Software Requirements Specification

for

Information system for the management of the police inspectorate

Version 1.7 approved

Prepared by Barbu Florin Alexandru
Chiricioiu Roxana Elena
Cîrstea Adriana Mihaela

Faculty of Automation, Computers, and Electronics

March 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Cîrstea Adriana Mihaela	01.03.2025	Update Overall Description section	1.1
Chiricioiu Roxana Elena	03.03.2025	Update Introduction section	1.2
Barbu Florin Alexandru	04.03.2025	Update External Interface Requirements	1.3
Cîrstea Adriana Mihaela	07.03.2025	Adding functional and non-functional requirements	1.4
Barbu Florin Alexandru	07.03.2025	Adding functional and non-functional requirements	1.5
Chiricioiu Roxana Elena	07.03.2025	Adding functional and non-functional requirements	1.6
Barbu Florin Alexandru	08.03.2025	Update System Features section	1.7

1. Introduction

1.1 Purpose

The application is a client-server computer system that manages all activities of a police inspectorate. It allows for file management, monitoring of human and material resources, internal communication, data analysis, and incident reporting.

1.2 Document Conventions

Abbreviations

RBAC	Role-Based Access Control
PI	Police Inspectorate
PDF	Portable Document Format
OTP	One-Time Password
NARA	National Archives and Records Administration
ISO	International Organization for Standardization
GDPR	General Data Protection Regulation
DOCX	Document File Format
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
API	Application Programming Interface

1.3 Intended Audience and Reading Suggestions

- **Developers:** Need technical details about the system architecture, software functionalities, and integration with other systems used in the inspectorate.
- **Project Managers:** Will focus on the general system requirements, implementation objectives, and delivery planning.
- **Operational Staff:** Will refer to the document to understand how to use the software in their daily activities, including data entry and retrieval.
- **Testers:** Will focus on the sections related to validation and verification of software functionalities, identifying potential issues, and reporting bugs.

1.4 Product Scope

The application must facilitate efficient case and police activity management by offering features such as case allocation, status updates, search, and secure archiving. It should also allow the management of suspects, and incidents, provide a system for quick notifications and alerts, and facilitate internal communication among officers. Additionally, the application will include features for police station map visualization, and control access to sensitive information based on user roles (administrator, inspector, officer).

1.5 References

[1] Google Java Style Guide. https://google.github.io/styleguide/javaguide.html
[2] Leaflet - a JavaScript library for interactive maps. https://leafletjs.com/
[3] European Union, "General Data Protection Regulation (GDPR)". https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679, 2016.
[4] Jenifer Tidwell, Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly Media, 2010.
[5] J.F. Kurose and K.W. Ross, *Computer Networking: A Top-Down Approach*, Pearson, 2017.
[6] C. Richardson, *Microservices Patterns*, Manning Publications, 2018.
[7] IEEE. (n.d.). *IEEE Software Engineering Standards*. https://standards.ieee.org/standard/index.html, n.d.
[8] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, v. 24, n. 11, 1981, pp. 770–772.
[9] J. Daemen and V. Rijmen, "The Design of Rijndael: AES - The Advanced Encryption Standard," Springer, 2002.
[10] E. Vasilescu, "Encryption Techniques for Data Protection," *Romanian Academy Publishing House*, 2020.
[11] International Organization for Standardization (ISO), *ISO/IEC 27001: Information Security Management*, 2013.
[12] National Archives and Records Administration (NARA), *Electronic Records Management in Law Enforcement*, 2017.

2. Overall Description

2.1 Product Perspective

The application must facilitate efficient case and police activity management by offering features such as case allocation, status updates, search, and secure archiving. It should also allow the management of suspects, vehicles, and incidents, provide a system for quick notifications and alerts, and facilitate internal communication among officers. Additionally, the application will include features for patrol planning, report generation, police station map visualization, and control access to sensitive information based on user roles (administrator, inspector, officer).

