accuracy and consistency of the observations.   
7. The Algorithm processes the weather data to normalize and categorize it (e.g., precipitation, temperature, visibility).   
8. The system integrates the processed weather data with Traffic Metric Data and Road Condition Data.   
9. The Map Display is updated using the Michigan Geographic Framework to show the current weather conditions.   
10. The Icon Layer is adjusted to highlight weather-related impacts such as reduced visibility or icy roads.   
11. The system stores the processed weather observations in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator that the weather observations have been successfully processed.   
13. The Administrator confirms the process and closes the operation.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and processing steps.   
4. If the Administrator cancels, the system returns to the main menu without processing the weather observations.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Modify Weather Record   
Use Case ID: UC-11   
Actors: Administrator, User Interface, Weather Data, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Algorithm, Alert, Traveler Information, Traffic Metric Data, Road Condition Data   
Preconditions: The system is connected to the Data Source for weather data. The Administrator has access to the User Interface. The Data Cache and Data Archive are available for storing and retrieving weather records. The Map Display and Icon Layer are active for visualizing weather conditions. The Algorithm is configured to process and validate weather data.   
Postconditions: The weather record is successfully modified in the system. The Data Cache and Data Archive are updated with the new weather data. The Map Display reflects the updated weather conditions. An Alert is generated to confirm the modification.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Modify Weather Record" option.   
3. The system prompts the Administrator to input the identifier of the weather record (e.g., location and timestamp).   
4. The Administrator enters the required identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Source to retrieve the existing weather record.   
7. The system displays the current weather data to the Administrator for review.   
8. The Administrator modifies the required fields of the weather record (e.g., temperature, precipitation, visibility).   
9. The system applies the Algorithm to normalize and validate the updated data.   
10. The system updates the weather record in the Data Cache and Data Archive.   
11. The Map Display is updated using the Michigan Geographic Framework to reflect the modified weather conditions.   
12. The Icon Layer is adjusted to represent the new weather status (e.g., rain, snow, fog).   
13. The system generates an Alert confirming the successful modification of the weather record.   
14. The Administrator reviews the updated data and closes the process.   
  
Alternative Flow:   
1. If the specified weather record identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.   
5. If the updated data fails validation, the system displays an Alert and prompts the Administrator to correct the data.   
6. The Administrator corrects the data and the system repeats the validation and modification steps.  
  
Use Case Name: Road Condition Monitoring   
Use Case ID: UC-12   
Actors: Administrator, User Interface, Road Condition Data, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Alert, Traveler Information, Traffic Data, Weather Data, Algorithm, Output Format, Presentation Method   
Preconditions: The system is connected to the Data Source for real-time and historical road condition data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active and configured to visualize road conditions. The Algorithm is available to process and classify road condition data.   
Postconditions: Road condition data is successfully retrieved, processed, and displayed. The Map Display reflects current road conditions. The Data Cache and Data Archive are updated with the latest road condition data. An Alert is generated to confirm the successful monitoring.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Road Condition Monitoring" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for road condition data.   
4. The Administrator provides the required parameters.   
5. The system retrieves Road Condition Data from the Data Source based on the specified area and time.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The Algorithm processes the data to identify road conditions such as potholes, flooding, or icy surfaces.   
8. The system formats the road condition data using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show the current road conditions.   
10. The Icon Layer is adjusted to reflect specific road condition incidents.   
11. The system stores the processed road condition data in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator that the road condition monitoring is complete.   
13. The Administrator reviews the road condition map and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and processing steps.   
4. If the Administrator cancels, the system returns to the main menu without displaying road conditions.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Infer Road Surface Status   
Use Case ID: UC-13   
Actors: Administrator, User Interface, Road Condition Data, Traffic Data, Weather Data, Algorithm, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Alert, Data Cache, Data Archive, Traveler Information   
Preconditions: The system is connected to the Data Source for Road Condition, Traffic, and Weather Data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for visualization. The Algorithm is configured to infer road surface status based on integrated data.   
Postconditions: Road surface status is successfully inferred and displayed. The Map Display and Icon Layer are updated to show the inferred conditions. The results are stored in the Data Cache and Data Archive. An Alert is generated to confirm the inference process is complete.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Infer Road Surface Status" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for the inference.   
4. The Administrator provides the required parameters.   
5. The system retrieves Road Condition Data, Traffic Data, and Weather Data from the Data Source.   
6. The system performs a Data Quality Check on the retrieved data to ensure accuracy and consistency.   
7. The Algorithm processes the data to infer the current road surface status (e.g., icy, wet, dry, flooded).   
8. The system formats the inferred road surface status using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show the inferred road surface conditions.   
10. The Icon Layer is adjusted to represent the road surface status visually.   
11. The system stores the inferred data in the Data Cache and Data Archive for future reference.   
12. The system generates an Alert to inform the Administrator that the inference is complete.   
13. The Administrator reviews the inferred road surface status and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and inference steps.   
4. If the Administrator cancels, the system returns to the main menu without performing the inference.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Update Road Condition   
Use Case ID: UC-14   
Actors: Administrator, User Interface, Road Condition Data, Data Source, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Alert, Weather Data, Traffic Metric Data, Data Cache, Data Archive, Algorithm, Traveler Information   
Preconditions: The system is connected to the Data Source for real-time and historical road condition data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for visualization. The Algorithm is configured to process and normalize road condition data.   
Postconditions: The road condition is successfully updated in the system. The Data Cache and Data Archive are updated with the latest road condition data. The Map Display reflects the current road condition. An Alert is generated to confirm the update.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Update Road Condition" option.   
3. The system prompts the Administrator to input the road segment identifier (e.g., location, road name, or segment ID).   
4. The Administrator enters the required road segment identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system retrieves the existing road condition data from the Data Source.   
7. The system displays the current road condition data to the Administrator for review.   
8. The Administrator modifies the required fields of the road condition data (e.g., surface condition, visibility, or road closure status).   
9. The system applies the Algorithm to normalize and validate the updated road condition data.   
10. The system updates the road condition data in the Data Cache and Data Archive.   
11. The Map Display is updated using the Michigan Geographic Framework to reflect the modified road condition.   
12. The Icon Layer is adjusted to visually represent the updated road status.   
13. The system generates an Alert confirming the successful update of the road condition.   
14. The Administrator reviews the updated data and closes the process.   
  
Alternative Flow:   
1. If the specified road segment identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.   
5. If the updated data fails validation, the system displays an Alert and prompts the Administrator to correct the data.   
6. The Administrator corrects the data and the system repeats the validation and update steps.  
  
Use Case Name: Incident Detection   
Use Case ID: UC-15   
Actors: Administrator, User Interface, Traffic Data, Road Condition Data, Weather Data, Algorithm, Data Source, Data Quality Check, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Traveler Information, Data Cache, Data Archive   
Preconditions: The system is connected to the Data Source for real-time traffic, road condition, and weather data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for visualizing incidents. The Algorithm is configured to detect incidents based on integrated data.   
Postconditions: Traffic incidents are successfully detected and displayed. The Map Display and Icon Layer are updated to reflect incident locations. The results are stored in the Data Cache and Data Archive. An Alert is generated to notify the Administrator of the incident detection.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Incident Detection" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for incident detection.   
4. The Administrator provides the required parameters.   
5. The system retrieves Traffic Data, Road Condition Data, and Weather Data from the Data Source.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The Algorithm processes the data to detect incidents such as accidents, breakdowns, or road closures.   
8. The system formats the detected incidents using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show incident locations.   
10. The Icon Layer is adjusted to visually represent the incidents on the map.   
11. The system stores the incident data in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator that the incident detection is complete.   
13. The Administrator reviews the incident map and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and detection steps.   
4. If the Administrator cancels, the system returns to the main menu without detecting or displaying incidents.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Manage Incident Details   
Use Case ID: UC-16   
Actors: Administrator, User Interface, Incident, Traffic Data, Road Condition Data, Weather Data, Data Source, Data Quality Check, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Data Cache, Data Archive, Traveler Information   
Preconditions: The system is connected to the Data Source for real-time incident, traffic, road condition, and weather data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for visualizing incidents. The Data Cache and Data Archive are ready for storing or updating incident details. The Data Quality Check module is available to ensure data accuracy.   
Postconditions: Incident details are successfully managed (added, updated, or removed). The Data Cache and Data Archive are updated with the latest incident data. The Map Display reflects the current incident status. An Alert is generated to confirm the incident management process is complete.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Manage Incident Details" option.   
3. The system displays a list of available actions: "Add Incident," "Update Incident," or "Remove Incident."   
4. The Administrator selects "Add Incident."   
5. The system prompts the Administrator to input incident details (e.g., location, type, severity, and time of occurrence).   
6. The Administrator enters the required incident information.   
7. The system validates the input data using the Data Quality Check module.   
8. The system connects to the Data Source to check if the incident already exists in the database.   
9. If the incident is not already registered, the system proceeds to store the new incident.   
10. The system saves the new incident data in the Data Cache and Data Archive.   
11. The Map Display is updated using the Michigan Geographic Framework to show the new incident location.   
12. The Icon Layer is modified to represent the incident visually.   
13. The system generates an Alert confirming the successful addition of the incident.   
14. The Administrator confirms the action and closes the process.   
  
Alternative Flow:   
1. If the incident is already registered and the Administrator selects "Add Incident," the system displays an Alert indicating the duplication.   
2. The Administrator can choose to update the existing incident or cancel the operation.   
3. If the Administrator chooses to update, the system allows the input of new or modified incident details.   
4. The system validates the updated data and stores it in the Data Cache and Data Archive.   
5. The Map Display and Icon Layer are updated to reflect the changes.   
6. If the Administrator chooses to cancel, the system returns to the main menu without making any changes.   
7. If the Administrator selects "Remove Incident," the system prompts for the incident identifier (e.g., location and timestamp).   
8. The Administrator enters the incident identifier.   
9. The system validates the input and queries the Data Source to confirm the existence of the incident.   
10. If the incident exists, the system proceeds to delete it from the Data Cache and Data Archive.   
11. The Map Display and Icon Layer are updated to remove the incident representation.   
12. The system generates an Alert confirming the successful removal of the incident.  
  
Use Case Name: Delete Incident Record   
Use Case ID: UC-17   
Actors: Administrator, User Interface, Incident, Data Source, Data Quality Check, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Data Cache, Data Archive   
Preconditions: The system is connected to the Data Source for incident records. The Administrator has access to the User Interface. The Data Cache and Data Archive contain the incident data. The Map Display and Icon Layer are active and reflect current incident locations.   
Postconditions: The incident record is successfully deleted from the system. The Data Cache and Data Archive are updated. The Map Display and Icon Layer are updated to remove the deleted incident's representation. An Alert is generated to confirm the deletion.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Delete Incident Record" option.   
3. The system prompts the Administrator to input the incident identifier (e.g., location and timestamp).   
4. The Administrator enters the required incident identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Source to confirm the existence of the incident record.   
7. If the record exists, the system proceeds to delete it.   
8. The system removes the incident data from the Data Cache and Data Archive.   
9. The Map Display is updated using the Michigan Geographic Framework to remove the incident's location.   
10. The Icon Layer is adjusted to remove the visual representation of the incident.   
11. The system generates an Alert confirming the successful deletion of the incident record.   
12. The Administrator confirms the deletion and closes the process.   
  
