

Customer	Municipal Government of Palmira
User	Municipal authorities, security operators, citizens and transport entities
Functional requirements	<p>RF1 – Writing JSON Files (Routes Registry)</p> <p>RF2- Reading JSON files (routes)</p> <p>RF3 – Route Management</p> <p>RF4 – Search for the Best Route</p> <p>RF5 – Writing JSON Files (Incident Log)</p> <p>RF6- Reading JSON files (incidents)</p> <p>RF7 – Incident Management</p> <p>RF8 – Incident Search</p> <p>RF9 – Writing JSON Files (Passenger Registration)</p> <p>RF10- Reading JSON files (passengers)</p> <p>RF11 – Writing JSON Files (Driver Registration)</p> <p>RF12- Reading JSON files (drivers)</p> <p>RF13 – Driver Search</p> <p>RF14 – Report Generation</p> <p>RF15- Monitoring of emergency vehicles and security patrols</p> <p>RF16- Tracking of emergency vehicles and security patrols</p> <p>RF17- Real-time information on traffic and transport</p> <p>RF18- Optimization of public and private transport routes</p> <p>RF19- Statistics generation</p> <p>RF20- Notification to citizens about incidents, alternative routes and safety recommendations</p>

Context of the problem	<p><i>The city of Palmira (Valle del Cauca, Colombia) faces serious challenges in security and mobility due to rapid economic and population growth. The increase in crime, robberies, accidents and traffic congestion have generated the need to create a SGMMS:</i></p> <p><i>The purpose of SGMMS is to optimize security and mobility, ensuring the provision of technologies that enable real-time monitoring, facilitating rapid responses to incidents, providing up-to-date information on traffic status and transport routes, and generating reports and statistics to support decision-making by authorities.</i></p>
Non-functional requirements	<p>RNF1 – The system must process operations and generate reports in a reasonable time</p> <p>RNF2 – The application must display clear, user-friendly and understandable messages</p> <p>RNF3 - Maintainability</p> <p>RNF4- Usability</p> <p>RNF5- Scalability</p> <p>RNF6- System performance</p> <p>RNF7- Information Security</p> <p>RNF8- System availability</p>

Identifier and name	
---------------------	--

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	RF1 – Writing JSON Files (Route Registry)		
Summary	<i>The system must allow writing JSON files containing information about transport routes. The data must include: route ID, distance (km), estimated time (minutes), start point and end point. And the system must also allow reading this file which is to load the data already recorded.</i>		
Tickets	Entry name	Data type	Condition valid values
	Route ID	String	[String of numbers from 0 to 9]
	Distance	String	[It must be a positive number, and must additionally be expressed in km]
	Estimated travel time	int	[It must be a positive number, and must additionally be expressed in minutes]
	Start	String	[It must be a valid reference point within the city of Palmira]
	End	String	[It must be a valid reference point for the city of Palmyra]
Result or Postcondition	Data is loaded into a custom linked list structure. When writing, a properly formatted JSON file is generated or updated at the specified location, and a text (string) representation of the JSON file's contents is displayed in the console.		
Departures	Name of output	Data type	Format
	Route	String	<i>"routes : [</i> <i>{Id: R1, Distance: 10,</i> <i>Time: 15, Start: A,</i> <i>End: B}</i>

			<pre>{id: R2, distance: 10, time: 8, start: C, end: D}]</pre>
--	--	--	---

Identifier and name	RF2 – Reading JSON Files (Routes Registry)		
Summary	<i>The system must allow reading JSON files with route information.</i>		
Tickets	Entry name	Data type	Condition valid values
Result or Postcondition	Data is loaded into memory and stored in a linked list.		
Departures	Name of output	Data type	Format
	List of routes	List	<pre>“ Data in linked list or JSON structure “</pre>

Identifier and name	RF3 – Route Management
Summary	<p>The system must sort the linked list of routes (obtained in RF1) using two criteria: one primary and one secondary. The user will select the primary sorting criterion, choosing between 1 (distance) or 2 (time).</p> <p>For example, if "distance" is selected as the primary criterion, the system will organize the routes from shortest to longest distance and, among those with the same distance, it will prioritize those with the</p>

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	shortest estimated travel time. Similarly, if "time" is chosen as the primary criterion, the routes will be ordered from shortest to longest time, and in case of a tie, "distance" will be used as the secondary criterion.		
Tickets	Entry name	Data type	Condition valid values
Result or Postcondition	The list of routes is sorted in ascending order by first applying the selected primary criterion (1: distance or 2: time) and then the other criterion as secondary to refine the order (for example, for criterion 1: between routes with equal distance, it is sorted by time from shortest to longest).		
Departures	Name of output	Data type	Format
	Ordered list	List	<i>[Linked list of routes ordered by primary and secondary criteria. Optionally, can be displayed in JSON (text) format for console viewing.]</i>