2.2 Product Functions

- **Case Allocation:** Assigning a case to an officer or a team of police officers.
- **Case Status Update:** Updating the status of a case (open, in progress, resolved, archived).
- **Case Search:** Quickly searching cases based on various criteria (case number, suspect name, location, etc.).
- **Case Storage:** Securely archiving and storing electronic files with the ability to quickly access them when needed.
- **Police Station Location:** Displaying all police stations on a map.
- **Adding and Updating Stations:** Ability to add new police stations or update information about existing ones.
- **Assigning Agents to Stations:** Assigning agents to police stations and defining hierarchical relationships between them.
- **Availability View:** Displaying the availability of officers and teams.
- **Report Generation:** Producing periodic reports on police activities (number of cases opened/closed, reported incidents, etc.).
- **Inventory Management:** This helps to store and keep track of the equipment assigned to agents.
- **Incident Management:** Recording and managing information about reported incidents.
- **Facilitating Internal Communication:** An internal messaging system for communication between officers and teams.
- **Sending Quick Alerts:** Sending rapid alerts to all officers in case of emergency.
- **Creating and Managing Team Discussions:** Creating discussion channels for specific teams.
- **User Authentication:** Secure authentication system for all users.
- **Access Control:** Controlling access to sensitive information based on the user's role (officer, inspector, administrator).
- **Citizen help:** Citizens can help by sending a form about a possible crime and get notified about their sensitization.

2.3 User Classes and Characteristics

- **System Administrator:** manage users, configure permissions, monitor activity, update the system;
- **Head of the inspectorate:** manage files, analyze reports, send alerts;
- **Police Officer/Agent:** records and updates cases and incidents, communicates with colleagues through internal messaging, receives and responds to quick alerts, searches information in databases (suspects), receives patrol assignments, uses the map to view important locations;

2.4 Operating Environment

- **Operating System (OS):** Windows: Windows 10 or newer (version 21H1 or later), Linux: Ubuntu 20.04 LTS or other modern distributions with kernel 5.4+.
- **Web Browsers:** Google Chrome: Version 100 or later, Microsoft Edge: Version 95 or later.
- **Internet Connection:** Stable connection of at least 5 Mbps for real-time synchronization and updates.

2.5 Design and Implementation Constraints

- PostgreSQL (version 17)
- IntelliJ IDEA Ultimate Edition 2024.3.1.1
- GitHub
- Visual Studio Code
- JUnit

