

Software Requirements Specification

Antivirus Software System

Saurabh Ashok Sawant

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Definitions, acronyms, and abbreviations.....	1
1.4 References	2
1.5 Overview	2
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product functions	3
2.3 User Characteristics	4
2.4 Constraints	4
2.5 Assumptions and dependencies	5
2.6 Apportioning of requirements.....	5
3. Specific Requirements	6
3.1 External interface Requirements.....	6
3.1.1 User interfaces.....	6
3.1.2 Hardware interfaces	7
3.1.3 Software interfaces.....	7
3.1.4 Communications interfaces.....	7
3.2 Functional requirements.....	8
3.2.1 User Class 1- The User	8
3.2.2 User Class 2- System Administrator.....	11
3.2.3 User Class 3- Developer/Security Analysts	11
3.3 Performance requirements	12
3.4 Design constraints	14
3.5 Software system attributes	15

1.Introduction

Antivirus software plays a critical role in safeguarding computer systems against malicious attacks and unauthorized access. This document presents a detailed specification for developing such software.

1.1 Purpose

The purpose of this document is to define the software requirements for a robust and efficient antivirus application. This software is designed to protect computer systems from a wide range of malicious threats including viruses, worms, ransomware, spyware, trojans, and other forms of malware. It outlines the functional and non-functional requirements necessary to ensure that the antivirus software effectively detects, prevents, and removes malware, while minimizing system performance impact.

1.2 Scope

This antivirus software will provide real-time protection, periodic system scans, malware detection, quarantine, and removal. It will support Windows, macOS, and Linux systems. The software will include automatic updates to ensure protection against the latest threats and will operate with minimal system resource consumption. Users will have access to a user-friendly interface for configuring scan schedules and managing threat reports. The antivirus will also include web protection features to block access to malicious websites. Additionally, it will generate security reports and logs to help users monitor the system's health and threat history.

It will support both manual and automated scanning modes, giving users flexibility in how they manage system security. The software will also be capable of integrating with email clients to scan attachments for potential threats.

1.3 Definitions, acronyms, and abbreviations

Table 1 - Definitions

Term	Definition
User	Someone who installs and uses the antivirus software on their device.
Admin/Administrator	A user with elevated privileges who manages security settings and configurations.
Malware	Malicious software such as viruses, worms, trojans, ransomware, or spyware.
Real-Time Protection	A feature that continuously monitors the system to detect and block threats instantly.
Quarantine	A secure location where detected malware is isolated to prevent further harm.
Scan	The process of analyzing files and programs to detect any malicious content.

Threat Database	A regularly updated collection of known malware signatures used for detection.
GUI	Graphical User Interface- the visual interface users interact with
OS	Operating System- the platform on which the antivirus software runs (e.g., Windows, macOS, Linux).
VPN	Virtual Private Network- a service that provides secure and private internet access.
Firewall	A security system that monitors and controls incoming and outgoing network traffic.
Heuristics	A detection method that identifies malware based on behavior patterns rather than known signatures
Update	The process of refreshing the virus definitions and software components to ensure up-to-date protection.
Cloud Scanning	A feature that uses cloud-based analysis to detect and block newly emerging threats

1.4 References

- [1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
- [2] Stallings, W. “Computer Security: Principles and Practice”, Pearson Education, 3rd Edition, 2015.
- [3] Bishop, M. “Introduction to Computer Security”, Addison-Wesley, 2004.
- [4] Symantec Corporation, “Best Practices for Antivirus Protection”, Technical White Paper, 2018.
- [5] National Institute of Standards and Technology (NIST), “Guide to Malware Incident Prevention and Handling”, Special Publication 800-83 Revision 1, 2013.

1.5 Overview

The rest of this document includes three chapters and appendices. Chapter 2 gives an overview of the antivirus software, its functions, user roles, and system constraints. Chapter 3 details the specific requirements, including interfaces and system behavior. Chapter 4 focuses on requirement prioritization and the reasons for chosen methods. The appendices include prioritization results and the release plan. This document is structured to guide developers, testers, and stakeholders throughout the software lifecycle. It ensures all technical and non-technical users understand the system’s goals and functionality.

Each section builds upon the previous to maintain logical flow and traceability. The SRS aims to serve as a single source of truth for all project-related requirements.