Identifier and name	RF4 – Finding the best route
Summary	<p>The system will allow finding the best route based on criteria defined by the user.</p> <p>The user will select the primary sorting criterion, choosing between 1 (distance) or 2 (time). For example, if "distance" is selected as the primary criterion, the system will organize the routes from shortest to longest distance and, among those with the same distance, will prioritize those with the shortest estimated travel time. Similarly, if "time" is chosen as the primary criterion, the routes will be ordered from shortest to longest time,</p>

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	and in case of a tie, "distance" will be used as the secondary criterion.		
Tickets	Entry name	Data type	Condition valid values
	List of routes	List	<i>[Linked list generated from the data read in the RF1 (each route with attributes: ID, distance, time, start point and end point).]</i>
	Criterion	int	<i>[Integer value: 1 to sort by distance or 2 to sort by time.]</i>
Result or Postcondition	The system must show the user the best route taking into account their sorting criteria.		
Departures	Name of output	Data type	Format
	Ordered list	List	<i>[Linked list of routes ordered by primary and secondary criteria. Optionally, can be displayed in JSON (text) format for console viewing.]</i>

Identifier and name	RF5 – Writing JSON files (Incident log)
Summary	<p>The system must allow for recording and storing information on security incidents.</p> <p>The user must enter the incident ID, the type of incident, the location where it occurred, the date and time of the incident, and its resolution status.</p>

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

Tickets	Entry name	Data type	Condition valid values
	Incident ID	String	<i>[User must enter ID which is a string from 0 to 9]</i>
	Type of incident	String	<i>[The user must enter what type of incident it is. (Ex.: theft, accident, fire, etc.)]</i>
	Location	String	<i>[The user must enter the address or coordinates of where the incident occurs. (Ex.: Carrera 45 #2, Alcázares neighborhood)]</i>
	Date/Time	String	<i>[Must be YYYY-MM-DD and HH:MM:SS format]</i>
	State	String	<i>[The user must define whether the status is pending, in process or resolved]</i>
Result or Postcondition	The system should launch detailed information about the incident		
Departures	Name of output	Data type	Format
	Error message	String	The incident registration was not successful
	Incident information	String	<i>"The incident was recorded with ID: 1, type of incident: robbery, location of incident: Alcaceres neighborhood, date/time: YYYY-MM-DD and HH:MM:SS."</i>

Identifier and name	RF6 – Reading JSON files (Incident log)		
Summary	The system must allow reading and storing JSON files with information about security incidents.		
Tickets	Entry name	Data type	Condition valid values
Result or Postcondition	Data is loaded into memory and stored in a linked list.		
Departures	Name of output	Data type	Format
	List of incidents	List	A linked list containing all incidents read from the JSON file, including their ID, type, location, date/time, and status.

Identifier and name	RF7 – Incident Ordering		
Summary	The system should sort the incidents according to their date/time of report, from the most recent to the oldest. The first thing it should do is check if there are any recorded incidents, then it sorts them and generates an ordered list with the original structure.		
Tickets	Entry name	Data type	Condition valid values
Result or Postcondition	The system should launch detailed information about the incident		

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	Name of output	Data type	Format
Departures	Ordered list	List	<i>[E.G.: "Id: 2, type: theft, location: 5th street #10, date_time: 2025-03-02_14:30:00, status: pending"]</i>

Identifier and name	RF8 – Incident Search		
Summary	The system must allow searching for recorded incidents, taking into account that the data must match the entered criteria. The system verifies that the incident ID evidently exists and that the information it contains is accurate.		
Tickets	Entry name	Data type	Condition valid values
	Incident ID	String	<i>[String of numbers from 0 to 9]</i>
Result or Postcondition	The system will display complete information about the incident being searched for.		
Departures	Name of output	Data type	Format
	Incident data	String	<i>[E.G.: "Id: 2, type: theft, location: 5th street #10, date_time: 2025-03-02_14:30:00, status: pending"]</i>

Identifier and name	RF9–/Writing JSON files (Passenger registration)
Summary	

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	The system must allow users to register as passengers by providing information such as: ID, name, assigned route and contact information. The system verifies each piece of information to see if it is correct or not.		
Tickets	Entry name	Data type	Condition valid values
	Passenger ID	String	<i>[The id must be unique, it must contain a string of numbers]</i>
	Name of passenger	String	<i>[Must contain only letters]</i>
	Assigned route	String	<i>[The user must enter one of the routes already registered in the system]</i>
	Passenger contact	String	<i>[User must enter their phone number (10 digits)]</i>
Result or Postcondition	The system must store the user in a database		
Departures	Name of output	Data type	Format
	Confirmation message	String	<i>"User registration was successful"</i>
	Error message	String	<i>If the data does not match, or if not all the data was entered, then the system displays "the registration was incorrect, please try again."</i>

Identifier and name	RF10--/Reading JSON files (Passenger registration)
Summary	The system must allow users to register as passengers by providing information such as: ID, name, assigned route and contact information.