2.6 User Documentation

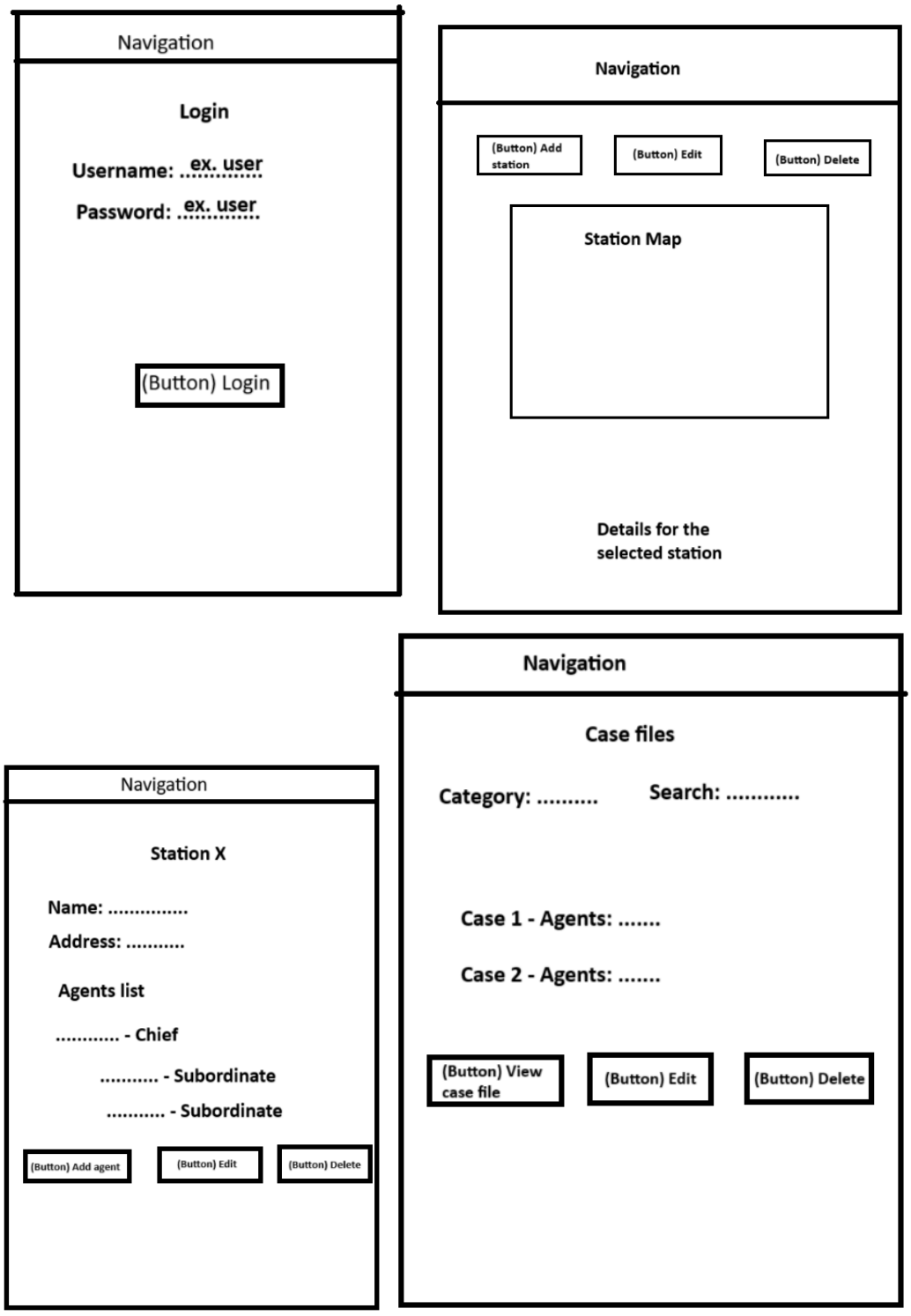
The user documentation will include a short and concise document that provides basic instructions for starting to use the application (e.g., how to log in, how to navigate menus), designed to guide users in exploring and efficiently using the application. It also offers detailed instructions on: navigating the application's interface, performing specific operations such as data entry.

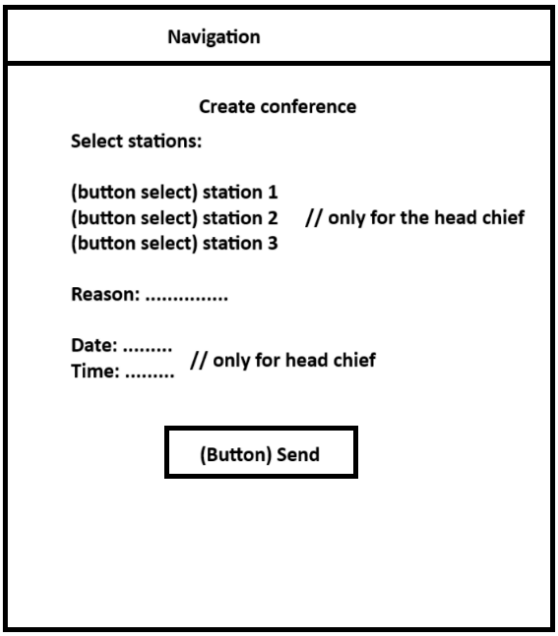
2.7 Assumptions and Dependencies

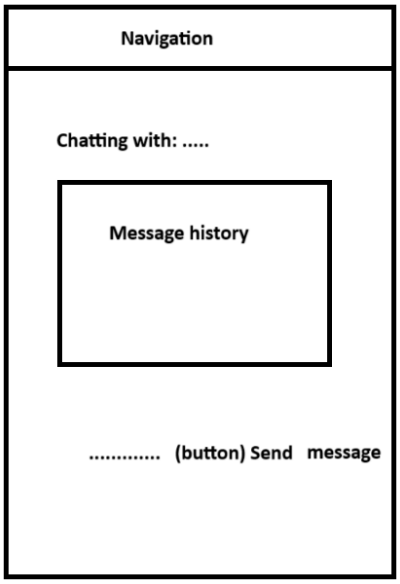
Not applicable!

