

- Required APIs to be developed for different objectives (internal integration, integration with external applications, integration with the other administrations).
- Required data migration from existing applications.
- Required reporting.

Deliverables:

- **BRD** – Business Requirements Document.

Acceptance:

- Validation of the BRD by all NSSF Directorates.

7.4 SOLUTION DESIGN AND TECHNICAL ARCHITECTURE

Definition:

- The solution designed is based on the Needs Assessment and Business Requirements' phase.
- The software components and their integration will be defined, finalized, and approved.

Architecture Objectives:

- Design of the Technical Architecture of the future Software Applications based on the following requirements:
 - Unified technical environment.
 - Unified and reliable applicative security.
 - Unified Integration between different modules.
 - Unified Integration between external stakeholders and the NSSF.
 - Unified and dynamic reporting.
 - Digital Services Readiness through secure APIs.

Main Tasks:

- Define and provide all software components and activities that are necessary to complete the full installation, configuration, integration, and proper functioning of the software applications.
- Define which modules are to be developed and which ones can be based on available ready-made applications in the market to fulfill all requirements.
- Specify the required infrastructure for the deployment of the future Software Applications.
- Define a detailed implementation plan of future Software Applications.



Deliverables:

- **TAD** – Technical Architecture Document.

Acceptance:

- Validation of the TAD by the NSSF.
- Compliance with non-functional requirements (see paragraph 99).

7.5 IMPLEMENTATION – CODING AND TESTING

Environments Preparation:

- Development Environment which could be hosted in the Company premises or in the NSSF Infrastructure.
- Testing Environment to be built in the NSSF Infrastructure.
- UAT (User Acceptance Testing) Environment to be built in the NSSF Infrastructure.
- Live Environment to be built in the NSSF Infrastructure.

Technical Design, Development, and Testing:

- **TDD – Technical Design Document:**
 - Once the **BRD** and the **TAD** are approved, the Company must prepare a comprehensive **TDD** (Technical Design Document) that will contain all physical design related elements (database structure, schematics, maps, diagrams and flow charts, screen shots, etc.). The technical design must include a detailed working prototype model of the new systems as a proof of concept. The technical design documents must address functional and design requirements of the following: Workflows, Documents Management, Integration with existing applications, APIs, and Web Services.
- **Coding and Unit Testing:**
 - Coding and Unit Testing of all components based on the exact and complete approved requirements defined in the **BRD**, the **TAD**, and the **TDD**.
 - Perform code reviews to ensure consistency and efficiency.
 - Provide the developed and/or customized full Source code.
 - Ensure the development be done in accordance with best practices.
 - Install the developed components on the Testing Environment.
 - In case ready-made applications are adopted for some functions and based on the approved requirements defined in the **BRD**, the **TAD**, and the **TDD**.
 - Customize the applications, where needed.

- Develop the needed interfaces to integrate them with the above developed programs.
- Install the customized applications on the Testing Environment.

- **Functional Testing:**

- Prepare the **FTP** (Functional Testing Plan) including testing data, tests to be executed, testing methodology and testing reporting.
- Prepare the Testing Environment.
- Prepare the functional data for the testing.
- Fix any errors, bugs, and problems to ensure that the system components function together properly and that the project implementation meets the business requirements.
- Modify, add, and/or omit source code as required to fix errors, bugs, and problems.
- **Performed and Validated by Functional NSSF Staff**
 - Execute/Re-execute all tests based on the FTP.
 - Update the FTP with added scenarios.
 - Elaborate the testing reports.
 - Declare all bugs, defects, issues, and problems to be fixed by the Company.
 - Document all test results as testing occurs.
 - Approve and Sign the **Functional Testing Acceptance Form**.

- **Technical Testing**

- Prepare the **TTP** (Technical Testing Plan) including testing data, tests to be executed, testing methodology and testing reporting.
- Prepare the Testing Environment.
- Prepare the technical data for the testing (performance, security, integration, load balancing, others)
- **Performed and Validated by NSSF Technical Staff**
 - Execute all tests based on the TTP.
 - Update the TTP with new scenarios.
 - Declare all bugs, defects, issues, and problems to be fixed by the Company.
 - Elaborate the testing reports.
 - Approve and Sign the **Technical Testing Acceptance Form**.