2. Overall description

This section provides an overview of the antivirus software system. It explains the system in its context, including how it interacts with operating systems, external update servers, and user environments. The basic functionality such as real-time protection, system scanning, and malware removal is introduced here.

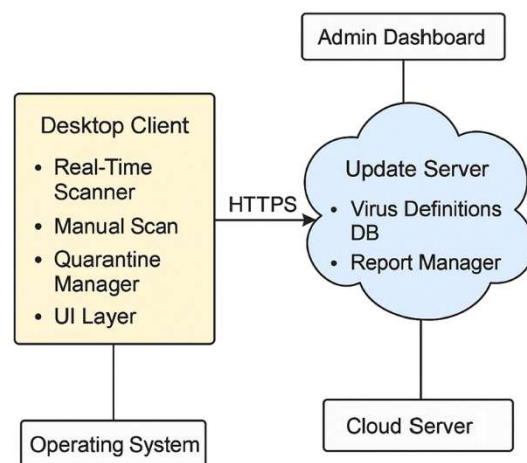
It also outlines the different types of stakeholders general users, IT administrators, and system auditors and describes the key features available to each. Finally, the section presents system constraints, assumptions, and any dependencies required for proper functioning.

2.1 Product perspective

The antivirus system will consist of two main components: a desktop client application and a cloud-based update and reporting server. The desktop client will provide real-time protection, system scanning, threat detection, and quarantine capabilities, while the cloud server will manage virus definition updates, log synchronization, and remote policy management (for enterprise use).

The client application will interact with the operating system to monitor files, processes, and network activity. It will use a local engine for scanning and will periodically connect to the cloud server to fetch the latest malware signatures and upload scan logs. The client operates continuously in the background and presents alerts through a user-friendly interface.

Since the software heavily depends on updated virus definitions and reporting, a central server is used. This server handles requests from multiple clients for updates and collects data for analytics and administrative review. All communication between the client and server is secured via HTTPS to prevent tampering and ensure integrity.



The antivirus application has lightweight system requirements to avoid impacting user performance. It is optimized to use no more than 100 MB of RAM in idle mode and under 500 MB of disk space for logs, quarantine, and definition storage. The system is designed to run on Windows, macOS, and Linux platforms with minimal configuration.

2.2 Product functions

The antivirus software will allow users to perform real-time and manual scans on their devices to detect and remove potential threats. Users will also be able to manage quarantined files and interact with the system via a user-friendly interface.

The desktop client will continuously monitor the system for suspicious activity and automatically respond to threats as they are detected. It will also allow users to initiate manual scans of specific files or the entire system.

Through the UI, users can access scan reports, update virus definitions, and configure scan settings. All updates and definition files will be retrieved securely via HTTPS from the cloud-based update server.

Administrators will have access to a web-based dashboard to view system reports, manage threat definitions, and monitor activity across connected devices.

2.3 User characteristics

There are three primary types of users who interact with the antivirus system: end users, system administrators, and developers or security analysts. Each user type has distinct roles and requirements within the system.

- 1) End Users interact with the desktop client. They rely on the antivirus software for protecting their devices against malware and other threats. Their primary interactions include:
 - Running real-time and manual scans
 - Viewing and restoring quarantined items
 - Updating virus definitions
 - Viewing scan reports
 - Configuring scan schedules and basic settingsThese users generally have limited technical knowledge, so the interface must be intuitive and easy to use.
- 2) System Administrators access the web-based Admin Dashboard. They are responsible for overseeing antivirus deployments across multiple devices in an organization. Their responsibilities include:
 - Viewing and managing security reports
 - Monitoring scan logs across clients
 - Configuring centralized policies (e.g., automatic scan schedules)
 - Managing virus definition rollouts
 - Responding to detected threats across the network
- 3) Security Analysts / Developers may use both the Admin Dashboard and back-end tools. They require deeper access for:
 - Investigating reported threats
 - Managing and updating the virus definition database
 - Debugging client-side issues
 - Analyzing false positives and updating detection logicThese users are technically skilled and require more detailed logs and data access.