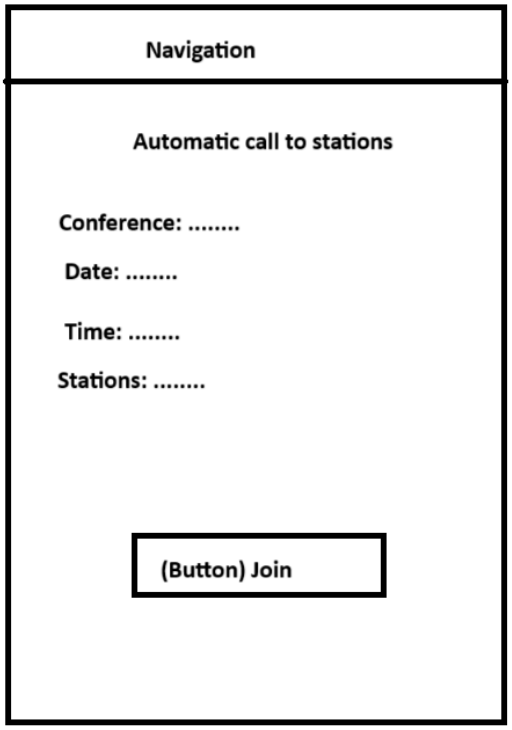
3. External Interface Requirements

3.1 User Interfaces









3.2 Hardware Interfaces

Not applicable!

3.3 Software Interfaces

Agent:

- public addAgent();
- public editAgent(agentId);
- public deleteAgent(agentId);
- public getAgent(agentId);
- public getAgents();

CaseFile:

- public addCaseFile();
- public editCaseFile(caseFileId);
- public deleteCaseFile(caseFileId);
- public getCaseFile(caseFileId);
- public getCaseFiles();

Station:

- public addStation();
- public editStation(stationId);
- public deleteStation(stationId);
- public getStation(stationId);
- public getStations();

3.4 Communications Interfaces

Not applicable!

4. System Features

4.1 Police Station Management

4.1.1 Description and Priority

- **Description**: This allows the creation, modification, and visualization of police stations, including their locations, representations, and connections between stations. It also supports the management of agents within each station, including their hierarchy and relocation.

- **Priority**: High

4.1.2 Stimulus/Response Sequences

- **Input**: Admin adds a new police station with a specific geographical location.
  Output: System displays the station on a map with its graphical representation.

- **Input**: Admin connects stations with lines annotated with distances.
  Output: System saves the connections and displays them on the map.
- **Input**: Head of inspectorate adds or updates agent information within a station.
  Output: System saves the updated agent information and displays it in the station's graphical representation.

4.1.3 Functional Requirements

REQ-1: Police stations will have predefined geographical locations, with the ability to add new stations and save their locations.
REQ-2: Each station will have a graphical representation, with connections between stations displayed as annotated lines indicating distances.
REQ-3: The graphical representation of a police station must allow the addition and visualization of agent information, including hierarchical relationships.
REQ-4: The geographical distribution of stations and the allocation of agents must be saved in a file in a predefined format.
REQ-5: For storing information and managing, a database server must be configured.
REQ-6: The system must allow the addition of new agents and the removal of those who are no longer part of the station.
REQ-7: The system must allow updating information about agents (professional training, moral conduct, resolved cases).
REQ-8: The system must allow the tracking of agent availability (e.g., on duty, off duty, on leave).

4.2 Case File Management

4.2.1 Description and Priority

- **Description**: This enables the storage, retrieval, and management of files related to criminal activities. Files can be categorized, searched, and modified, with a record of the last agent who made changes.

- **Priority**: High

4.2.2 Stimulus/Response Sequences

- **Input**: Head of inspectorate creates a new file and assigns it to a category of criminal activity.
  Output: System saves the file and assigns it a unique case ID.
- **Input**: User searches for files by name, agent names, or category.
  Output: System displays the file with the "fingerprint" of the last modifying agent.
- **Input**: User updates file information.
  Output: System records the modifying agent and updates the file.

- **Input**: User requests to view the status of a case file.
  Output: System displays the current status (e.g., Open, In Progress, Closed, Archived).