- **Installation/Configuration:**

- Install and Configure the whole system to be ready for the data migration.

Deliverables:

- **TDD** – Technical Design Document.
- **FTP** – Functional Testing Plan.
- **TTP** – Technical Testing Plan.
- Functional Testing Report.
- Technical Testing Report.
- All custom and non-custom software source code.
- Backup/Restore Plan.
- Users Guide.
- Deployment Guide.

Acceptance

- Acceptance of all deliverables.
- Approval and signature of the Functional Testing Acceptance Form.
- Approval and signature of the Technical Testing Acceptance Form.
- Compliance with non-functional requirements (see paragraph 99).

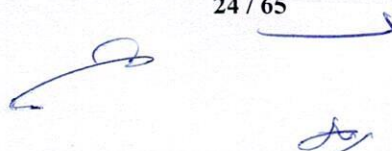
7.6 DATA MIGRATION

Definition:

- Migrate existing data to the new designed and implemented database.

Main Tasks:

- **Data Mapping:**
 - Identification of data to be migrated
 - Identification of the existing data structure
 - Identification of the targeted data structure
 - Definition of mapping rules from the existing to the new system
- **Data Cleansing:**
 - Control of all mapped data
 - Unification of all coding tables



- Cleansing of duplicated data
- **Elaboration of the migration plan including:**
 - Existing data structures
 - Targeted data structures
 - Mapping rules
 - Cleansing rules
 - Testing and data migration integrity controls methodology.
- **Development and testing of the data migration.**
- **Full migration of data and controlling integrity of migrated data.**

Deliverables:

- Migration Plan.
- Migration Report.
- Data migration integrity controls.

Acceptance:

- Validation of the Migration Plan by the NSSF.
- Validation of the Data Migration based on the Migration Plan by the NSSF.

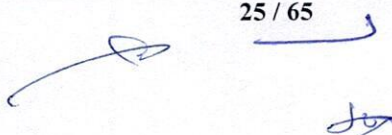
7.7 UAT (USER ACCEPTANCE TESTING) – PARALLEL RUN

Definition:

- Allow the end users of Directorates to run the new applications in parallel with the existing ones to ensure that all functionalities are implemented as required.

Main Tasks:

- Prepare the UAT Environment with the migrated data.
- Run the new applications in parallel with the existing ones to ensure that all functionalities are implemented as required.
- Fix any errors, bugs, and problems to ensure that the system components function together properly and that the project implementation meets the business requirements.
- Modify, add, and/or omit source code as required to fix errors, bugs, and problems.
- **Performed and Accepted by NSSF End Users delegated by all NSSF Directorates**
 - Perform the same actions by the existing application and the new one.
 - Declare all bugs, defects, issues, and problems to be fixed by the Company.



- Approve and Sign the **UAT Acceptance Form**.

Acceptance:

- Approval and signature of the **UAT Acceptance Form** by all Directorates.
- Compliance with non-functional requirements (see paragraph 99).

7.8 DEPLOYMENT AND POST DELIVERY SUPPORT

Definition:

- Allow the **NSSF IT Support Team** to support end users to better understand and use the new applications.

Main Tasks:

- Train the **NSSF IT Support Team** to the new applications.
- Define with the **NSSF IT Support Team** a schedule to start using the new applications.
- Ensure a closed support to the **NSSF IT Support Team** during the two months after delivery.
- Ensure the maintenance of the new applications before moving them to the operations.

Acceptance:

- Approval and signature of the **NSSF IT Support Team** to support the new applications.

7.9 CHANGE MANAGEMENT

7.9.1 INTRODUCTION




Successful implementation of a software applications project needs necessarily to be accompanied by a well-defined change management process.

This process will handle all 'non-technical' tasks that will ensure that adequate communication, coordination, training, awareness, preparing the changes... are properly managed through the whole lifetime of the project.