Alternative Flow:   
1. If the incident identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and deletion steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: View Incident Information   
Use Case ID: UC-18   
Actors: Administrator, User Interface, Incident, Data Source, Data Quality Check, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Data Cache, Data Archive   
Preconditions: The system is connected to the Data Source for incident-related data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for visualizing incidents. The Data Cache and Data Archive are available for retrieving incident data.   
Postconditions: Incident information is successfully retrieved and displayed. The Map Display and Icon Layer reflect the incident location and details. The Data Cache contains the incident data for quick access. An Alert is generated to confirm the successful viewing of the incident.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "View Incident Information" option.   
3. The system prompts the Administrator to input the incident identifier (e.g., location, type, or timestamp).   
4. The Administrator enters the required incident identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Source to retrieve the incident data.   
7. The system fetches the incident details and location data.   
8. The retrieved incident data is stored in the Data Cache for quick access.   
9. The Map Display is updated using the Michigan Geographic Framework to show the incident location.   
10. The Icon Layer is adjusted to represent the incident visually.   
11. The system generates an Alert to confirm the retrieval and display of the incident information.   
12. The Administrator reviews the incident details and closes the process.   
  
Alternative Flow:   
1. If the incident identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and data retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying incident information.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Data Caching Management   
Use Case ID: UC-19   
Actors: Administrator, User Interface, Data Cache, Data Archive, Data Source, Algorithm, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Data Quality Check   
Preconditions: The system is connected to the Data Source for real-time and historical data. The Administrator has access to the User Interface. The Data Cache and Data Archive are available for caching and archiving data. The Algorithm is configured to manage and optimize the caching process.   
Postconditions: Data caching is successfully managed (added, updated, or removed). The Data Cache and Data Archive are updated accordingly. The Map Display and Icon Layer are updated to reflect any changes in data representation. An Alert is generated to confirm the caching process is complete.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Data Caching Management" option.   
3. The system displays a list of available actions: "Add Data to Cache," "Update Cache Data," or "Remove Data from Cache."   
4. The Administrator selects "Add Data to Cache."   
5. The system prompts the Administrator to specify the data type and identifier (e.g., vehicle, road, or incident data).   
6. The Administrator enters the required data type and identifier.   
7. The system validates the input using the Data Quality Check module.   
8. The system queries the Data Source to retrieve the corresponding data.   
9. The Algorithm processes the data to determine optimal caching strategy and format.   
10. The system stores the data in the Data Cache and archives a copy in the Data Archive.   
11. The Map Display and Icon Layer are updated to reflect the cached data.   
12. The system generates an Alert confirming the successful caching of the data.   
13. The Administrator confirms the action and closes the process.   
  
Alternative Flow:   
1. If the data identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and data retrieval steps.   
4. If the Administrator selects "Update Cache Data," the system prompts for the data identifier and the new data values.   
5. The system validates the updated data and updates the Data Cache and Data Archive accordingly.   
6. The Map Display and Icon Layer are adjusted to reflect the updated data.   
7. If the Administrator selects "Remove Data from Cache," the system prompts for the data identifier.   
8. The system validates the identifier and removes the corresponding data from the Data Cache.   
9. The system updates the Data Archive if necessary.   
10. The Map Display and Icon Layer are updated to remove the data representation.   
11. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Archive Historical Data   
Use Case ID: UC-20   
Actors: Administrator, User Interface, Data Archive, Data Cache, Data Source, Traffic Data, Weather Data, Road Condition Data, Incident, Data Quality Check, Michigan Geographic Framework, Map Display, Icon Layer, Alert   
Preconditions: The system is connected to the Data Source for historical data. The Administrator has access to the User Interface. The Data Cache contains the most recent data. The Data Archive is available for long-term storage. The Map Display and Icon Layer are active and reflect current data.   
Postconditions: Historical data is successfully archived in the Data Archive. The Data Cache is cleared or updated as needed. The Map Display and Icon Layer remain unchanged. An Alert is generated to confirm the successful archiving.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Archive Historical Data" option.   
3. The system prompts the Administrator to specify the data type and time range for archiving (e.g., traffic, weather, or incident data).   
4. The Administrator enters the required parameters.   
5. The system performs a Data Quality Check to ensure the selected data is complete and consistent.   
6. The system retrieves the specified historical data from the Data Cache or Data Source.   
7. The system compresses and formats the data for archiving using the appropriate format (e.g., SAE J2354 or TMDD).   
8. The system stores the historical data in the Data Archive.   
9. The system generates an Alert confirming the successful archiving of the data.   
10. The Administrator confirms the archiving and closes the process.   
  
Alternative Flow:   
1. If the specified data type or time range is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the validation and data retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without archiving any data.   
5. If the Data Source or Data Cache is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Retrieve Archived Data   
Use Case ID: UC-21   
Actors: Administrator, User Interface, Data Archive, Data Cache, Data Source, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Data Quality Check   
Preconditions: The system is connected to the Data Archive for historical data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for data visualization. The Data Cache is available for temporary storage.   
Postconditions: The requested archived data is successfully retrieved and displayed. The Data Cache is updated with the retrieved data. The Map Display and Icon Layer reflect the archived data as needed. An Alert is generated to confirm the retrieval.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Retrieve Archived Data" option.   
3. The system prompts the Administrator to input the data type and identifier (e.g., vehicle, road condition, or incident data).   
4. The Administrator enters the required data type and identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Archive to retrieve the specified historical data.   
7. The retrieved data is temporarily stored in the Data Cache for quick access.   
8. The system processes and formats the archived data using the appropriate Output Format and Presentation Method.   
9. If the data involves spatial information, the Map Display is updated using the Michigan Geographic Framework.   
10. The Icon Layer is adjusted to visually represent the archived data on the map.   
11. The system generates an Alert to confirm the successful retrieval of the archived data.   
12. The Administrator reviews the retrieved data and closes the process.   
  
Alternative Flow:   
1. If the data type or identifier is invalid or not found in the Data Archive, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and data retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying data.   
5. If the Data Archive is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Data Quality Validation   
Use Case ID: UC-22   
Actors: Administrator, User Interface, Data Source, Data Quality Check, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Alert   
Preconditions: The system is connected to the Data Source for traffic, weather, and road condition data. The Administrator has access to the User Interface. The Data Quality Check module is active and configured to validate data integrity. The Data Cache and Data Archive are ready for data processing. The Map Display and Icon Layer are active for visualization.   
Postconditions: The system validates the quality of incoming data. Invalid or corrupted data is flagged or corrected. Valid data is stored in the Data Cache and Data Archive. The Map Display and Icon Layer are updated if necessary. An Alert is generated to notify the Administrator of the validation outcome.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Data Quality Validation" option.   
3. The system prompts the Administrator to specify the data type (e.g., Traffic Data, Weather Data, or Road Condition Data) and the time range for validation.   
4. The Administrator enters the required data type and time range.   
5. The system retrieves the selected data from the Data Source or Data Cache.   
6. The Data Quality Check module analyzes the data for consistency, completeness, and accuracy.   
7. The system identifies any invalid or corrupted data entries and flags them for review.   
8. Valid data is stored in the Data Cache and Data Archive for further processing.   
9. The Map Display and Icon Layer are updated to reflect any changes in data quality status.   
10. The system generates an Alert summarizing the validation results (e.g., number of valid and invalid records).   
11. The Administrator reviews the validation results and closes the process.   
  
Alternative Flow:   
1. If the specified data type or time range is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and validation steps.   
4. If the Administrator cancels, the system returns to the main menu without performing the validation.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.   
6. If the validation reveals a high number of invalid data entries, the system may suggest reprocessing the data or initiating a data correction workflow.  
  
Use Case Name: Modify Data Quality Rule   
Use Case ID: UC-23   
Actors: Administrator, User Interface, Data Quality Check, Data Cache, Data Archive, Traffic Data, Weather Data, Road Condition Data, Algorithm, Alert   
Preconditions: The system is connected to the Data Source for rule definitions and data quality metrics. The Administrator has access to the User Interface. The Data Quality Check module is active and available for rule modification. The Data Cache and Data Archive are ready to reflect data quality rule changes.   
Postconditions: The data quality rule is successfully modified in the system. The Data Quality Check module is updated to use the new rule. The Data Cache and Data Archive are updated to reflect the changes in data validation. The Administrator is notified via an Alert.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Modify Data Quality Rule" option.   
3. The system displays a list of existing data quality rules for review.   
4. The Administrator selects the rule to be modified.   
5. The system prompts the Administrator to input the new rule parameters (e.g., validation thresholds, data formats, or error tolerances).   
6. The Administrator enters the updated rule details.   
7. The system validates the new rule using the Data Quality Check module.   
8. The system updates the rule in the Data Quality Check module and stores the rule in the Data Archive.   
9. The system generates an Alert confirming the successful modification of the data quality rule.   
10. The Administrator confirms the modification and closes the process.   
  
Alternative Flow:   
1. If the rule parameters are invalid or conflicting, the system displays an Alert indicating the failure.   
2. The Administrator can choose to correct the rule or cancel the modification.   
3. If the Administrator corrects the rule, the system repeats the validation and modification steps.   
4. If the Administrator cancels, the system returns to the main menu without modifying the rule.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Administrator Login   
Use Case ID: UC-00   
Actors: Administrator, User Interface, Data Cache, Alert   
Preconditions: The User Interface is accessible. The system is connected to the Data Source for administrator credentials. The Data Cache contains cached login attempts for session tracking. The administrator has valid login credentials.   
Postconditions: The administrator is authenticated and granted access to the system. The Data Cache updates the login session status. An Alert is generated to confirm successful login.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The system displays the login screen with fields for username and password.   
3. The Administrator enters their username and password.   
4. The system validates the credentials against the Data Source.   
5. If the credentials are valid, the system grants access to the administrator dashboard.   
6. The system updates the Data Cache with the login session information.   
7. An Alert is generated to confirm the successful login.   
8. The Administrator proceeds to use the system.   
  
Alternative Flow:   
1. If the credentials are invalid, the system displays an Alert indicating the login failure.   
2. The Administrator can choose to re-enter the credentials or cancel the login attempt.   
3. If the Administrator re-enters the credentials, the system repeats the validation process.   
4. If the Administrator cancels, the system returns to the login screen or exits the User Interface.   
5. If the Data Source is unreachable, the system generates an Alert and halts the login process.  
  
Use Case Name: Access User Interface   
Use Case ID: UC-24   
Actors: Administrator, User Interface, Data Cache, Alert   
Preconditions: The User Interface is accessible and operational. The Administrator has valid login credentials. The Data Cache is ready to store session information.   
Postconditions: The Administrator successfully accesses the User Interface. The session is recorded in the Data Cache. An Alert is generated to confirm the access.   
  
Main Flow:   
1. The Administrator navigates to the User Interface.   
2. The system displays the login screen for authentication.   
3. The Administrator enters their username and password.   
4. The system validates the credentials against the Data Source.   
5. If the validation is successful, the system grants access to the User Interface dashboard.   
6. The system updates the Data Cache with session details.   
7. An Alert is generated to inform the Administrator of successful access.   
8. The Administrator begins interacting with the User Interface.   
  
Alternative Flow:   
1. If the credentials are invalid, the system displays an Alert indicating the access failure.   
2. The Administrator can choose to re-enter the credentials or cancel the access attempt.   
3. If the Administrator re-enters the credentials, the system repeats the validation process.   
4. If the Administrator cancels, the system exits the login screen.   
5. If the Data Source is unreachable, the system generates an Alert and halts the access process.  
  