4.2.3 Functional Requirements

REQ-9: Files must be stored by categories of criminal activity.
REQ-10: Retrieving files must be possible by the file name, the names of the agents working in the investigation team, or by category.
REQ-11: Each file must store the "fingerprint" of the last agent who modified it.
REQ-12: Each case file must allow the attachment of documents, images, and videos as evidence.
REQ-13: Case files should have an assigned status (Open, In Progress, Closed, Archived).
REQ-14: The system must support version control for case files, allowing users to view previous versions of a file.
REQ-15: The system must allow the addition of notes and comments to case files by authorized agents.
REQ-16: The system must allow the assignment of case files to specific agents or teams.

4.3 Communication and Collaboration

4.3.1 Description and Priority

- **Description**: This facilitates communication between users (agents, head of inspectorate) through a chat system, conference planning, and real-time notifications.

- **Priority**: High

4.3.2 Stimulus/Response Sequences

- **Input**: User initiates a chat session with another user or group.
  Output: System establishes the chat session and displays the chat interface.
- **Input**: User schedules a conference with a specified reason and personal code.
  Output: System sends real-time notifications for conference requests.
- **Input**: User requests to view scheduled conferences.
  Output: System displays the list of scheduled conferences.
- **Input**: System automatically calls police stations at scheduled times.
  Output: System logs the call and notifies participants.

4.3.3 Functional Requirements

REQ-17: A predefined call protocol will be established for each week, with the ability to retain it for subsequent weeks.
REQ-18: The system must allow the implementation of a communication program

(chat) with the possibility that at any given time two users (the head of the inspectorate and an agent from a station or two agents from different stations) or more users (the head of the inspectorate and agents from multiple stations) can be logged in.

REQ-19: The system must allow the planning of conferences, the launching of conference requests by the head of the inspectorate or station agents, with the specification of the reason and the personal code.

REQ-20: The system must perform automatic calling of police stations at the scheduled date and time, using specific codes.

REQ-21: The system must allow the scheduling of recurring conferences.

REQ-22: The system must support file sharing within the chat system (e.g. PDF, DOCX).

REQ-23: The system must provide a notification system for urgent messages or updates.

REQ-24: Both clients and servers must have a simple and intuitive graphical interface.

4.4 Crime Analytics

4.4.1 Description and Priority

- **Description**: This provides tools for analyzing crime data, generating reports, and identifying trends or hotspots.

- **Priority**: Low

4.4.2 Stimulus/Response Sequences

- **Input**: User requests to view crime statistics.
  Output: System displays a dashboard with crime statistics (e.g., crime rate, types of crimes, solved cases).
- **Input**: User requests predictive analytics for potential crime hotspots.
  Output: System displays predicted hotspots on a map.
- **Input**: User creates a custom report by selecting filters (e.g., crime type, location, time range).
  Output: System generates and displays the custom report.

4.4.3 Functional Requirements

REQ-25: The system must provide a dashboard to visualize crime statistics (e.g., crime rate, types of crimes, solved cases).

REQ-26: The system must provide predictive analytics to identify potential crime hotspots.

REQ-27: The system must allow the creation of custom reports based on user-defined filters.

REQ-28: The system must generate automated reports on crime trends and patterns.

4.5 Resource Management

4.5.1 Description and Priority

- **Description**: This allows the management of police resources (e.g., vehicles, weapons, equipment) and their allocation to stations or agents.

- **Priority**: Medium

4.5.2 Stimulus/Response Sequences

- **Input**: Head of inspectorate adds a new resource (e.g., vehicle, weapon) to the system.
  Output: System saves the resource and displays it in the resource management interface.
- **Input**: Head of inspectorate assigns a resource to a station or agent.
  Output: System updates the allocation and displays the updated resource list.
- **Input**: Head of inspectorate requests to view the maintenance schedule for a resource.
  Output: System displays the maintenance schedule.

4.5.3 Functional Requirements

REQ-29: The system must allow the management of police resources (e.g., vehicles, weapons, equipment) and their allocation to stations or agents.
REQ-30: The system must allow the addition of new resources and their categorization (e.g., patrol cars, motorcycles, drones).
REQ-31: The system must track the maintenance schedule for police vehicles and equipment.