7.9.2 AWARENESS

Main Tasks:

- Prepare two presentations: one for the top management explaining the expected objectives and benefits of the project (presentation of one hour), and one for the Directors and Heads of Branches explaining the expected objectives and benefits of the project and the different impacted processes in the organization.



- Identify, with the Directorates, the individuals that will facilitate the process and lead the endeavor. Most change systems acknowledge that knowing what to improve creates a solid foundation for clarity, ease, and successful implementation.

Deliverables:

- Presentation for the top management.
- Presentation for the Directors and Heads of Branches.
- Minutes of Meetings with the top management.
- Minutes of Meetings with the Directors and Heads of Branches.

Acceptance:

- Approval of Deliverables.

7.9.3 PLAN FOR CHANGE**Main Tasks:**

- Meet with the identified individuals to present them the project (the same presentation for the Directorates and the Heads of Branches) and define their roles in the change management process:
 - **Project Presenter:** make specific presentations for employees.
 - **Business Requirements Facilitator:** assist the implementation team to collect information and validate processes during the needs assessment and business analysis phase.
 - **User Interface Validator:** organize presentation sessions to know about the user interface design and get feedback from the end users.
 - **Testing Assistant:** assist the implementation team to test and validate the different functionalities of the project.
- Plan the different interventions for the next phases of the project and define the associated milestones to be tracked by the Project Manager

Deliverables:

- Interventions Planning.
- Interventions Progress Report.

Acceptance:

- Approval of Deliverables.



7.9.4 COMMUNICATION

Main Tasks:

- Prepare the different planned intervention during the project, get the feedback from the end users, and integrate the needed changes in the project to the initial scope.
- Define the new demands not included in the initial scope and send to the top management for decision (to be added to the project scope or postponed to other releases of the application). If the decision is to add, a change request should be initiated for the extension of the project.

Deliverables:

- Interventions Planning.
- Minutes of Meetings with the End Users.

Acceptance:

- Approval of Deliverables.

7.9.5 TRAINING

Main Tasks:

- The overall objective of the training phase is the training of users on new skills and knowledge needed to operate the newly delivered products. The main activities of this phase:
 - Prepare a training plan including all training sessions, training materials, training logistics, training delivery, attendance, and evaluation forms.
 - Prepare a user-friendly training material that uses simple language, contains graphical depictions of specific procedures (i.e., flow charts), includes FAQs, and standard forms and examples of completed forms for: End Users, Super Users, and Administrators.

Deliverables:

- Training Plan.
- Training materials for End Users, Super Users, and Administrators.
- Delivery of the training sessions.
- Training Reports.
- Attendance Forms.
- Evaluation Forms.

Acceptance:

- Approval of Deliverables.
- Control of Attendance Forms.
- Level of satisfaction of the Evaluation Forms > 80%.

7.9.6 REVIEW, REVISE, AND CONTINUOUSLY IMPROVE**Main Tasks:**

- Organize a meeting for the lessons learned for the organization.
- Define the new roles to be added for the deployment of the application.
- Define the next steps to be done.

Deliverables:

- Minutes of Meetings for the lessons learned.
- Recommendations for the next steps.

Acceptance:

- Approval of Deliverables.

8. PHASE 04 – HANDOVER FROM THE COMPANY TO THE NSSF

Definition:

- Develop and execute a handover plan to ensure the success of the transition between the Company and the NSSF or any other party specified by the NSSF.

Main Tasks:

- Assign the needed resources from the NSSF or from a delegated third party to participate to the handover phase at the beginning of the contract.
- Develop the handover plan between the NSSF team and the Company team.
- Execute all tasks of the handover plan and finalize them with the signature of both parties.

Deliverables:

- Handover Plan.
- Handover Acceptance Sheets.

Required Service Level Agreements:

- Availability of the Transition Team from the NSSF and the Company.