2.4 Constraints

The antivirus system is subject to several constraints that may impact its performance and compatibility. Firstly, the desktop client is dependent on the underlying operating system, which means that certain scanning operations and access to system files may vary based on the OS version and its built-in security policies. Additionally, the application requires a stable Internet connection to fetch the latest virus definitions, submit scan reports, and communicate with the update server. Without Internet access, the antivirus may operate with outdated definitions, reducing its ability to detect new threats. Performance constraints on client devices, such as limited CPU, RAM, or storage, can also affect the smooth functioning of real-time scanning, especially during system startup or high resource usage. Another important constraint is the dependency on the update server, which if unavailable or overloaded, may delay updates or report synchronization. Moreover, both the desktop client and the admin dashboard share access to a centralized cloud database. Heavy traffic or simultaneous requests can lead to increased

latency or slower response times. Lastly, the system must adhere to data privacy and security regulations, such as GDPR, which may restrict data collection and retention policies, further influencing system behavior and design.

2.5 Assumptions and dependencies

It is assumed that the antivirus software will be installed on systems that meet the minimum hardware and operating system requirements. If the system lacks sufficient resources such as processing power, memory, or disk space the application may not function as expected, particularly during real-time scanning or full-system scans.

Another key assumption is that users have a stable Internet connection, which is necessary for virus definition updates, server communication, and log synchronization. Without Internet access, the software's ability to detect newly emerging threats will be limited.

The antivirus system also assumes that the operating system provides consistent and secure access to system-level events, files, and processes. If the OS restricts these permissions or changes its access policies in future updates, the software may require modification to maintain compatibility.

Additionally, the system depends on a reliable cloud-based update server for pushing virus definitions and receiving reports. Any downtime or delay in server responsiveness may impact software performance and user protection. It is also assumed that third-party security features (e.g., firewalls or other antivirus tools) will not interfere with the antivirus software's operation.

2.6 Apportioning of requirements

If the development schedule is delayed, certain non-critical features may be postponed and included in a later version of the antivirus software. These deferred requirements will be prioritized for the third release, as outlined in Appendix IV. Core functionalities such as real-time scanning, manual scans, virus definition updates, and quarantine management will be delivered in the initial release. Advanced features like centralized remote management, detailed threat analytics, and customizable scan profiles may be rescheduled depending on resource availability and project timelines.

3. Specific requirements

This section outlines all functional and non-functional requirements of the antivirus software. It provides detailed descriptions of the system's behavior, features, and performance expectations.

3.1 External interface Requirements

This section describes the system's inputs and outputs, along with its hardware, software, and communication interfaces. It also outlines the basic structure of the user interface and how users will interact with the system.

3.1.1 User interfaces

When a user opens the antivirus software for the first time, they are presented with a Login Screen, as shown in Figure 1. This screen prompts the user to enter their username and password, with an optional "Remember me" checkbox for convenience. New users are expected to register before logging in, while returning users can proceed directly to access the application's features.

Once logged in, the user is directed to the Manual Scan interface (see Figure 2), where they can choose between a Quick Scan, Full Scan, or Custom Scan to suit their needs. A prominent Start Scan button initiates the chosen scan type. Additionally, a language selection option is available, allowing users to customize the interface language according to their preference.

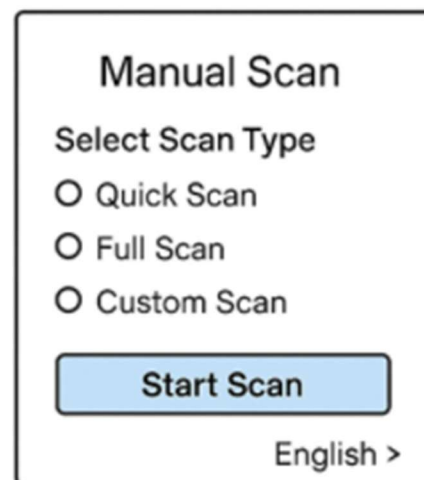
If threats are detected during a scan, they are moved to the Quarantine section, displayed in Figure 3. This screen lists the infected files along with the date they were detected. Users can then choose to Restore or Delete the quarantined files, depending on whether they believe the threat was a false positive or a real infection.

Administrators and advanced users can access the Web Dashboard, as illustrated in Figure 4. This dashboard provides a centralized view of the antivirus activity, including the total number of threats blocked and a button to View Reports. These reports help track system security over time and support better decision-making regarding system protection and maintenance.



The diagram shows a rectangular box titled "Login Screen". Inside the box, there are two input fields: "User Name" and "Password". Below these fields is a checkbox labeled "Remember me". At the bottom of the box is a blue button labeled "Log In".