Use Case Name: Browse Traffic Information   
Use Case ID: UC-25   
Actors: Administrator, User Interface, Traffic Data, Map Display, Icon Layer, Data Source, Data Quality Check, Michigan Geographic Framework, Alert, Data Cache   
Preconditions: The system is connected to the Data Source for real-time traffic data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active for traffic visualization. The Data Quality Check module is available to ensure data accuracy. The Data Cache is ready to store frequently accessed traffic data.   
Postconditions: Traffic information is successfully retrieved and displayed. The Map Display and Icon Layer reflect the current traffic status. The Data Cache is updated for quick access. An Alert is generated to confirm the successful browsing of traffic information.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Browse Traffic Information" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for traffic data.   
4. The Administrator enters the required parameters.   
5. The system validates the input using the Data Quality Check module.   
6. The system retrieves Traffic Data from the Data Source based on the specified area and time.   
7. The system processes and normalizes the data using the Algorithm.   
8. The Map Display is updated using the Michigan Geographic Framework to show the current traffic conditions.   
9. The Icon Layer is adjusted to visually represent traffic incidents, congestion, or delays.   
10. The system stores the retrieved traffic data in the Data Cache for quick access.   
11. The system generates an Alert to inform the Administrator that the traffic information is ready for review.   
12. The Administrator reviews the displayed traffic data and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the validation and data retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying traffic information.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Display Map Overview   
Use Case ID: UC-26   
Actors: Administrator, User Interface, Map Display, Icon Layer, Michigan Geographic Framework, Data Source, Data Cache, Data Quality Check, Traffic Data, Road Condition Data, Weather Data, Alert   
Preconditions: The system is connected to the Data Source for traffic, road condition, and weather data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active and ready for visualization. The Data Cache is available to provide quick access to frequently used data.   
Postconditions: The map overview is successfully displayed. The Map Display and Icon Layer reflect current traffic, road, and weather conditions. The Data Cache is updated for quick reference. An Alert is generated to confirm the successful display.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Display Map Overview" option.   
3. The system retrieves the latest Traffic Data, Road Condition Data, and Weather Data from the Data Source or Data Cache.   
4. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
5. The Algorithm processes the data to determine visual representation priorities and updates.   
6. The Map Display is updated using the Michigan Geographic Framework to show the current overview.   
7. The Icon Layer is adjusted to visually represent traffic incidents, road conditions, and weather effects.   
8. The system generates an Alert confirming the successful display of the map overview.   
9. The Administrator reviews the map and closes the process.   
  
Alternative Flow:   
1. If the Data Source is unreachable or returns corrupted data, the system displays an Alert indicating the failure.   
2. The Administrator can choose to retry the data retrieval or cancel the operation.   
3. If the Administrator selects to retry, the system reconnects to the Data Source and repeats the data retrieval and processing steps.   
4. If the Administrator cancels, the system returns to the main menu without displaying the map overview.   
5. If the Data Quality Check fails due to inconsistent or incomplete data, the system alerts the Administrator and suggests data correction or reprocessing.  
  
Use Case Name: Toggle Icon Layer   
Use Case ID: UC-27   
Actors: Administrator, User Interface, Map Display, Icon Layer, Data Cache, Alert, Michigan Geographic Framework, Traffic Data, Road Condition Data, Weather Data, Incident   
Preconditions: The system is connected to the Data Source for real-time traffic, road condition, and weather data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active and initialized. The Data Cache is available to store the current state of the Icon Layer.   
Postconditions: The Icon Layer is toggled on or off on the Map Display. The Data Cache is updated to reflect the current visibility status of the Icon Layer. The Administrator is notified via an Alert of the toggle action.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Toggle Icon Layer" option.   
3. The system prompts the Administrator to choose whether to enable or disable the Icon Layer.   
4. The Administrator selects the desired option.   
5. The system updates the visibility status of the Icon Layer on the Map Display.   
6. The system stores the updated status in the Data Cache for quick reference.   
7. The system generates an Alert confirming the successful toggle of the Icon Layer.   
8. The Administrator confirms the change and closes the process.   
  
Alternative Flow:   
1. If the Map Display is not active or the Icon Layer is not properly initialized, the system displays an Alert indicating the failure.   
2. The Administrator can choose to resolve the issue or cancel the operation.   
3. If the Administrator cancels, the system returns to the main menu without toggling the Icon Layer.  
  
Use Case Name: Declutter Map View   
Use Case ID: UC-28   
Actors: Administrator, User Interface, Map Display, Icon Layer, Data Cache, Data Archive, Traffic Data, Road Condition Data, Weather Data, Michigan Geographic Framework, Alert   
Preconditions: The system is connected to the Data Source for traffic, road condition, and weather data. The Administrator has access to the User Interface. The Map Display and Icon Layer are active and contain data visualizations. The Data Cache stores current map display settings.   
Postconditions: The Map Display is decluttered to improve visual clarity. The Icon Layer is updated to hide or show specific data layers as configured. The Data Cache is updated with the new decluttering settings. An Alert is generated to confirm the decluttering is complete.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Declutter Map View" option.   
3. The system prompts the Administrator to choose which data layers to hide or show (e.g., traffic incidents, road conditions, weather effects).   
4. The Administrator selects the desired data layers to declutter.   
5. The system updates the visibility of the selected layers on the Map Display and Icon Layer.   
6. The system stores the decluttering settings in the Data Cache for quick access in future sessions.   
7. The system generates an Alert to confirm the successful decluttering of the map view.   
8. The Administrator confirms the decluttering and closes the process.   
  
Alternative Flow:   
1. If the Administrator selects an invalid or unsupported data layer, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the selection or cancel the operation.   
3. If the Administrator re-enters the selection, the system repeats the decluttering process.   
4. If the Administrator cancels, the system returns to the main menu without changing the map view.   
5. If the Data Cache is unreachable or corrupted, the system generates an Alert and halts the process.  
  
Use Case Name: Add Data Source   
Use Case ID: UC-29   
Actors: Administrator, User Interface, Data Source, Data Cache, Data Archive, Alert, Data Quality Check, Michigan Geographic Framework, Map Display   
Preconditions: The Administrator has access to the User Interface. The system is not currently connected to the new Data Source. The Data Cache and Data Archive are ready to store metadata about the new Data Source. The Map Display and Michigan Geographic Framework are available for visualization if location-based data is involved.   
Postconditions: The new Data Source is successfully added to the system. The metadata about the Data Source is stored in the Data Cache and Data Archive. The Administrator is notified via an Alert. The system is prepared to retrieve and process data from the newly added source.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Add Data Source" option.   
3. The system prompts the Administrator to input the Data Source details (e.g., name, type, connection parameters, data format).   
4. The Administrator enters the required information.   
5. The system validates the input data using the Data Quality Check module.   
6. The system checks if the Data Source is already registered in the system.   
7. If the Data Source is not already registered, the system proceeds to add it.   
8. The system stores the Data Source metadata in the Data Cache and Data Archive.   
9. The system generates an Alert to confirm the successful addition of the Data Source.   
10. The Administrator confirms the addition and closes the process.   
  
Alternative Flow:   
1. If the Data Source is already registered, the system displays an Alert indicating the duplication.   
2. The Administrator can choose to update the existing Data Source information or cancel the addition process.   
3. If the Administrator chooses to update, the system allows the input of new or modified Data Source details.   
4. The system validates the updated information and stores it in the Data Cache and Data Archive.   
5. The system generates an Alert to confirm the update.   
6. If the Administrator chooses to cancel, the system returns to the main menu without making any changes.  
  
Use Case Name: Configure Algorithm   
Use Case ID: UC-30   
Actors: Administrator, User Interface, Algorithm, Data Source, Data Quality Check, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Alert   
  
Preconditions:   
- The Administrator has access to the User Interface.   
- The system is connected to the Data Source for algorithm configuration parameters.   
- The Algorithm module is available and ready to be configured.   
- The Data Quality Check is active to validate input parameters.   
- The Map Display and Icon Layer are active for visualization if needed.   
  
Postconditions:   
- The Algorithm is successfully configured with the specified parameters.   
- The configuration details are stored in the Data Cache and Data Archive.   
- An Alert is generated to confirm the successful configuration.   
- The Map Display and Icon Layer are updated if the configuration affects visual representation.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Configure Algorithm" option.   
3. The system displays a list of available algorithms and their configuration parameters.   
4. The Administrator selects the algorithm to be configured.   
5. The system prompts the Administrator to input the new configuration parameters (e.g., thresholds, weights, or data formats).   
6. The Administrator enters the required parameters.   
7. The system validates the input using the Data Quality Check module.   
8. The system updates the Algorithm with the new configuration.   
9. The system stores the updated configuration in the Data Cache and Data Archive.   
10. The system generates an Alert to inform the Administrator of the successful configuration.   
11. The Administrator confirms the configuration and closes the process.   
  
Alternative Flow:   
1. If the specified algorithm is not available or the parameters are invalid, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the parameters or cancel the configuration.   
3. If the Administrator re-enters the parameters, the system repeats the validation and configuration steps.   
4. If the Administrator cancels, the system returns to the main menu without modifying the algorithm.   
5. If the Data Source is unreachable or returns corrupted configuration data, the system generates an Alert and halts the process.  
  
Use Case Name: Update Algorithm   
Use Case ID: UC-31   
Actors: Administrator, User Interface, Algorithm, Data Source, Data Quality Check, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Alert   
  
Preconditions:   
- The system is connected to the Data Source for algorithm updates or configurations.   
- The Administrator has access to the User Interface.   
- The Algorithm module is available and ready for modification.   
- The Data Quality Check is active to validate the new algorithm settings.   
- The Map Display and Icon Layer are active for visualization if the algorithm affects spatial data.   
- The Data Cache and Data Archive are available for storing the updated algorithm configuration.   
  
Postconditions:   
- The Algorithm is successfully updated with the new configuration or version.   
- The updated algorithm is stored in the Data Cache and Data Archive for reference.   
- The Map Display and Icon Layer are updated if the algorithm change affects data visualization.   
- An Alert is generated to confirm the successful update.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Update Algorithm" option.   
3. The system displays the current algorithm configuration for reference.   
4. The Administrator inputs the new algorithm parameters or selects an updated version from the Data Source.   
5. The system validates the new configuration using the Data Quality Check module.   
6. The system updates the Algorithm with the new settings or version.   
7. The system stores the updated algorithm configuration in the Data Cache and Data Archive.   
8. The system generates an Alert to confirm the successful update of the Algorithm.   
9. The Administrator confirms the update and closes the process.   
  
Alternative Flow:   
1. If the new algorithm parameters are invalid or incompatible, the system displays an Alert indicating the failure.   
2. The Administrator can choose to correct the input or cancel the update.   
3. If the Administrator corrects the input, the system repeats the validation and update steps.   
4. If the Administrator cancels, the system returns to the main menu without modifying the Algorithm.   
5. If the Data Source is unreachable or returns corrupted algorithm data, the system generates an Alert and halts the process.  
  
Use Case Name: Delete Algorithm   
Use Case ID: UC-32   
Actors: Administrator, User Interface, Algorithm, Data Source, Data Quality Check, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Alert   
  
Preconditions:   
- The system is connected to the Data Source for algorithm information.   
- The Administrator has access to the User Interface.   
- The Algorithm module is registered in the system.   
- The Data Cache and Data Archive are ready to store or update algorithm-related data.   
- The Map Display and Icon Layer are active for visualization if the algorithm affects spatial data.   
  