Acceptance:

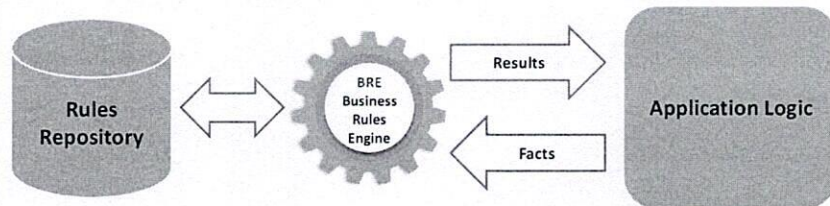
- Handover Acceptance Sheets Signed by both parties.



9. TECHNICAL COMPONENTS REQUIREMENTS

9.1 BUSINESS RULES ENGINE REQUIREMENTS

The NSSF will need to have integrated a business rules engine with the following requirements:

Requirement ID #	Requirement
TC-BRE-01	Business Rules Engine tool must be open source: No license or maintenance should be required to use this tool.
TC-BRE-02	BRE should be generic: the BRE tool should be utilized by different future integration with the NSSF (hospitals, pharmacies, doctors, administrations, etc.).
TC-BRE-03	BRE Architecture Components: the main architecture components of the BRE tool should contain at least the following: <ul style="list-style-type: none"> Rules repository that stores business rules. Data store that hosts customer records or customer master. Processing engine. Data store to capture the results of the processing and maintain a member register. End-to-end workflow orchestration mechanism.
TC-BRE-04	BRE Architecture: the BRE Architecture should be based on at least the following: <div data-bbox="497 1382 1324 1586" data-label="Diagram">  <pre> graph LR A[(Rules Repository)] <--> B((BRE Business Rules Engine)) B -- Results --> C[Application Logic] C -- Facts --> B </pre> <p>The diagram illustrates the BRE Architecture. It consists of three main components: a 'Rules Repository' (represented by a cylinder), a 'BRE Business Rules Engine' (represented by a gear), and 'Application Logic' (represented by a rounded rectangle). The 'Rules Repository' and the 'BRE Business Rules Engine' are connected by a double-headed arrow, indicating bidirectional communication. The 'BRE Business Rules Engine' sends 'Results' to the 'Application Logic' via a right-pointing arrow. Conversely, the 'Application Logic' sends 'Facts' back to the 'BRE Business Rules Engine' via a left-pointing arrow.</p> </div>

Example: DROOLS (<https://www.drools.org/>).

Drools is a Business Rules Management System (BRMS) solution.

It provides a core Business Rules Engine (BRE), a web authoring and rules management application (Drools Workbench), full runtime support for Decision Model and Notation (DMN)

models at Conformance level 3 and an Eclipse IDE plugin for core development.

Drools is open-source software, released under the Apache License 2.0. It is written in 100% pure Java™, runs on any JVM and is available in the Maven Central repository too.

The rules engine by design is configurable, allowing business rules that can be configurable and maintained with rule-specific metadata for instance type of rule, category, and so on. The rules evaluation results are enriched and captured with details to maintain history to understand the member journey.