Figure 1 Login



The diagram shows a rectangular box titled "Manual Scan". Inside the box, there is a section titled "Select Scan Type" with three radio button options: "Quick Scan", "Full Scan", and "Custom Scan". Below these options is a blue button labeled "Start Scan". At the bottom right of the box is a link labeled "English >".

Figure 2 Manual scan



Figure 3 Quarantine



Figure 4 Web Dashboard

3.1.2 Hardware interfaces

The antivirus software, including both the desktop application and the web-based dashboard, does not require any specialized hardware and therefore has no direct hardware interfaces. Operations such as file access, threat detection, and system scanning rely on the underlying operating system to interact with the device's file system and memory. Any connection to cloud-based services or databases is managed through standard network interfaces provided by the host system. Additionally, low-level interactions such as reading or writing to disk and accessing system processes are handled by the OS's built-in APIs and services.

3.1.3 Software interfaces

The antivirus software interacts with the underlying operating system to access system files, processes, and directories in order to perform threat detection and scanning operations, see Figure 2. It also interfaces with the database to store and retrieve information about identified threats, quarantined files, and scan reports. The desktop application performs both read and write operations on the database, such as logging new threats or updating quarantine status. The web dashboard primarily reads data from the database to display reports and system statistics to administrators, see Figure 4.

3.1.4 Communications interfaces

The communication between the different components of the antivirus system—such as the desktop application, the database, and the web dashboard—is essential as they rely on one another for proper functionality. However, the specific method by which this communication is implemented is abstracted away from the application itself and is managed by the underlying operating systems and networking protocols of the user's machine and the server infrastructure. This ensures seamless data transfer for operations like scanning, quarantining, and reporting without requiring the application to directly manage the communication layer.

3.2 Functional requirements

This section outlines the essential functions that the antivirus software must perform for its users.

3.2.1 User Class 1 - The User

3.2.1.1 Functional requirement 1.1

ID: FR1

TITLE: Download the Antivirus Software

DESC: The user must be able to download the antivirus software from an official source such as a website or app store. The software should be available free of cost.

RAT: To ensure users can install and access the antivirus software

DEP: None

3.2.1.2 Functional requirement 1.2

ID: FR2

TITLE: Check for and Notify About Software Updates

DESC: The software should periodically check for new or updated versions and notify the user. The user should be able to download and install updates directly from within the application.

RAT: To ensure users are using the latest and most secure version of the software.

DEP: FR1

3.2.1.3 Functional requirement 1.3

ID: FR3

TITLE: User registration - Desktop/Web Application

DESC: Once the antivirus software is installed, users should be able to register by creating an account with a username, password, and email address. Optionally, users can add a phone number for recovery or notifications.

RAT: To allow personalized services and access to advanced features like cloud scanning and device tracking.

DEP: FR1

3.2.1.4 Functional requirement 1.4

ID: FR4

TITLE: User log-in – Antivirus software

DESC: Given that a user has registered, the user should be able to log in to the antivirus application (either desktop or mobile).

The log-in information can be stored locally on the device if the user opts in, allowing future automatic log-in for convenience.

RAT: In order for a user to access and manage their antivirus settings, scan history, and subscription details.

DEP: FR1, FR3

3.2.1.5 Functional requirement 1.5

ID: FR5

TITLE: Retrieve password

DESC: Given that a user has registered, then the user should be able to retrieve his/her password by e-mail.

RAT: In order for a user to retrieve his/her password.

DEP: FR1

3.2.1.6 Functional requirement 1.6

ID: FR6

TITLE: Virus/Malware Scan – Dashboard View

DESC: Given that a user is logged into the antivirus application, the first page shown should be the main dashboard with a prominent option to scan the system.

The user should be able to perform a system scan based on several scan options:

- Quick Scan
- Full System Scan
- Custom Scan (user selects specific folders/files)
- Scheduled Scan

The dashboard should also provide a search bar to look up threats from a threat database (e.g., by virus name, ID, or behavior).

RAT: In order for a user to initiate different types of scans and search known threats from the antivirus database.

DEP: FR4

3.2.1.7 Functional requirement 1.7

ID: FR7

TITLE: Quarantine and Delete Threats

DESC: Users should be able to take actions on detected threats, such as quarantining or deleting infected files. They should also be able to view quarantine history and restore files if needed.