Postconditions:   
- The specified Algorithm is successfully deleted from the system.   
- The Data Cache and Data Archive are updated to remove the algorithm configuration.   
- The Map Display and Icon Layer are updated if the deleted algorithm affects visualizations.   
- An Alert is generated to confirm the deletion of the algorithm.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Delete Algorithm" option.   
3. The system prompts the Administrator to input the identifier of the Algorithm to be deleted (e.g., name or version).   
4. The Administrator enters the required Algorithm identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Source to confirm the existence of the Algorithm.   
7. If the Algorithm exists, the system proceeds to delete it.   
8. The system removes the Algorithm configuration from the Data Cache and Data Archive.   
9. The Map Display and Icon Layer are updated if the deleted algorithm affects spatial data visualization.   
10. The system generates an Alert confirming the successful deletion of the Algorithm.   
11. The Administrator confirms the deletion and closes the process.   
  
Alternative Flow:   
1. If the Algorithm identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and deletion steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Format Data Output   
Use Case ID: UC-33   
Actors: Administrator, User Interface, Data Source, Data Cache, Data Archive, Output Format, Presentation Method, Algorithm, Alert, Map Display, Icon Layer, Michigan Geographic Framework   
  
Preconditions:   
- The system is connected to the Data Source for the data to be formatted.   
- The Administrator has access to the User Interface.   
- The Output Format and Presentation Method modules are available for configuration.   
- The Data Cache and Data Archive are ready to store or update formatted data.   
- The Algorithm is available for data transformation.   
  
Postconditions:   
- The data is successfully formatted according to the specified Output Format and Presentation Method.   
- The formatted data is stored in the Data Cache and Data Archive.   
- The Map Display and Icon Layer are updated if the formatted data affects visualization.   
- An Alert is generated to confirm the successful formatting of the data.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Format Data Output" option.   
3. The system prompts the Administrator to select the data type (e.g., traffic, weather, road condition, or incident data) and specify the desired Output Format (e.g., SAE J2354, TMDD, or custom format).   
4. The Administrator chooses the data type and selects the Output Format and Presentation Method.   
5. The system retrieves the selected data from the Data Source or Data Cache.   
6. The system applies the Algorithm to transform the data into the specified Output Format.   
7. The system validates the formatted data using the Data Quality Check module.   
8. The system stores the formatted data in the Data Archive and updates the Data Cache.   
9. If the formatted data affects visualization, the Map Display and Icon Layer are updated.   
10. The system generates an Alert to inform the Administrator that the data has been successfully formatted.   
11. The Administrator confirms the formatting and closes the process.   
  
Alternative Flow:   
1. If the selected data type is invalid or not available in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the data type or cancel the operation.   
3. If the Administrator re-enters the data type, the system repeats the data retrieval and formatting steps.   
4. If the Administrator cancels, the system returns to the main menu without formatting the data.   
5. If the Output Format or Presentation Method is unsupported, the system displays an Alert and prompts the Administrator to choose a valid option.   
6. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Modify Output Format   
Use Case ID: UC-34   
Actors: Administrator, User Interface, Output Format, Presentation Method, Data Cache, Data Archive, Algorithm, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Data Source   
  
Preconditions:   
- The system is connected to the Data Source for the data to be reformatted.   
- The Administrator has access to the User Interface.   
- The Output Format and Presentation Method modules are active and available for modification.   
- The Data Cache and Data Archive are ready to store or update data in the new format.   
- The Algorithm is available to apply the transformation.   
  
Postconditions:   
- The Output Format is successfully modified according to the Administrator's specifications.   
- The Presentation Method is updated to reflect the new format.   
- The data is reformatted and stored in the Data Cache and Data Archive.   
- The Map Display and Icon Layer are updated if the new format affects spatial data visualization.   
- An Alert is generated to confirm the successful modification of the Output Format.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Modify Output Format" option.   
3. The system displays the current Output Format and Presentation Method for reference.   
4. The Administrator chooses the data type to be reformatted (e.g., traffic, weather, road condition, or incident data).   
5. The system prompts the Administrator to select the new Output Format (e.g., SAE J2354, TMDD, or custom format).   
6. The Administrator selects the new Output Format and Presentation Method.   
7. The system retrieves the data from the Data Source or Data Cache.   
8. The system uses the Algorithm to convert the data into the new format.   
9. The system validates the reformatted data using the Data Quality Check module.   
10. The system updates the Output Format and Presentation Method configurations.   
11. The system stores the reformatted data in the Data Cache and Data Archive.   
12. If the format change affects visualizations, the Map Display and Icon Layer are updated.   
13. The system generates an Alert to confirm the successful modification of the Output Format.   
14. The Administrator confirms the change and closes the process.   
  
Alternative Flow:   
1. If the selected Output Format or Presentation Method is invalid or unsupported, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the format or cancel the modification.   
3. If the Administrator re-enters the format, the system repeats the validation and modification steps.   
4. If the Administrator cancels, the system returns to the main menu without changing the Output Format.   
5. If the Data Source is unreachable or returns corrupted data during retrieval, the system generates an Alert and halts the process.  
  
Use Case Name: Publish Data to SAE J2354   
Use Case ID: UC-35   
Actors: Administrator, User Interface, Data Source, Data Cache, Data Archive, SAE J2354 Format, Algorithm, Data Quality Check, Alert, Traffic Data, Road Condition Data, Weather Data, Traveler Information, Michigan Geographic Framework, Map Display, Icon Layer   
  
Preconditions:   
- The system is connected to the Data Source for traffic, road condition, and weather data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive contain the data to be published.   
- The Algorithm is configured to transform data into SAE J2354 format.   
- The Data Quality Check module is active to ensure data integrity.   
- The Map Display and Icon Layer are active for visualization if needed.   
  
Postconditions:   
- The data is successfully published in SAE J2354 format.   
- The Data Cache and Data Archive are updated to reflect the publication status.   
- An Alert is generated to confirm the data has been published.   
- The Map Display and Icon Layer are updated if the published data affects traffic visualization.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Publish Data to SAE J2354" option.   
3. The system prompts the Administrator to choose the data type to publish (e.g., traffic, road condition, or weather data).   
4. The Administrator selects the data type.   
5. The system retrieves the selected data from the Data Cache or Data Source.   
6. The system performs a Data Quality Check to ensure the data is valid and consistent.   
7. The Algorithm transforms the data into SAE J2354 format.   
8. The system publishes the formatted data to the SAE J2354 interface.   
9. The system updates the Data Cache and Data Archive to reflect the publication.   
10. The Map Display and Icon Layer are updated if the data change affects visual representation.   
11. The system generates an Alert confirming the successful publication of the data.   
12. The Administrator confirms the publication and closes the process.   
  
Alternative Flow:   
1. If the selected data type is not available or the data fails the Data Quality Check, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the data type or cancel the operation.   
3. If the Administrator re-enters the data type, the system repeats the data retrieval and transformation steps.   
4. If the Administrator cancels, the system returns to the main menu without publishing the data.   
5. If the SAE J2354 interface is unreachable or the publication fails, the system generates an Alert and halts the process.  
  
Use Case Name: Publish Data to TMDD   
Use Case ID: UC-36   
Actors: Administrator, User Interface, Data Source, Data Cache, Data Archive, TMDD Format, Algorithm, Data Quality Check, Alert, Traffic Data, Road Condition Data, Weather Data, Traveler Information, Michigan Geographic Framework, Map Display, Icon Layer   
  
Preconditions:   
- The system is connected to the Data Source for real-time traffic, road condition, and weather data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive contain the data to be published.   
- The Algorithm is configured to transform data into TMDD format.   
- The Data Quality Check module is active to ensure data accuracy.   
- The Map Display and Icon Layer are active for visualization if needed.   
  
Postconditions:   
- The data is successfully published in TMDD format.   
- The Data Cache and Data Archive are updated to reflect the publication status.   
- An Alert is generated to confirm the data has been published.   
- The Map Display and Icon Layer are updated if the published data affects traffic visualization.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Publish Data to TMDD" option.   
3. The system prompts the Administrator to choose the data type to publish (e.g., traffic, road condition, or weather data).   
4. The Administrator selects the data type.   
5. The system retrieves the selected data from the Data Cache or Data Source.   
6. The system performs a Data Quality Check to ensure the data is valid and consistent.   
7. The Algorithm transforms the data into TMDD format.   
8. The system publishes the formatted data to the TMDD interface.   
9. The system updates the Data Cache and Data Archive to reflect the publication.   
10. The Map Display and Icon Layer are updated if the data change affects visual representation.   
11. The system generates an Alert confirming the successful publication of the data.   
12. The Administrator confirms the publication and closes the process.   
  
Alternative Flow:   
1. If the selected data type is not available or the data fails the Data Quality Check, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the data type or cancel the operation.   
3. If the Administrator re-enters the data type, the system repeats the data retrieval and transformation steps.   
4. If the Administrator cancels, the system returns to the main menu without publishing the data.   
5. If the TMDD interface is unreachable or the publication fails, the system generates an Alert and halts the process.  
  
Use Case Name: Generate Alert   
Use Case ID: UC-37   
Actors: Administrator, User Interface, Alert, Data Cache, Data Archive, Data Quality Check, Traffic Data, Weather Data, Road Condition Data, Incident, Map Display, Icon Layer, Michigan Geographic Framework   
  
Preconditions:   
- The system is connected to the Data Source for real-time traffic, weather, and road condition data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive are available for storing alert-related data.   
- The Data Quality Check module is active to validate data before alert generation.   
- The Map Display and Icon Layer are active for visualizing the alert context if applicable.   
  
Postconditions:   
- An Alert is successfully generated based on the specified criteria.   
- The alert details are stored in the Data Cache and Data Archive.   
- The Map Display and Icon Layer are updated to reflect the alert location or context.   
- The Administrator is notified of the alert generation.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Generate Alert" option.   
3. The system prompts the Administrator to specify the alert criteria (e.g., type, severity, geographic area, and time frame).   
4. The Administrator provides the required parameters.   
5. The system validates the input using the Data Quality Check module.   
6. The system retrieves relevant data (e.g., Traffic Data, Road Condition Data, Weather Data, or Incident) from the Data Source or Data Cache.   
7. The system processes the data to determine if an alert condition is met (e.g., high congestion, road closure, severe weather).   
8. If the condition is met, the system generates the Alert with the specified details.   
9. The system stores the Alert in the Data Cache and Data Archive.   
10. The Map Display is updated using the Michigan Geographic Framework to show the alert location or context.   
11. The Icon Layer is adjusted to visually represent the alert.   
12. The system generates an Alert notification to inform the Administrator.   
13. The Administrator confirms the alert generation and closes the process.   
  
Alternative Flow:   
1. If the alert criteria are invalid or inconsistent, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the criteria or cancel the alert generation.   
3. If the Administrator re-enters the criteria, the system repeats the validation and data retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without generating an alert.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Modify Alert Settings   
Use Case ID: UC-38   
Actors: Administrator, User Interface, Alert, Data Cache, Data Archive, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Presentation Method, Output Format   
  
Preconditions:   
- The Administrator has access to the User Interface.   
- The system is connected to the Data Source for alert configuration.   
- The Alert module is active and ready to accept new or modified settings.   
- The Data Cache and Data Archive are available to store updated alert settings.   
- The Map Display and Icon Layer are active if alert settings involve spatial visualization.   
  