4.6 Public Interaction and Reporting

4.6.1 Description and Priority

- **Description**: This provides a public interface for citizens to report crimes, track cases, and interact with law enforcement.

- **Priority**: Medium

4.6.2 Stimulus/Response Sequences

- **Input**: Citizen reports a crime through the public interface.
  Output: System generates a unique case ID and sends an acknowledgment to the citizen.
- **Input**: Citizen uploads evidence (e.g., photos, videos) when reporting a crime.
  Output: System saves the evidence and links it to the case file.
- **Input**: Citizen requests to track the status of a reported case.
  Output: System displays the current status of the case.

4.6.3 Functional Requirements

REQ-32: The system must provide a public interface for citizens to report crimes or suspicious activities.
REQ-33: The system must allow citizens to upload evidence (e.g., photos, videos, documents) when reporting a crime.
REQ-34: The system must allow citizens to track the status of their reported cases.
REQ-35: The system must send an acknowledgment (e.g., email, SMS) to the citizen after a crime report is submitted.

4.7 User Authentication

4.7.1 Description and Priority

- **Description**: This ensures secure access to the system by authenticating users through usernames/email and passwords, with additional security measures like encryption and role-based access control.

- **Priority**: High

4.7.2 Stimulus/Response Sequences

- **Input**: User enters their username and password to log in.
  Output: System authenticates the user and grants access based on their role.
- **Input**: User requests to reset their password.
  Output: System sends a password reset link or OTP to the user's registered email or phone number.
- **Input**: Administrator modifies user roles or permissions.
  Output: System updates the user's access rights.

4.7.3 Functional Requirements

REQ-36: The system must allow registered users to log in using their email/username and password.

REQ-37: The system must display an error message for invalid login credentials without revealing specific details (e.g., "Invalid username or password").
REQ-38: The system must allow users to reset their password via a "Forgot Password" feature.
REQ-39: The system must encrypt all passwords and sensitive user data stored in the database.
REQ-40: The system must restrict access to specific features based on the user's role (e.g., only administrators can add new police stations).
REQ-41: Passwords must be at least 8 characters long and contain at least one digit and one special character.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

REQ-42: The computing system characteristics for installing the software system are: a 1 GHz or faster processor; at least 128 MB of memory; a minimum of 10 MB of hard disk space.
REQ-43: The system should provide timely responses to user requests, ensuring that the majority of operations are handled swiftly and efficiently.
REQ-44: The system must load all static assets (e.g., images, CSS, JavaScript) quickly.

5.2 Safety Requirements

Not applicable!

5.3 Security Requirements

REQ-45: All data transmitted between stations and the inspectorate must be encrypted.
REQ-46: The system must generate activity logs for all critical actions (e.g., case modifications, access to sensitive data).
REQ-47: The system must implement role-based access control (RBAC) to restrict access to sensitive features and data.

5.4 Software Quality Attributes

- Block comments are indented at the same level as the surrounding code. They may be in /* ... */ style or // ... style. For multi-line /* ... */ comments, subsequent lines must start with * aligned with the * on the previous line.
- There shall be no unreachable code.
- The private modifier is used for methods and variables that need to be accessible only within the same class.

- The public modifier is avoided when access outside the class or package is not necessary.
- Each top-level class resides in a source file of its own.
- Braces are used with if, else, for, do and while statements, even when the body is empty or contains only a single statement.
- Method names are written in lowerCamelCase.
- Method names are typically verbs or verb phrases. For example, sendMessage or stop
- Class names are written in UpperCamelCase.
- Class names are typically nouns or noun phrases. For example, Character or ImmutableList
- Parameter names are written in lowerCamelCase.
- The source file name consists of the case-sensitive name of the top-level class it contains (of which there is exactly one), plus the .java extension.

5.5 Business Rules

REQ-48: The system must comply with GDPR regulations regarding the protection of personal data.
REQ-49: Personal data processed by the system must be used only for law enforcement purposes (e.g., criminal investigations, public safety).
REQ-50: Access to personal data must be restricted to authorized personnel only (e.g., agents, administrators).
REQ-51: Users should have the minimum level of access required to perform their duties. For example, a patrol officer should not have access to administrative functions.

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List