RAT: To manage infected files and secure the device.

DEP: FR6

3.2.1.8 Functional requirement 1.8

ID: FR8

TITLE: Scan Results – List View

DESC: Scan results should be viewable as a detailed list, where each row represents one detected threat.

Each list item should include:

- File name and path
- Threat name/type
- Severity level
- Detected timestamp
- Recommended action
- A “More Info” link

Maximum 100 threats should be displayed at a time, with scrolling or pagination if needed.

If sorted by severity, list order should be:

1. Severity
2. File location
3. Date detected

If sorted by file path or time:

1. File path
2. Severity
3. Date detected

A header should allow users to select different sorting preferences.

The list view should also include a Filter button to refine visible results.

RAT: The way results should be displayed in a list.

DEP: FR6

3.2.1.9 Functional requirement 1.9

ID: FR9

TITLE: Automatic Background Scan

DESC:

The antivirus should run periodic background scans based on default or user-defined schedules, even when the app is not open.

RAT: To provide ongoing protection without manual input.

DEP: FR7, FR8

3.2.1.10 Functional requirement 1.10

ID: FR10

TITLE: Real-Time Protection Toggle

DESC: Users should be able to enable or disable real-time protection from the settings panel. A warning message must appear when disabling this feature.

RAT: To allow users control while ensuring awareness of security implications.

DEP: FR7, FR8

3.2.1.11 Functional requirement 1.11

ID: FR11

TITLE: Scan Result – Graphical View

DESC:

Scan results can be viewed in a map-style interface. Each infected file location is marked with a colored pin. Pins have info links with threat name, severity, and file path. A maximum of 100 threats should be shown. Default zoom level and filtering button should be provided.

RAT: To provide a visual overview of threat locations.

DEP: FR7, FR8

3.2.1.12 Functional requirement 1.12

ID: FR12

TITLE: Threat Navigation and Actions

DESC: Users should be able to tap a pin or list item to open a detailed threat report. Options to quarantine, delete, or ignore should be available.

RAT: To let users manage each detected threat individually.

DEP: FR8

3.2.2 User Class 2 – System Administrator

3.2.2.1 Functional requirement 2.1

ID: FR22

TITLE: Admin Login

DESC: The system administrator should be able to log in using secure credentials through a dedicated admin interface. Multi-factor authentication must be enabled for additional security.

DEP: None

3.2.2.2 Functional requirement 2.2

ID: FR23

TITLE: Manage User Accounts

DESC: The System administrator should be able to create, update, deactivate or delete end-user accounts and manage user privileges.

RAT: To allow administrators to manage user access to the antivirus system.

DEP: FR22

3.2.2.3 Functional requirement 2.3

ID: FR24

TITLE: Configure Scan Policies

DESC: The system administrator should be able to define and enforce system-wide scan schedules, scanning depths, and response actions (e.g., quarantine, delete).

RAT: To ensure consistent antivirus policies across all endpoints.

DEP: FR22

3.2.2.4 Functional requirement 2.4

ID: FR25

TITLE: Generate and Download Reports

DESC: System administrators should be able to generate reports on user activity, detected threats, scan histories, and updates. Reports should be exportable in PDF and CSV format.

RAT: For auditing and compliance purposes.

DEP: FR22

3.2.3 User Class 3 – Developer/Security Analysts

3.2.3.1 Functional requirement 3.1

ID: FR26

TITLE: Access Threat Logs

DESC: Developers or security analysts should be able to access raw and aggregated threat logs, categorized by severity, source, and time of detection.

RAT: To analyze system vulnerabilities and threat patterns.

DEP: Admin access approval

3.2.3.2 Functional requirement 3.2

ID: FR27

TITLE: Submit Malware Definitions

DESC: Developers should be able to submit new malware definitions, either manually or through integration with a malware database or sandbox system.

RAT: To keep the antivirus engine updated with the latest threats.

DEP: FR26

3.2.3.3 Functional requirement 3.3

ID: FR28

TITLE: Perform Threat Simulations

DESC: Security analysts should be able to simulate malware attacks in a sandboxed test environment to assess detection and response.

RAT: For testing and improving system defense.

DEP: FR26

3.2.3.4 Functional requirement 3.4

ID: FR29

TITLE: View Update History

DESC: Developers should be able to view logs of all antivirus signature updates, engine updates, and version changes.