Postconditions:   
- The alert settings are successfully modified in the system.   
- The updated settings are stored in the Data Cache and Data Archive.   
- The Map Display and Icon Layer are updated if the alert settings affect visual representation.   
- An Alert is generated to confirm the modification.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Modify Alert Settings" option.   
3. The system displays the current alert settings for review.   
4. The Administrator modifies the alert settings (e.g., severity thresholds, notification frequency, or data types to trigger alerts).   
5. The system validates the modified settings using the Data Quality Check module.   
6. The system updates the Alert configuration with the new settings.   
7. The system stores the updated alert settings in the Data Cache and Data Archive.   
8. The system generates an Alert to confirm the successful modification of the alert settings.   
9. The Administrator confirms the modification and closes the process.   
  
Alternative Flow:   
1. If the modified alert settings are invalid or inconsistent, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the settings or cancel the modification.   
3. If the Administrator re-enters the settings, the system repeats the validation and modification steps.   
4. If the Administrator cancels, the system returns to the main menu without applying changes.   
5. If the Data Source is unreachable or returns corrupted data during validation, the system generates an Alert and halts the process.  
  
Use Case Name: View Alert History   
Use Case ID: UC-39   
Actors: Administrator, User Interface, Alert, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Data Quality Check   
  
Preconditions:   
- The system is connected to the Data Source for historical alert data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive contain the alert history records.   
- The Map Display and Icon Layer are active and ready for visualization if spatial context is needed.   
- The Data Quality Check module is available to ensure the integrity of the alert data.   
  
Postconditions:   
- The alert history is successfully retrieved and displayed to the Administrator.   
- The Data Cache is updated with the retrieved alert history for quick access.   
- The Map Display and Icon Layer reflect the locations and context of historical alerts.   
- An Alert is generated to confirm the successful retrieval of the alert history.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "View Alert History" option.   
3. The system prompts the Administrator to specify a time range or filter criteria for the alerts.   
4. The Administrator provides the required parameters.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Archive to retrieve the alert history based on the specified criteria.   
7. The system retrieves the relevant Alert data, including details such as type, severity, time, and location.   
8. The system stores the retrieved alert history in the Data Cache for quick reference.   
9. If the alert data includes spatial information, the Map Display is updated using the Michigan Geographic Framework to show alert locations.   
10. The Icon Layer is adjusted to represent the historical alerts visually.   
11. The system generates an Alert to confirm the successful retrieval of the alert history.   
12. The Administrator reviews the displayed alert history and closes the process.   
  
Alternative Flow:   
1. If the specified time range or filter is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the criteria or cancel the operation.   
3. If the Administrator re-enters the criteria, the system repeats the validation and retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying alert history.   
5. If the Data Archive is unreachable or returns corrupted alert data, the system generates an Alert and halts the process.  
  
Use Case Name: Travel Demand Analysis   
Use Case ID: UC-40   
Actors: Administrator, User Interface, Travel Demand, Data Source, Data Cache, Data Archive, Algorithm, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Traffic Data, Road Condition Data, Weather Data, Traveler Information   
  
Preconditions:   
- The system is connected to the Data Source for travel demand-related data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive are ready to store or retrieve travel demand data.   
- The Algorithm is configured to analyze travel patterns and predict demand.   
- The Map Display and Icon Layer are active for visualizing travel demand.   
  
Postconditions:   
- Travel demand data is successfully analyzed and visualized.   
- The results are stored in the Data Cache and Data Archive.   
- The Map Display and Icon Layer reflect the current travel demand analysis.   
- An Alert is generated to confirm the analysis completion.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Travel Demand Analysis" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for the analysis.   
4. The Administrator provides the required parameters.   
5. The system retrieves relevant data from the Data Source, including Traffic Data, Weather Data, and Road Condition Data.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The Algorithm processes the data to calculate and predict travel demand based on historical trends and real-time conditions.   
8. The system formats the analysis results using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show travel demand distribution.   
10. The Icon Layer is adjusted to visually represent areas with high or low demand.   
11. The system stores the analysis results in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator that the travel demand analysis is complete.   
13. The Administrator reviews the analysis and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the analysis.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and analysis steps.   
4. If the Administrator cancels, the system returns to the main menu without performing the analysis.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Update Travel Demand Model   
Use Case ID: UC-41   
Actors: Administrator, User Interface, Travel Demand, Data Source, Algorithm, Data Cache, Data Archive, Map Display, Icon Layer, Traffic Data, Road Condition Data, Weather Data, Michigan Geographic Framework, Alert   
  
Preconditions:   
- The system is connected to the Data Source for real-time and historical travel demand data.   
- The Administrator has access to the User Interface.   
- The Algorithm is configured to update and refine the travel demand model.   
- The Data Cache and Data Archive are ready to store updated model parameters.   
- The Map Display and Icon Layer are active for visualizing travel demand patterns.   
  
Postconditions:   
- The travel demand model is successfully updated with new or modified parameters.   
- The updated model is stored in the Data Cache and Data Archive.   
- The Map Display reflects the updated travel demand visualization.   
- An Alert is generated to confirm the model update is complete.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Update Travel Demand Model" option.   
3. The system prompts the Administrator to input new or modified parameters for the model (e.g., population growth, travel patterns, or traffic influence factors).   
4. The Administrator enters the required model parameters.   
5. The system validates the input using the Data Quality Check module.   
6. The system retrieves existing travel demand data from the Data Source or Data Cache.   
7. The Algorithm processes the new parameters and updates the travel demand model accordingly.   
8. The system formats the updated model using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show the new travel demand distribution.   
10. The Icon Layer is adjusted to reflect changes in demand levels.   
11. The system stores the updated model in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator of the successful update.   
13. The Administrator reviews the updated model and closes the process.   
  
Alternative Flow:   
1. If the input parameters are invalid or incompatible with the current model, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the parameters or cancel the update.   
3. If the Administrator re-enters the parameters, the system repeats the validation and model update steps.   
4. If the Administrator cancels, the system returns to the main menu without modifying the model.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Asset Condition Monitoring   
Use Case ID: UC-42   
Actors: Administrator, User Interface, Asset Condition, Data Source, Data Cache, Data Archive, Algorithm, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Traveler Information   
  
Preconditions:   
- The system is connected to the Data Source for asset-related data.   
- The Administrator has access to the User Interface.   
- The Map Display and Icon Layer are active and configured to visualize asset conditions.   
- The Data Cache and Data Archive are available for storing and retrieving asset data.   
- The Algorithm is configured to analyze and classify asset conditions.   
  
Postconditions:   
- Asset condition data is successfully retrieved, analyzed, and displayed.   
- The Map Display and Icon Layer reflect the current asset condition status.   
- The results are stored in the Data Cache and Data Archive.   
- An Alert is generated to confirm the successful monitoring of asset conditions.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Asset Condition Monitoring" option.   
3. The system prompts the Administrator to specify the geographic area and asset type for monitoring.   
4. The Administrator provides the required parameters.   
5. The system retrieves Asset Condition data from the Data Source or Data Cache.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The Algorithm processes the data to analyze asset conditions (e.g., damage, wear, or performance status).   
8. The system formats the analysis results using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show asset condition status.   
10. The Icon Layer is adjusted to reflect the current condition of the assets (e.g., warning icons for damaged assets).   
11. The system stores the analysis results in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator that the asset condition monitoring is complete.   
13. The Administrator reviews the asset condition map and closes the process.   
  
Alternative Flow:   
1. If the geographic area or asset type is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the data retrieval and analysis steps.   
4. If the Administrator cancels, the system returns to the main menu without displaying asset conditions.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Update Asset Status   
Use Case ID: UC-43   
Actors: Administrator, User Interface, Asset Condition, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Alert, Algorithm   
  
Preconditions:   
- The system is connected to the Data Source for asset status and condition data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive are available for storing or updating asset data.   
- The Map Display and Icon Layer are active and ready for visualization.   
- The Algorithm is configured to process and validate asset condition updates.   
  
Postconditions:   
- The asset status is successfully updated in the system.   
- The Data Cache and Data Archive are updated with the new asset condition data.   
- The Map Display and Icon Layer are updated to reflect the current asset status.   
- An Alert is generated to confirm the successful update of the asset status.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Update Asset Status" option.   
3. The system prompts the Administrator to input the asset identifier (e.g., asset ID or location).   
4. The Administrator enters the required asset identifier.   
5. The system validates the identifier using the Data Quality Check module.   
6. The system retrieves the current asset condition data from the Data Source.   
7. The system displays the current asset status to the Administrator for review.   
8. The Administrator modifies the asset status as needed (e.g., operational, under maintenance, or decommissioned).   
9. The system applies the Algorithm to normalize and validate the updated asset condition.   
10. The system updates the asset status in the Data Cache and Data Archive.   
11. The Map Display is updated using the Michigan Geographic Framework to reflect the new asset status.   
12. The Icon Layer is adjusted to visually represent the updated asset condition.   
13. The system generates an Alert confirming the successful update of the asset status.   
14. The Administrator confirms the update and closes the process.   
  
Alternative Flow:   
1. If the asset identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the update.   
3. If the Administrator re-enters the identifier, the system repeats the validation and retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.   
5. If the updated asset status fails validation, the system displays an Alert and prompts the Administrator to correct the input.   
6. The Administrator corrects the input, and the system repeats the validation and update steps.   
7. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Michigan Geographic Framework Integration   
Use Case ID: UC-44   
Actors: Administrator, User Interface, Michigan Geographic Framework, Map Display, Icon Layer, Data Source, Data Cache, Data Archive, Algorithm, Alert, Traffic Data, Road Condition Data, Weather Data, Traveler Information   
  
Preconditions:   
- The system is connected to the Data Source for geographic and related data.   
- The Administrator has access to the User Interface.   
- The Michigan Geographic Framework is available for integration with the Map Display.   
- The Data Cache and Data Archive are ready to store integrated data.   
- The Algorithm is configured to process and align data with geographic coordinates.   
- The Map Display and Icon Layer are active for visualization.   
  
Postconditions:   
- The Michigan Geographic Framework is successfully integrated with the system's data visualization components.   
- The Map Display and Icon Layer reflect the integration and spatial alignment of the data.   
- The Data Cache and Data Archive are updated with the integrated data.   
- An Alert is generated to confirm the integration is complete.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Michigan Geographic Framework Integration" option.   
3. The system prompts the Administrator to choose the data types to be integrated (e.g., Traffic Data, Road Condition Data, or Weather Data).   
4. The Administrator selects the data types to integrate.   
5. The system retrieves the selected data from the Data Source or Data Cache.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The Algorithm aligns the data with the Michigan Geographic Framework based on spatial coordinates.   
8. The Map Display is updated using the Michigan Geographic Framework to show the integrated data in a geographic context.   
9. The Icon Layer is adjusted to visually represent the integrated data on the map.   
10. The system stores the integrated data in the Data Cache and Data Archive.   
11. The system generates an Alert to confirm the successful integration of the geographic framework with the data.   
12. The Administrator reviews the updated map and closes the process.   
  
Alternative Flow:   
1. If the selected data types are invalid or not supported by the Michigan Geographic Framework, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the data types or cancel the operation.   
3. If the Administrator re-enters the data types, the system repeats the data retrieval and integration steps.   
4. If the Administrator cancels, the system returns to the main menu without integrating the geographic framework.   
5. If the Michigan Geographic Framework is not available or the integration fails, the system generates an Alert and halts the process.  
  