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	The system verifies each piece of information to see if it is correct or not.		
Tickets	Entry name	Data type	Condition valid values
Result or Postcondition	The system shall process and display the list of registered passengers.		
Departures	Name of output	Data type	Format
	Passenger list	List	<pre>[{ "ID": "P1", "Name": "Juan Pérez", "Route": "R1", "Contact": "3214567890" }]</pre>

Identifier and name	RF11 – Writing JSON files (Driver Registration)		
Summary	The system must allow users to register as drivers, providing information such as: driver ID, name, license plate of the assigned vehicle and status, whether available or en route.		
Tickets	Entry name	Data type	Condition valid values
	Driver ID	String	<i>[The id must be unique, it must contain a string of numbers]</i>
	Driver Name	String	<i>[Must contain only letters]</i>

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	Assigned Vehicle Plate	String	<i>[must be unique in the system, must be a set of 3 letters and therefore 3 numbers]</i>
	State	String	<i>[path must be available]</i>
Result or Postcondition	The system must store the driver in the database		
Departures	Name of output	Data type	Format
	Confirmation message	String	<i>"The driver's registration was successful"</i>
	Error message	String	<i>If the data does not match, or if not all the data was entered, then the system displays "the registration was incorrect, please try again."</i>

Identifier and name	RF12 – Reading JSON files (Driver Registration)		
Summary	The system must allow reading JSON files with information on registered drivers.		
Tickets	Entry name	Data type	Condition valid values
Result or Postcondition	Data is loaded into memory and stored in a linked list.		
Departures	Name of output	Data type	Format
	List of drivers	List	<i>"["</i>

			<pre>{ "ID": "C1", "Name": "Pedro Gómez", "Plate": "ABC123", "Status": "Available" }]</pre>
--	--	--	---

Identifier and name	RF13 – Search for drivers by name		
Summary	The system must allow the search of drivers registered in the database already stored by the system based on their name. For this to happen, the system must first verify that the list of drivers is not empty and perform an efficient search in the linked list.		
Tickets	Entry name	Data type	Condition valid values
	Driver Name	String	<i>[Must contain only letters]</i>
Result or Postcondition	The system should display the complete information of the driver searched for		
Departures	Name of output	Data type	Format
	Driver Found	String	<i>"ID: 34242, Name: Carlos, Vehicle: efg 123, Status: Route available"</i>
	Error message	String	<i>"error, no driver was found with the name entered."</i>

Identifier and name	RF14 – Report generation
---------------------	--------------------------

Summary	The system must generate and show the user all reports on routes ordered from shortest to longest time or distance, incidents ordered from most recent to oldest, and finally the results of specific searches performed by the user.		
Tickets	Entry name	Data type	Condition valid values
Result or Postcondition	The system will generate the reports automatically taking all the previous information		
Departures	Name of output	Data type	Format
	Report of ordered routes	String	<i>"The information of all routes is displayed, ordered either by time or distance: ID: 1, Distance: 3km, estimated travel time: 45min, start: A, end: B"</i>
	Ordered incident reporting	String	<i>"Information about all incidents is displayed by date/time: ID: 1, type: theft, location: Carrera 25 #45 Alcázares neighborhood, date/time: YYYY-MM-DD _ HH:MM:SS, status: in process"</i>
	Specific search results	String	<i>The results of specific searches made by the user will be displayed.</i>

Identifier and name	RF15 – Real-time monitoring of emergency vehicles and security patrols		
Summary	The system must monitor the location of emergency vehicles and security patrols in real time.		
Tickets	Entry name	Data type	Condition valid values
	Vehicle plate	String	<i>[must be alphanumeric in three letters and three numbers format]</i>
Result or Postcondition	The system must process and save the vehicle's location and status in a database.		
Departures	Name of output	Data type	Format
	Vehicle status	String	<code>{"vehicles": [{"Plate": "ABC123", "Location": "(3,451, -76,532)", "Status": "In emergency"}], {"Plate": "XYZ789", "Location": "(3,457, -76,540)", "Status": "Available"}}</code>