9.2 API MANAGEMENT PLATFORM REQUIREMENTS

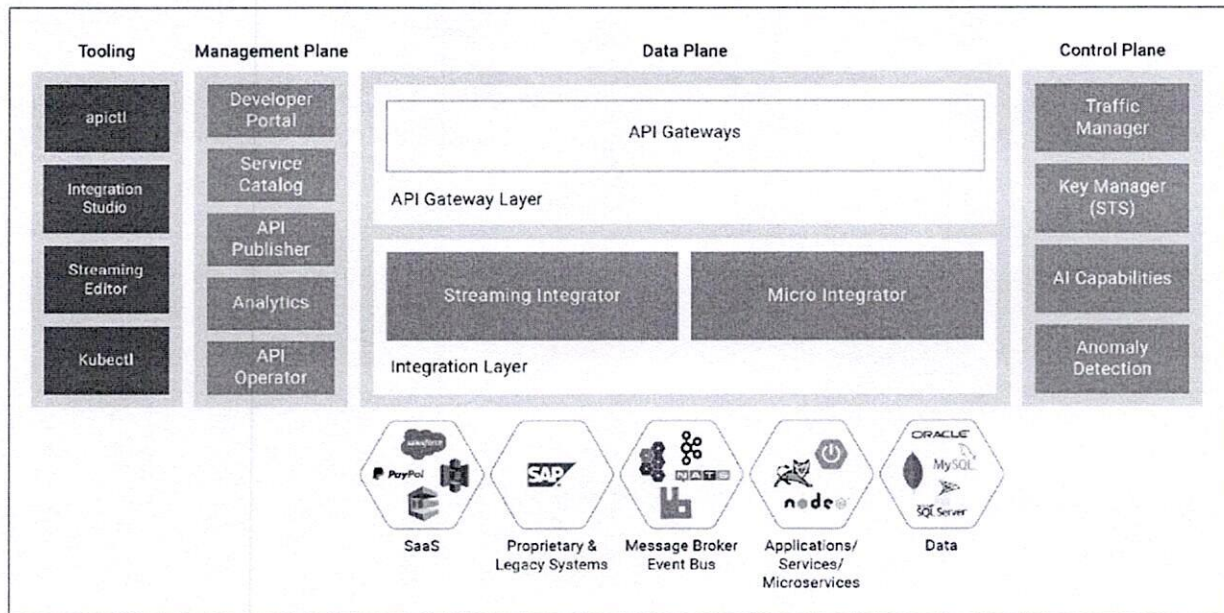
The NSSF will need to have integrated an API Management Platform with the following requirements:

Requirement ID #	Requirement
TC-APIMP-01	API Management Platform tool must be open source: No license or maintenance should be required to use this tool.
TC-APIMP -02	API Management Platform should be generic: the API Management Platform tool should be utilized by different future integration with the NSSF (hospitals, pharmacies, doctors, administrations, etc.).
TC-APIMP-03	Architecture components of API Management Platform: the main architecture components of the API Management Platform tool should contain at least the following: <ul style="list-style-type: none">• API Publishing.• API Developer Portal.• Service Catalogue.• API Analytics.• API Gateway.• Key Manager.• Traffic Manager.

Example: WSO2 API Manager (<https://wso2.com/api-manager/>).

WSO2 API Manager is a market-leading full lifecycle API management platform for building, integrating, securing, and exposing an enterprise's digital services as managed APIs in cloud, on-premises, and hybrid architectures. Fast-track your API strategy with all the capabilities needed by API designers, product managers, operations, and consumers.

From simple scenarios to comprehensive protocol support, WSO2 API Manager can handle it all. Use industry standards or extend our platform to integrate with your existing business needs, applications, and architectures. Tailor it to your exact use case, whether that means customizing the user interface, mediation, security, or integrating third-party solutions.



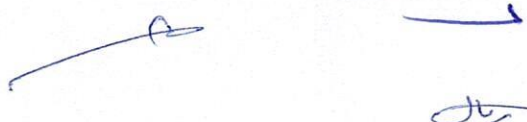
10. NON-FUNCTIONAL REQUIREMENTS

The below non-functional requirements must be taken into consideration during the design and implementation of the software applications:

10.1 SECURITY REQUIREMENTS

The new applications should abide by the following **Security Requirements**:

Requirement ID #	Requirement
NF-SEC-01	Security by design is to be applied: security by design is an approach to software development that seeks to make systems as free of vulnerabilities and impervious to attack as possible through such measures as continuous testing, authentication safeguards and adherence to best programming practices. Security should be built into the applications by design, instead of being added later. The applications must be designed to be secure from the ground up and built in such a way as to minimize flaws that could compromise security.
NF-SEC-02	Data privacy by design: identification, authentication, authorization, integrity, confidentiality, privacy, and non-repudiation requirements.
NF-SEC-03	<p>Addressing OWASP 10 Web Application Security Risks. The application should address OWASP 2021 top 10 Web application security risks:</p> <ul style="list-style-type: none"> • A01:2021-Broken Access Control. • A02:2021-Cryptographic Failures. • A03:2021-Injection. • A04:2021-Insecure Design with a focus on risks related to design flaws. • A05:2021-Security Misconfiguration. • A06:2021-Vulnerable and Outdated Components. • A07:2021-Identification and Authentication Failures. • A08:2021-Software and Data Integrity Failures. • A09:2021-Security Logging and Monitoring Failures. • A10:2021-Server-Side Request Forgery.