Use Case Name: Modify Geo-referencing Settings   
Use Case ID: UC-45   
Actors: Administrator, User Interface, Michigan Geographic Framework, Map Display, Icon Layer, Algorithm, Data Cache, Data Archive, Alert   
  
Preconditions:   
- The Administrator has access to the User Interface.   
- The Michigan Geographic Framework is active and available for configuration.   
- The Map Display and Icon Layer are active for spatial data visualization.   
- The Data Cache and Data Archive are ready to store updated geo-referencing settings.   
- The Algorithm is configured to process and apply new geo-referencing parameters.   
  
Postconditions:   
- Geo-referencing settings are successfully modified in the Michigan Geographic Framework.   
- The Map Display and Icon Layer are updated to reflect the new geo-referencing configuration.   
- The updated settings are stored in the Data Cache and Data Archive.   
- An Alert is generated to confirm the modification of the geo-referencing settings.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Modify Geo-referencing Settings" option.   
3. The system displays the current geo-referencing parameters for review.   
4. The Administrator modifies the required settings (e.g., coordinate system, projection, or geographic scale).   
5. The system validates the modified settings using the Data Quality Check module.   
6. The system updates the Michigan Geographic Framework with the new settings.   
7. The Map Display and Icon Layer are refreshed to reflect the updated geo-referencing configuration.   
8. The system stores the updated settings in the Data Cache and Data Archive for future use.   
9. The system generates an Alert to confirm the successful modification of the geo-referencing settings.   
10. The Administrator confirms the changes and closes the process.   
  
Alternative Flow:   
1. If the modified geo-referencing settings are invalid or incompatible, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the settings or cancel the modification.   
3. If the Administrator re-enters the settings, the system repeats the validation and update steps.   
4. If the Administrator cancels, the system returns to the main menu without applying changes.   
5. If the Michigan Geographic Framework is not available or the update fails, the system generates an Alert and halts the process.  
  
Use Case Name: MI Drive Presentation Support   
Use Case ID: UC-46   
Actors: Administrator, User Interface, Traveler Information, Data Source, Data Cache, Data Archive, Presentation Method, Map Display, Icon Layer, Michigan Geographic Framework, Algorithm, Alert, Output Format   
  
Preconditions:   
- The system is connected to the Data Source for real-time and historical traveler information.   
- The Administrator has access to the User Interface.   
- The Presentation Method and Output Format modules are available for configuration.   
- The Data Cache and Data Archive are ready to store or update traveler information.   
- The Algorithm is configured to process and optimize data for presentation.   
- The Map Display and Icon Layer are active for visualizing traveler-related data.   
  
Postconditions:   
- Traveler information is successfully retrieved and presented in the selected Output Format.   
- The Presentation Method is updated to reflect the Administrator's preferences.   
- The Map Display and Icon Layer are updated to show relevant traveler data.   
- The Data Cache and Data Archive are updated with the latest traveler information.   
- An Alert is generated to confirm the successful presentation of traveler information.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "MI Drive Presentation Support" option.   
3. The system prompts the Administrator to specify the type of traveler information to present (e.g., traffic conditions, road closures, or weather impacts).   
4. The Administrator provides the required information type.   
5. The system retrieves the relevant Traveler Information from the Data Source or Data Cache.   
6. The system performs a Data Quality Check to ensure the data is accurate and consistent.   
7. The system applies the Algorithm to process and format the data for presentation.   
8. The system configures the Output Format and Presentation Method based on the Administrator's preferences.   
9. The system updates the Map Display using the Michigan Geographic Framework to show the relevant traveler information.   
10. The Icon Layer is adjusted to visually represent the traveler information (e.g., alerts, congestion, or road closures).   
11. The system stores the processed traveler information in the Data Cache and Data Archive.   
12. The system generates an Alert to inform the Administrator that the presentation is ready for review.   
13. The Administrator confirms the presentation and closes the process.   
  
Alternative Flow:   
1. If the specified traveler information type is invalid or not supported, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the type or cancel the operation.   
3. If the Administrator re-enters the type, the system repeats the data retrieval and presentation steps.   
4. If the Administrator cancels, the system returns to the main menu without presenting traveler information.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: View Traveler Information   
Use Case ID: UC-47   
Actors: Administrator, User Interface, Traveler Information, Data Source, Data Cache, Map Display, Icon Layer, Michigan Geographic Framework, Data Quality Check, Alert, Traffic Data, Road Condition Data, Weather Data, Algorithm   
  
Preconditions:   
- The system is connected to the Data Source for real-time and historical traveler-related data.   
- The Administrator has access to the User Interface.   
- The Data Cache is available for quick retrieval of traveler information.   
- The Map Display and Icon Layer are active for visualizing traveler-related data.   
- The Data Quality Check module is active to ensure data accuracy.   
- The Algorithm is configured to process and format traveler information.   
  
Postconditions:   
- Traveler information is successfully retrieved and displayed to the Administrator.   
- The Map Display and Icon Layer reflect the current traveler-related conditions.   
- The Data Cache is updated with the retrieved traveler information for quick access.   
- An Alert is generated to confirm the successful viewing of traveler information.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "View Traveler Information" option.   
3. The system prompts the Administrator to specify the geographic area and time frame for the query.   
4. The Administrator provides the required parameters.   
5. The system validates the input using the Data Quality Check module.   
6. The system retrieves Traveler Information, including Traffic Data, Road Condition Data, and Weather Data, from the Data Source or Data Cache.   
7. The Algorithm processes the data to ensure it is ready for display.   
8. The system formats the data using the selected Output Format and Presentation Method.   
9. The Map Display is updated using the Michigan Geographic Framework to show the traveler information.   
10. The Icon Layer is adjusted to represent relevant traveler data (e.g., traffic congestion, road closures, or weather conditions).   
11. The system stores the retrieved traveler information in the Data Cache for future reference.   
12. The system generates an Alert to confirm the successful retrieval and display of traveler information.   
13. The Administrator reviews the traveler information and closes the process.   
  
Alternative Flow:   
1. If the geographic area or time frame is invalid, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the validation and data retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying traveler information.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Update Traveler Notification   
Use Case ID: UC-48   
Actors: Administrator, User Interface, Traveler Information, Data Cache, Data Archive, Alert, Map Display, Icon Layer, Michigan Geographic Framework, Data Source, Algorithm   
  
Preconditions:   
- The system is connected to the Data Source for real-time traveler information and notification settings.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive are available for storing and updating traveler notification data.   
- The Map Display and Icon Layer are active for visualizing traveler notification updates.   
- The Algorithm is configured to process and validate notification content.   
  
Postconditions:   
- Traveler notification is successfully updated in the system.   
- The Data Cache and Data Archive are updated with the latest notification data.   
- The Map Display and Icon Layer reflect the updated notification as needed.   
- An Alert is generated to confirm the successful update.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Update Traveler Notification" option.   
3. The system prompts the Administrator to input the notification identifier (e.g., location or message ID).   
4. The Administrator enters the required identifier.   
5. The system validates the identifier using the Data Quality Check module.   
6. The system retrieves the existing traveler notification data from the Data Source.   
7. The system displays the current notification details to the Administrator.   
8. The Administrator modifies the notification content (e.g., message text, severity, or display duration).   
9. The system applies the Algorithm to normalize and validate the updated notification.   
10. The system updates the notification in the Data Cache and Data Archive.   
11. The Map Display is updated using the Michigan Geographic Framework to reflect the new notification location.   
12. The Icon Layer is adjusted to visually represent the updated notification.   
13. The system generates an Alert confirming the successful update of the traveler notification.   
14. The Administrator confirms the update and closes the process.   
  
Alternative Flow:   
1. If the notification identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.   
5. If the updated notification fails validation, the system displays an Alert and prompts the Administrator to correct the input.   
6. The Administrator corrects the input, and the system repeats the validation and update steps.   
7. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Delete Traveler Record   
Use Case ID: UC-49   
Actors: Administrator, User Interface, Traveler Information, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Data Quality Check, Alert   
  
Preconditions:   
- The system is connected to the Data Source for traveler information.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive contain the traveler record to be deleted.   
- The Map Display and Icon Layer are active and reflect current traveler data.   
- The Data Quality Check module is available to validate input data.   
  
Postconditions:   
- The traveler record is successfully deleted from the system.   
- The Data Cache and Data Archive are updated to remove the traveler information.   
- The Map Display and Icon Layer are updated to remove the deleted traveler's representation.   
- An Alert is generated to confirm the deletion.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Delete Traveler Record" option.   
3. The system prompts the Administrator to input the traveler identifier (e.g., user ID or traveler profile).   
4. The Administrator enters the required traveler identifier.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Source to confirm the existence of the traveler record.   
7. If the record exists, the system proceeds to delete it.   
8. The system removes the traveler data from the Data Cache and Data Archive.   
9. The Map Display is updated using the Michigan Geographic Framework to remove the traveler's location.   
10. The Icon Layer is adjusted to remove the visual representation of the traveler.   
11. The system generates an Alert confirming the successful deletion of the traveler record.   
12. The Administrator confirms the deletion and closes the process.   
  
Alternative Flow:   
1. If the traveler identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and deletion steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.   
5. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Manage DataRule   
Use Case ID: UC-50   
Actors: Administrator, User Interface, DataRule, Data Source, Data Cache, Data Archive, Algorithm, Data Quality Check, Alert   
  
Preconditions:   
- The system is connected to the Data Source for data rule definitions and configurations.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive are available for storing or retrieving data rules.   
- The DataRule module is active and ready for management (add, update, or remove).   
- The Algorithm is configured to apply or process data rules.   
- The Map Display and Icon Layer are active for visualization if the DataRule affects spatial data.   
  
Postconditions:   
- The DataRule is successfully managed (added, updated, or removed) in the system.   
- The Data Cache and Data Archive are updated with the latest data rule configuration.   
- The Algorithm is updated to reflect any changes to the DataRule.   
- The Map Display and Icon Layer are updated if the DataRule affects data visualization.   
- An Alert is generated to confirm the successful management of the DataRule.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Manage DataRule" option.   
3. The system displays a list of available actions: "Add DataRule," "Update DataRule," or "Remove DataRule."   
4. The Administrator selects "Add DataRule."   
5. The system prompts the Administrator to input the new DataRule details (e.g., rule name, condition logic, data fields involved, priority, and associated data formats).   
6. The Administrator enters the required DataRule parameters.   
7. The system validates the input using the Data Quality Check module to ensure the rule is logically consistent and compatible with the system.   
8. The system checks if the DataRule already exists in the Data Archive or Data Cache.   
9. If the DataRule is not already defined, the system proceeds to store it.   
10. The system adds the new DataRule to the Data Cache and Data Archive.   
11. The Algorithm is updated to include the new rule in its processing logic.   
12. If the rule affects spatial data visualization, the Map Display and Icon Layer are updated to reflect the rule’s impact.   
13. The system generates an Alert confirming the successful addition of the DataRule.   
14. The Administrator confirms the action and closes the process.   
  