RAT: For version tracking and troubleshooting.

DEP: FR26

3.2.3.5 Functional requirement 3.5

ID: FR30

TITLE: API Access for Integration

DESC: Developers should be able to access a secured API to integrate antivirus data (scan results, logs, etc.) with external systems like SIEM or incident response platforms.

RAT: To enable integration with enterprise security ecosystems.

DEP: Admin approval and FR16

3.3 Performance requirements

The requirements in this section provide a detailed specification of the user interaction with the antivirus software and performance expectations from the system.

3.3.1 Scan Initiation Speed

ID: QR1

TITLE: Scan Initiation Speed

DESC: When a user initiates a scan (quick, full, or custom), the system should begin the scan process immediately with minimal delay.

RAT: To ensure a responsive and smooth user experience when starting a scan.

DEP: None

3.3.2 Scan Completion Time

ID: QR2

TITLE: Scan Completion Time

DESC: The antivirus system should complete a quick scan within 30 seconds and a full scan within 5 minutes for a device with standard files.

RAT: To reduce user wait time and ensure efficient device scanning.

DEP: QR1

3.3.3 Threat Result Rendering in List View

ID: QR3

TITLE: Threat Result List View Usage

DESC: The results displayed in list view should be user-friendly, easy to scroll, sortable, and selecting a threat should require only one click.

RAT: To allow users to efficiently act on scan results.

DEP: None

3.3.4 Threat Result Rendering in Map View

ID: QR4

TITLE: Threat Result Map View Usage

DESC: The results displayed on a threat map must be clearly visible, pins should be clickable with a single tap, and threat information should pop up instantly.

RAT: To provide intuitive threat navigation.

DEP: None

3.3.5 Information Link Behavior

ID: QR5

TITLE: Information Link Usage

DESC: Every threat result must include a clear, clickable "More Info" link that opens the threat description in one click.

RAT: To provide immediate access to detailed threat information.

DEP: None

3.3.6 Response Time for Real-Time Protection

ID: QR6

TAG: ResponseTime

GIST: Delay between detecting a threat in real-time and notifying the user.

SCALE: Response time to detect and alert on threats.

METER: Testing based on 500 real-time simulations.

MUST: No more than 1 second for all detections.
WISH: Under 500ms for 90% of detections.

3.3.7 CPU and RAM Usage During Scan

ID: QR7

TAG: ResourceEfficiency

GIST: The antivirus should not slow down the system during scans.

SCALE: Memory and CPU usage during full scan.

METER: Tests on standard systems with average file loads.

MUST: CPU usage < 60%, RAM usage < 500MB 90% of the time.

WISH: CPU usage < 40%, RAM usage < 350MB.

3.3.8 Update Performance

ID: QR8

TITLE: Update Download and Install

DESC: Antivirus definition updates should download and install within 30 seconds on a 10 Mbps connection.

RAT: To ensure the software stays current with minimal user interruption.

DEP: None

3.4 Design constraints

This section includes the design constraints on the antivirus software caused by the hardware or operating system limitations.

3.4.1 Hard drive space

ID: QR9

TAG: HardDriveSpace

GIST: Storage space occupied by the antivirus application.

SCALE: The application's need for hard drive space.

METER: MB

MUST: No more than 100 MB

PLAN: No more than 80 MB

WISH: No more than 50 MB

MB: DEFINED: Megabyte

3.4.2 Application memory usage

ID: QR10

TAG: ApplicationMemoryUsage

GIST: The amount of system memory (RAM) used by the antivirus application during normal operation.

SCALE: MB

METER: Observations from the performance log during testing

MUST: No more than 200 MB

PLAN: No more than 150 MB

WISH: No more than 100 MB

MB: DEFINED: Megabyte.

3.4.3 Operating System Compatibility

ID: QR12

TAG: OSCompatibility

GIST: Platforms on which the antivirus software must run.

SCALE: Desktop and mobile operating systems.

METER: Execution and functionality tests across platforms

MUST: Windows 10+, Linux (Ubuntu 20.04+), macOS 11+

PLAN: Android 10+, iOS 14+

WISH: Cross-platform support with UI consistency

ID: QR12

TAG: OSCompatibility

GIST: Platforms on which the antivirus software must run.