Identifier and name	RF16 – Real-time tracking of emergency vehicles and security patrols
---------------------	--

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

Summary	The system must continuously track the location of emergency vehicles and security patrols, storing their position history and issuing alerts when there are significant changes in their location.		
Tickets	Entry name	Data type	Condition valid values
	Vehicle plate	String	<i>[must be alphanumeric in three letters and three numbers format]</i>
Result or Postcondition	The system stores a history of each vehicle's locations and generates notifications if there are relevant movements.		
Departures	Name of output	Data type	Format
	Vehicle status	String	<pre>{ "vehicle": { "Plate": "ABC123", "Current_Location": "(3,451, -76,532)", "Status": "On Route", "History": [{ "Location": "(3,450, -76,530)", "Time": "10:00:00" }, { "Location": "(3,451, -76,531)", "Time": "10:05:00" }], "Notification": "The vehicle has moved 2 km to the north." } }</pre>

Identifier and name	RF17 – Real-time information on traffic and transport		
Summary	The system should provide real-time information on traffic conditions and the availability of public transport.		
Tickets	Entry name	Data type	Condition valid values

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	Track ID	String	<i>[unique, alphanumeric string]</i>
	Traffic level	String	<i>["fluid", "moderate", "congested"]</i>
	State of transport	String	<i>["Available", "Delayed", "Out of Service"]</i>
	Location	String	<i>[The user must enter the address or coordinates of where the incident occurs. (Ex.: Carrera 45 #2, Alcázares neighborhood)]</i>
Result or Postcondition	The system processes and stores information in a database updated in real time.		
Departures	Name of output	Data type	Format
	Traffic status	String	<i>"traffic": { "ID_via": "V123", "Location": "5th Street", "Status": "Congested"}</i>
	State of transport	String	<i>"transport": { "Line": "Bus 45", "Status": "Delayed"}</i>

Identifier and name	RF18 – Providing real-time information on traffic status and public transport availability.		
Summary	The system should provide real-time information on traffic conditions and the availability of public transport.		
Tickets	Entry name	Data type	Condition valid values

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	Track ID	String	<i>[unique, alphanumeric string]</i>
	Traffic level	String	<i>["fluid", "moderate", "congested"]</i>
	State of transport	String	<i>["Available", "Delayed", "Out of Service"]</i>
	Location	String	<i>[The user must enter the address or coordinates of where the incident occurs. (Ex.: Carrera 45 #2, Alcázares neighborhood)]</i>
Result or Postcondition	The system processes and stores information in a database updated in real time.		
Departures	Name of output	Data type	Format
	Traffic status	String	<i>"traffic": { "ID_via": "V123", "Location": "5th Street", "Status": "Congested"}</i>
	State of transport	String	<i>"transport": { "Line": "Bus 45", "Status": "Delayed"}</i>

Identifier and name	RF19 – Statistics generation for decision-making by the authorities.
Summary	The system must generate reports and statistical analysis based on real-time traffic, incidents and public transport, providing relevant information for decision-making by authorities.

Tickets	Entry name	Data type	Condition valid values
Result or Postcondition	The system will take into account the data from the previous requirements to generate a report and detailed statistics to extract recommendations that help optimize mobility management and security.		
Departures	Name of output	Data type	Format
	Report and statistics	String	<pre> { "report": { "Type": "Security", "Period": "Last month", "Region": "Center", "Summary": "12% increase in nighttime road incidents.", "Recommendations": "Increase patrols in critical areas from 8:00 p.m. to 5:00 a.m." } } </pre>

Identifier and name	RF20 – Notification to citizens about incidents, alternative routes and safety recommendations		
Summary	The system must send real-time notifications to citizens regarding incidents, alternate routes, changes in routes, recommendations to improve mobility and safety.		
Tickets	Entry name	Data type	Condition valid values
Result or Postcondition			

Icesi University
Department of Computing and Intelligent Systems
Problem identification and requirements analysis
Algorithms and Programming II

	The system will generate alerts to citizens that allow them to make informed decisions about their safety and mobility.		
Departures	Name of output	Data type	Format
	Incident notification	String	<i>{ "Type": "Incident", "Message": "Accident on Calle 5, avoid the area." }</i>
	Suggested alternative route	String	<i>{ "Type": "Alternate Route", "Message": "Use 7th Street instead of 5th Street." }</i>
	Warning about your safety	String	<i>{ "Type": "Security", "Message": "Increase in robberies in the area after 9 pm" }</i>