Alternative Flow:   
1. If the DataRule already exists and the Administrator selects "Add DataRule," the system displays an Alert indicating the duplication.   
2. The Administrator can choose to update the existing DataRule or cancel the operation.   
3. If the Administrator chooses to update, the system allows the input of new or modified DataRule details.   
4. The system validates the updated DataRule and stores it in the Data Cache and Data Archive.   
5. The Algorithm is updated to reflect the modified DataRule.   
6. The Map Display and Icon Layer are updated if the DataRule change affects visual representation.   
7. The system generates an Alert confirming the successful update of the DataRule.   
8. If the Administrator selects "Remove DataRule," the system prompts for the DataRule identifier (e.g., rule name or ID).   
9. The Administrator enters the DataRule identifier.   
10. The system validates the identifier and queries the Data Archive to confirm the rule's existence.   
11. If the rule exists, the system proceeds to remove it from the Data Cache and Data Archive.   
12. The Algorithm is updated to exclude the removed rule from its processing logic.   
13. The Map Display and Icon Layer are updated if the removed rule affected visualization.   
14. The system generates an Alert confirming the successful removal of the DataRule.   
15. If the specified DataRule identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
16. The Administrator can choose to re-enter the identifier or cancel the operation.   
17. If the Administrator re-enters the identifier, the system repeats the validation and management steps.   
18. If the Administrator cancels, the system returns to the main menu without making any changes.  
  
Use Case Name: Manage Traveler Record   
Use Case ID: UC-51   
Actors: Administrator, User Interface, Traveler Record, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Data Quality Check, Alert   
  
Preconditions:   
- The system is connected to the Data Source for traveler-related data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive contain the traveler records for quick retrieval or modification.   
- The Map Display and Icon Layer are active and ready to visualize traveler data if applicable.   
- The Data Quality Check module is available to validate the accuracy and consistency of the traveler record.   
  
Postconditions:   
- The traveler record is successfully managed (added, updated, or removed) in the system.   
- The Data Cache and Data Archive are updated with the latest traveler information.   
- The Map Display and Icon Layer are updated if the traveler record involves location data.   
- An Alert is generated to confirm the successful management of the traveler record.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Manage Traveler Record" option.   
3. The system displays a list of available actions: "Add Traveler Record," "Update Traveler Record," or "Remove Traveler Record."   
4. The Administrator selects "Add Traveler Record."   
5. The system prompts the Administrator to input traveler information (e.g., name, contact details, travel preferences, or associated vehicle information).   
6. The Administrator enters the required traveler details.   
7. The system validates the input using the Data Quality Check module to ensure data accuracy and completeness.   
8. The system checks the Data Source to confirm the traveler does not already exist in the system.   
9. If the traveler is not already registered, the system proceeds to store the new record.   
10. The system saves the new traveler record in the Data Cache and Data Archive.   
11. If the traveler is associated with location or vehicle data, the Map Display and Icon Layer are updated to reflect the new traveler information.   
12. The system generates an Alert confirming the successful addition of the traveler record.   
13. The Administrator confirms the action and closes the process.   
  
Alternative Flow:   
1. If the traveler is already registered and the Administrator selects "Add Traveler Record," the system displays an Alert indicating the duplication.   
2. The Administrator can choose to update the existing traveler record or cancel the operation.   
3. If the Administrator chooses to update, the system allows the input of new or modified traveler details.   
4. The system validates the updated data and stores it in the Data Cache and Data Archive.   
5. If the traveler is associated with location or vehicle data, the Map Display and Icon Layer are updated to reflect the changes.   
6. The system generates an Alert confirming the successful update of the traveler record.   
7. If the Administrator chooses to cancel, the system returns to the main menu without making any changes.   
8. If the Administrator selects "Remove Traveler Record," the system prompts for the traveler identifier (e.g., name, user ID, or contact information).   
9. The Administrator enters the traveler identifier.   
10. The system validates the input and queries the Data Source to confirm the existence of the traveler record.   
11. If the record exists, the system proceeds to delete it from the Data Cache and Data Archive.   
12. If the traveler is associated with location or vehicle data, the Map Display and Icon Layer are updated to remove the traveler's representation.   
13. The system generates an Alert confirming the successful removal of the traveler record.   
14. If the specified traveler identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
15. The Administrator can choose to re-enter the identifier or cancel the operation.   
16. If the Administrator re-enters the identifier, the system repeats the validation and management steps.   
17. If the Administrator cancels, the system returns to the main menu without making any changes.   
18. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Manage AssetStatus   
Use Case ID: UC-52   
Actors: Administrator, User Interface, AssetStatus, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Data Quality Check, Alert   
  
Preconditions:   
- The system is connected to the Data Source for asset status data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive are available for storing or retrieving asset status records.   
- The Map Display and Icon Layer are active for visualizing asset status updates.   
- The Data Quality Check module is active to ensure data integrity and consistency.   
- The AssetStatus module is configured to support the management of asset states.   
  
Postconditions:   
- The asset status is successfully managed (added, updated, or removed) in the system.   
- The Data Cache and Data Archive are updated with the latest asset status information.   
- The Map Display and Icon Layer are updated to reflect the current asset status.   
- An Alert is generated to confirm the successful management of the asset status.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Manage AssetStatus" option.   
3. The system displays a list of available actions: "Add AssetStatus," "Update AssetStatus," or "Remove AssetStatus."   
4. The Administrator selects "Add AssetStatus."   
5. The system prompts the Administrator to input the asset identifier (e.g., asset ID, location, or type) and the new asset status (e.g., operational, under maintenance, or non-functional).   
6. The Administrator enters the required asset identifier and status.   
7. The system validates the input using the Data Quality Check module to ensure accuracy and logical consistency.   
8. The system queries the Data Source to confirm that the asset exists in the system.   
9. If the asset exists, the system proceeds to store the new AssetStatus.   
10. The system updates the Data Cache and Data Archive with the new asset status.   
11. The Map Display is updated using the Michigan Geographic Framework to show the asset's updated status.   
12. The Icon Layer is adjusted to visually represent the new asset status (e.g., color-coded icons for operational vs. non-functional assets).   
13. The system generates an Alert to confirm the successful addition or update of the AssetStatus.   
14. The Administrator confirms the action and closes the process.   
  
Alternative Flow:   
1. If the asset identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the identifier or cancel the operation.   
3. If the Administrator re-enters the identifier, the system repeats the validation and management steps.   
4. If the Administrator selects "Update AssetStatus," the system prompts for the asset identifier and the new status.   
5. The system validates the identifier and updates the asset status in the Data Cache and Data Archive.   
6. The Map Display and Icon Layer are updated to reflect the new status.   
7. If the Administrator selects "Remove AssetStatus," the system prompts for the asset identifier.   
8. The system validates the identifier and removes the associated asset status record from the Data Cache and Data Archive.   
9. The Map Display and Icon Layer are updated to remove or reset the asset's visual representation.   
10. If the updated or removed status fails validation, the system displays an Alert and prompts the Administrator to correct the input.   
11. The Administrator corrects the input, and the system repeats the validation and update steps.   
12. If the Data Source is unreachable or returns corrupted data, the system generates an Alert and halts the process.  
  
Use Case Name: Configure Map Layer Setting   
Use Case ID: UC-53   
Actors: Administrator, User Interface, Map Layer Setting, Map Display, Icon Layer, Michigan Geographic Framework, Data Cache, Alert   
  
Preconditions:   
- The Administrator has access to the User Interface.   
- The Map Display and Icon Layer are active and ready for configuration.   
- The Map Layer Setting module is available for defining layer visibility, styling, and interaction rules.   
- The Michigan Geographic Framework is active and supports spatial configuration.   
- The Data Cache is ready to store the updated map layer settings for quick retrieval.   
- The system is connected to the Data Source for reference to existing data layers.   
  
Postconditions:   
- The Map Layer Setting is successfully configured according to the Administrator's preferences.   
- The Map Display and Icon Layer are updated to reflect the new settings (e.g., visibility, color, or priority of layers).   
- The Data Cache is updated with the new map layer settings for future use.   
- An Alert is generated to confirm the successful configuration of the map layer settings.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Configure Map Layer Setting" option.   
3. The system displays the current map layer settings for review.   
4. The Administrator modifies the settings (e.g., enables or disables specific layers, adjusts layer colors, or sets layer priority).   
5. The system validates the modified settings using the Data Quality Check module.   
6. The system updates the Map Layer Setting module with the new configuration.   
7. The Map Display and Icon Layer are refreshed to reflect the updated settings.   
8. The system stores the new map layer settings in the Data Cache for quick access in future sessions.   
9. The system generates an Alert to inform the Administrator that the configuration is complete.   
10. The Administrator confirms the configuration and closes the process.   
  
Alternative Flow:   
1. If the modified map layer settings are invalid or inconsistent, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-enter the settings or cancel the configuration.   
3. If the Administrator re-enters the settings, the system repeats the validation and configuration steps.   
4. If the Administrator cancels, the system returns to the main menu without applying changes.   
5. If the Map Display or Icon Layer is not active or fails to apply the new settings, the system generates an Alert and halts the process.  
  
Use Case Name: View Alert History   
Use Case ID: UC-54   
Actors: Administrator, User Interface, Alert History, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Michigan Geographic Framework, Data Quality Check   
  
Preconditions:   
- The system is connected to the Data Source for historical alert data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive contain the Alert History records.   
- The Map Display and Icon Layer are active and configured for visualizing alert locations.   
- The Michigan Geographic Framework is available for spatial alignment of historical alerts.   
- The Data Quality Check module is active to ensure the integrity of the alert data.   
  
Postconditions:   
- The Alert History is successfully retrieved and displayed to the Administrator.   
- The Data Cache is updated with the retrieved alert history for quick access.   
- The Map Display and Icon Layer reflect the locations and context of historical alerts.   
- An Alert is generated to confirm the successful retrieval and display of the Alert History.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "View Alert History" option.   
3. The system prompts the Administrator to specify a time range, severity level, or alert type to filter the history.   
4. The Administrator provides the required filter parameters.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Archive to retrieve the Alert History based on the specified filters.   
7. The system retrieves relevant Alert records, including details such as time, location, type, and severity.   
8. The Data Cache is updated with the retrieved Alert History for quick access in future queries.   
9. If the alert history includes spatial information, the Map Display is updated using the Michigan Geographic Framework to show the locations of historical alerts.   
10. The Icon Layer is adjusted to visually represent the alerts based on their type and severity.   
11. The system generates an Alert to confirm the successful retrieval and display of the Alert History.   
12. The Administrator reviews the displayed Alert History and closes the process.   
  
Alternative Flow:   
1. If the specified filter parameters are invalid or not supported, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the validation and retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying the Alert History.   
5. If the Data Archive is unreachable or returns corrupted Alert History data, the system generates an Alert and halts the process.  
  
Use Case Name: View AlgorithmExecutionLog   
Use Case ID: UC-55   
Actors: Administrator, User Interface, AlgorithmExecutionLog, Data Archive, Data Cache, Alert, Data Quality Check, Algorithm, Presentation Method, Output Format, Map Display   
  
Preconditions:   
- The system is connected to the Data Archive for storing and retrieving Algorithm Execution Logs.   
- The Administrator has access to the User Interface.   
- The Data Cache is available for quick retrieval of recent log entries.   
- The AlgorithmExecutionLog module is active and contains execution history.   
- The Presentation Method and Output Format modules are configured to display log data.   
- The Map Display is active if the logs involve location-based algorithm execution.   
  