SCALE: Desktop and mobile operating systems.

METER: Execution and functionality tests across platforms

MUST: Windows 10+, Linux (Ubuntu 20.04+), macOS 11+

PLAN: Android 10+, iOS 14+

WISH: Cross-platform support with UI consistency

3.4.4 Background Process Limitations

ID: QR13

TAG: BackgroundProcessEfficiency

GIST: Restrictions on background scans to avoid system slowdowns.

SCALE: CPU and I/O impact during background scanning.

METER: Tests on systems with limited resources

MUST: Background scan must not exceed 40% CPU usage

PLAN: No more than 25% CPU usage

WISH: Adaptive scan that runs only during idle CPU usage

3.5 Software system attributes

The requirements in this section specify the required reliability, availability, security, maintainability, and portability of the antivirus software system

3.5.1 Reliability

ID: QR9

TAG: SystemReliability

GIST: The reliability of the antivirus system.
SCALE: The accuracy and consistency of threat detection during scans.
METER: Measurements obtained from 1000 scan sessions during testing.
MUST: More than 98% of threats correctly identified.
PLAN: More than 99% of threats correctly identified.
WISH: 100% detection accuracy.

3.5.2 Availability

ID: QR7

TAG: SystemAvailability

GIST: The availability of the antivirus system when used by users.
SCALE: The average system uptime (excluding device or network failures).
METER: Measurements from 1000 hours of operation during testing.
MUST: More than 98% uptime.
PLAN: More than 99% uptime.
WISH: 100% uptime.

ID: QR22

TITLE: Internet Connectivity

DESC: The system should be connected to the Internet to fetch updates, definitions, and sync scan logs.
RAT: Required to ensure real-time protection and cloud-assisted scanning.
DEP: none

ID: QR23

TITLE: System Services Access

DESC: The application should have access to system services like file system, background tasks, and notifications.
RAT: Required for real-time protection and scheduled scans.
DEP: none

3.5.3 Security

ID: QR12

TAG: DataEncryption

GIST: Secure communication between client, server, and scan logs.
SCALE: Encryption of all sensitive information including user data and logs.
METER: 1000 encrypted sessions analyzed during testing.
MUST: 100% of sessions must use secure protocols like HTTPS/TLS.

ID: QR13

TAG: UserLoginSecurity

GIST: Secure login process for end users.
SCALE: Login attempts using invalid credentials.

METER: 1000 login attempts during testing.
MUST: 100% of invalid attempts must be blocked and logged.

ID: QR14

TAG: AdminLoginSecurity
GIST: Secure login validation for system administrators.
SCALE: Login attempts with non-existent accounts.
METER: 1000 attempts tested.
MUST: 100% of invalid admin login attempts must be blocked.

ID: QR15

TAG: UserAccountLockout
GIST: Lock user account after multiple failed attempts.
SCALE: Attempts after account lockout threshold (3 failures).
METER: 1000 blocked attempts.
MUST: Lockout for 30 minutes and disable login temporarily.

ID: QR16

TAG: AdminAccountLockout
GIST: Lock admin account after multiple failed login attempts.
SCALE: Attempts after lockout threshold (3 failures).
METER: 1000 login trials during lockout.
MUST: Lockout for 30 minutes and login should remain disabled.

ID: QR17

TAG: SecureUserRegistration
GIST: Prevent duplicate usernames during registration.
SCALE: Username conflict scenarios.
METER: Observed over 1000 account creation sessions.
MUST: 100% prompt to choose a different username if taken.

ID: QR18

TAG: DevAccessControl
GIST: Control access to sensitive logs and configurations.
SCALE: Unauthorized access attempts by unverified developers.
METER: 1000 test access attempts.
MUST: 100% unauthorized access must be denied and logged.

3.5.4 Maintainability

ID: QR19

TITLE: Code Modularity

DESC: The antivirus software should be modular and easy to extend.

RAT: To enable future enhancements like new detection engines or dashboards.

DEP: none

ID: QR21

TITLE: Test Infrastructure

DESC: Unit and integration test suites should be available for all system modules.

RAT: For regular and automated testing of features.

DEP: none

3.5.5 Portability

ID: QR20

TITLE: Cross-Platform Support

DESC: The antivirus should work across major platforms (Windows, Linux, Android).

RAT: For a wider user base and better market adaptability.

DEP: none