Postconditions:   
- The AlgorithmExecutionLog is successfully retrieved and displayed to the Administrator.   
- The Data Cache is updated with the retrieved log entries for quick access.   
- The Map Display is updated if the logs involve spatial data execution.   
- An Alert is generated to confirm the successful retrieval of the Algorithm Execution Log.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "View AlgorithmExecutionLog" option.   
3. The system prompts the Administrator to specify the time range, algorithm name, or execution status to filter the logs.   
4. The Administrator provides the required filter criteria.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Archive to retrieve the AlgorithmExecutionLog based on the specified criteria.   
7. The system retrieves the relevant Algorithm Execution Log entries, including details such as execution time, input parameters, output, and status.   
8. The system processes and formats the log data using the selected Output Format and Presentation Method.   
9. The Data Cache is updated with the retrieved AlgorithmExecutionLog data for quick access in future sessions.   
10. If the logs involve spatial processing, the Map Display is updated to show the relevant geographic context.   
11. The system generates an Alert to confirm the successful retrieval of the Algorithm Execution Log.   
12. The Administrator reviews the log data and closes the process.   
  
Alternative Flow:   
1. If the specified filter criteria are invalid or not supported, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the criteria or cancel the operation.   
3. If the Administrator re-enters the criteria, the system repeats the validation and retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying the Algorithm Execution Log.   
5. If the Data Archive is unreachable or returns corrupted log data, the system generates an Alert and halts the process.  
  
Use Case Name: Format DataFormatMapping   
Use Case ID: UC-56   
Actors: Administrator, User Interface, DataFormatMapping, Data Source, Data Cache, Data Archive, Algorithm, Output Format, Presentation Method, Data Quality Check, Map Display, Michigan Geographic Framework, Alert   
  
Preconditions:   
- The system is connected to the Data Source for raw or structured data requiring mapping to specific formats.   
- The Administrator has access to the User Interface.   
- The DataFormatMapping module is available for defining how data fields are mapped to standardized formats (e.g., SAE J2354 or TMDD).   
- The Data Cache and Data Archive are ready to store updated mapping configurations and associated data.   
- The Algorithm is configured to apply the mapping rules during data transformation.   
- The Map Display is active if the DataFormatMapping affects spatial or visual data representation.   
- The Data Quality Check module is available to ensure the integrity of the mapping process.   
  
Postconditions:   
- The DataFormatMapping is successfully configured or modified in the system.   
- The Data Cache and Data Archive are updated to store the current mapping rules.   
- The Algorithm is updated to apply the new or modified DataFormatMapping.   
- The Map Display is updated if the mapping change affects spatial data visualization.   
- An Alert is generated to confirm the successful formatting or mapping process.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Format DataFormatMapping" option.   
3. The system displays the current DataFormatMapping rules for reference.   
4. The Administrator modifies or adds new mapping rules to align data fields with the desired Output Format (e.g., SAE J2354 or TMDD).   
5. The system validates the new or modified mapping rules using the Data Quality Check module.   
6. The system updates the DataFormatMapping configuration with the new rules.   
7. The system stores the updated DataFormatMapping in the Data Cache and Data Archive for future use.   
8. The Algorithm is updated to use the new mapping rules during data processing.   
9. If the mapping affects spatial data, the Map Display is refreshed to reflect the new formatting.   
10. The system generates an Alert to confirm the successful formatting of the DataFormatMapping.   
11. The Administrator confirms the formatting and closes the process.   
  
Alternative Flow:   
1. If the mapping rules are invalid or incompatible with the target Output Format, the system displays an Alert indicating the failure.   
2. The Administrator can choose to correct the mapping rules or cancel the formatting process.   
3. If the Administrator corrects the rules, the system repeats the validation and mapping steps.   
4. If the Administrator cancels, the system returns to the main menu without applying changes.   
5. If the Data Source is unreachable or returns corrupted data during mapping validation, the system generates an Alert and halts the process.  
  
Use Case Name: View Incident History   
Use Case ID: UC-57   
Actors: Administrator, User Interface, Incident, Data Archive, Data Cache, Map Display, Icon Layer, Michigan Geographic Framework, Data Source, Data Quality Check, Alert   
  
Preconditions:   
- The system is connected to the Data Source for incident-related data.   
- The Administrator has access to the User Interface.   
- The Data Archive and Data Cache contain incident history records.   
- The Map Display and Icon Layer are active and ready to visualize historical incident locations.   
- The Michigan Geographic Framework is available for spatial alignment.   
- The Data Quality Check module is active to ensure the integrity and consistency of the incident history data.   
  
Postconditions:   
- The incident history is successfully retrieved and displayed to the Administrator.   
- The Data Cache is updated with the retrieved incident history for quick access.   
- The Map Display and Icon Layer reflect the locations and details of historical incidents.   
- An Alert is generated to confirm the successful retrieval of the incident history.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "View Incident History" option.   
3. The system prompts the Administrator to specify a time range, geographic area, or incident type to filter the history.   
4. The Administrator provides the required filter parameters.   
5. The system validates the input using the Data Quality Check module.   
6. The system queries the Data Archive to retrieve the incident history based on the specified filters.   
7. The system retrieves relevant incident records, including details such as location, time of occurrence, type, and severity.   
8. The Data Cache is updated with the retrieved incident history for quick reference in future queries.   
9. If the incident history includes spatial information, the Map Display is updated using the Michigan Geographic Framework to show the locations of past incidents.   
10. The Icon Layer is adjusted to visually represent historical incidents on the map.   
11. The system generates an Alert to confirm the successful retrieval of the incident history.   
12. The Administrator reviews the incident history and closes the process.   
  
Alternative Flow:   
1. If the specified filter parameters are invalid or not supported, the system displays an Alert indicating the error.   
2. The Administrator can choose to re-enter the parameters or cancel the operation.   
3. If the Administrator re-enters the parameters, the system repeats the validation and retrieval steps.   
4. If the Administrator cancels, the system returns to the main menu without retrieving or displaying the incident history.   
5. If the Data Archive is unreachable or returns corrupted incident data, the system generates an Alert and halts the process.  
  
Use Case Name: Manage Vehicle Registration   
Use Case ID: UC-58   
Actors: Administrator, User Interface, Vehicle Registration, Data Source, Data Cache, Data Archive, Map Display, Icon Layer, Data Quality Check, Michigan Geographic Framework, Alert   
  
Preconditions:   
- The system is connected to the Data Source for vehicle registration data.   
- The Administrator has access to the User Interface.   
- The Data Cache and Data Archive are available for storing and retrieving vehicle registration records.   
- The Map Display and Icon Layer are active and ready to visualize vehicle-related data.   
- The Data Quality Check module is active to ensure the integrity and accuracy of registration information.   
- The Michigan Geographic Framework is available for spatial representation of vehicle data.   
- The Vehicle Registration module is configured to support management actions such as add, update, or remove.   
  
Postconditions:   
- The vehicle registration is successfully managed (added, updated, or removed) in the system.   
- The Data Cache and Data Archive are updated to reflect the latest vehicle registration data.   
- The Map Display and Icon Layer are updated to show changes in vehicle registration status.   
- An Alert is generated to confirm the successful management of the vehicle registration.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Manage Vehicle Registration" option.   
3. The system displays a list of available actions: "Add Vehicle Registration," "Update Vehicle Registration," or "Remove Vehicle Registration."   
4. The Administrator selects "Add Vehicle Registration."   
5. The system prompts the Administrator to input vehicle details (e.g., make, model, license plate, vehicle identification number, and registration status).   
6. The Administrator enters the required vehicle registration information.   
7. The system validates the input using the Data Quality Check module to ensure data accuracy and completeness.   
8. The system checks the Data Source to confirm that the vehicle is not already registered.   
9. If the vehicle is not registered, the system proceeds to add the registration record.   
10. The system stores the new vehicle registration data in the Data Cache and Data Archive.   
11. The Map Display is updated using the Michigan Geographic Framework to show the newly registered vehicle's location.   
12. The Icon Layer is adjusted to represent the registered vehicle.   
13. The system generates an Alert to confirm the successful addition of the vehicle registration.   
14. The Administrator confirms the action and closes the process.   
  
Alternative Flow:   
1. If the vehicle is already registered and the Administrator selects "Add Vehicle Registration," the system displays an Alert indicating the duplication.   
2. The Administrator can choose to update the existing registration or cancel the operation.   
3. If the Administrator chooses to update, the system allows the input of new or modified registration details.   
4. The system validates the updated information and stores it in the Data Cache and Data Archive.   
5. The Map Display and Icon Layer are updated to reflect the changes in the vehicle's registration status.   
6. If the Administrator selects "Remove Vehicle Registration," the system prompts for the vehicle identification number (VIN) or license plate.   
7. The Administrator enters the required identifier.   
8. The system validates the identifier and queries the Data Source to confirm the existence of the registration.   
9. If the registration exists, the system proceeds to delete it.   
10. The system removes the vehicle registration data from the Data Cache and Data Archive.   
11. The Map Display and Icon Layer are updated to remove the vehicle's representation.   
12. The system generates an Alert to confirm the successful removal of the vehicle registration.   
13. If the specified identifier is invalid or not found in the Data Source, the system displays an Alert indicating the failure.   
14. The Administrator can choose to re-enter the identifier or cancel the operation.   
15. If the Administrator re-enters the identifier, the system repeats the validation and management steps.   
16. If the Administrator cancels, the system returns to the main menu without making any changes.   
17. If the Data Source is unreachable or returns corrupted registration data, the system generates an Alert and halts the process.  
  
Use Case Name: Delete Icon Layer   
Use Case ID: UC-59   
Actors: Administrator, User Interface, Map Display, Icon Layer, Data Cache, Alert, Michigan Geographic Framework   
  
Preconditions:   
- The system is connected to the Data Source for spatial and visual configuration data.   
- The Administrator has access to the User Interface.   
- The Map Display and Icon Layer are active and contain visual elements.   
- The Data Cache stores the current state of the Icon Layer.   
- The Michigan Geographic Framework is available for spatial alignment.   
- The Icon Layer to be deleted is valid and exists in the system.   
  
Postconditions:   
- The selected Icon Layer is successfully deleted from the Map Display.   
- The Data Cache is updated to reflect the removal of the Icon Layer.   
- The Michigan Geographic Framework remains unaffected by the deletion.   
- An Alert is generated to confirm the successful deletion of the Icon Layer.   
  
Main Flow:   
1. The Administrator opens the User Interface.   
2. The Administrator selects the "Delete Icon Layer" option.   
3. The system prompts the Administrator to select the specific Icon Layer to be deleted.   
4. The Administrator selects the Icon Layer from the available options.   
5. The system validates the selected Icon Layer to confirm its existence.   
6. The system proceeds to delete the selected Icon Layer from the Map Display.   
7. The system updates the Data Cache to remove the configuration related to the deleted Icon Layer.   
8. The Map Display is refreshed to no longer include the deleted Icon Layer.   
9. The system generates an Alert to confirm the successful deletion of the Icon Layer.   
10. The Administrator confirms the deletion and closes the process.   
  
Alternative Flow:   
1. If the selected Icon Layer is invalid or not found in the system, the system displays an Alert indicating the failure.   
2. The Administrator can choose to re-select the Icon Layer or cancel the operation.   
3. If the Administrator re-selects the Icon Layer, the system repeats the validation and deletion steps.   
4. If the Administrator cancels, the system returns to the main menu without making any changes.   
5. If the Data Cache is unreachable or corrupted during the update, the system generates an Alert and halts the process.