Requirement ID #	Requirement
NF-SEC-04	Clear security architecture should be designed and fully documented.
NF-SEC-05	Link with the Active Directory: users to be defined under the applications should be already defined under and enabled in the Active Directory of NSSF. Users' information can be imported from the Active Directory, groups in the Active Directory can be imported in the applications along with the member users, ... But no single sign-on is to be used.
NF-SEC-06	<p>Users' Authentication Means / Complex passwords requirements (to be set as parameters that can be managed by the system administrators – that can be imported from the security policy of the Active Directory):</p> <ul style="list-style-type: none"> • Minimum length • Mandatory complexity of used characters (lower and upper cases, numbers, special characters). • Lifetime duration in days. • Automatic warning: the system should notify the user prior to the expiry date after which the system should force the user to enter new password credentials, but user is not deactivated the account. The number of days is a parameter. • History of used passwords: a number N that will deny using the last N passwords. • Passwords should be encrypted with no possibility of reverse encryption. • Number of failed log-ins. • No repetitive or sequential characters... should be accepted in the passwords.
NF-SEC-07	Multi-factor authentication option for validating critical transactions and specific cases.

Requirement ID #	Requirement
NF-SEC-08	<p>Roles and profiles are to be managed within the applications:</p> <ul style="list-style-type: none"> • Access rights are defined by roles. • Users are assigned to role(s) whereby they will inherit the privileges access rights. • No exceptions to the roles should be allowed for the users. In case needed, a new role should be created. • Users are assigned managers to which escalation of transactions for validation can be done. • Roles can be also assigned limits whereby they cannot validate transactions with higher amounts than the defined limits. A workflow with an escalation to users with higher limits is to be managed.
NF-SEC-09	<p>Log files, audit trails and traceability. Log files should be available for all actions performed on the applications:</p> <ul style="list-style-type: none"> • Granting, revoking, modifying, disabling, temporary disabling, ... of a role, and / or a user. • Changing a password. • Entering a transaction, validating a transaction. • Modifying the value of a business parameter, ... • Errors handling. The primary objective of error handling and logging is to provide useful information for the user, administrators, and incident response teams. The objective is not to create massive amounts of logs, but high-quality logs. • Log files should be secure and unmodifiable. • Log files should show at least: user, time stamp (date and time), type of action performed, value before and value after, IP address of the workstation, ...



Requirement ID #	Requirement
NF-SEC-10	<p>Maker / checker or 4-eyes principles:</p> <ul style="list-style-type: none"> For some critical cases, a maker / checker or 4-eyes principle should be applied. Examples: <ul style="list-style-type: none"> Defining business parameters, Validating transactions with high amounts, Managing users' access rights.... Defining workflows, ... While defining the users' profile, makers and checkers are assigned. A checker cannot validate transactions that he / she made.
NF-SEC-11	<p>Secure communication and connectivity: the connectivity between the browsers and the core application should be secure through the most recent version of TLS encryption, independent of sensitivity of the content.</p>
NF-SEC-12	<p>Compatibility with all browsers and platforms: the applications should be compatible with all the market browsers (Google Chrome, Firefox, Edge, Safari, ...).</p>
NF-SEC-13	<p>Security reports should be made available, e.g.:</p> <ul style="list-style-type: none"> Roles / profiles access rights. Users' access rights. Log files.
NF-SEC-14	<p>APIs and Web Services: ensure that the application that uses trusted service layer APIs has:</p> <ul style="list-style-type: none"> Adequate authentication, session management and authorization of all web services. Input validation of all parameters that transit from a lower to higher trust level. Effective security controls for all API types.

Requirement ID #	Requirement
NF-SEC-15	<p>Sessions Management: one of the core components of any web-based application or API is the mechanism by which it controls and maintains the state for a user or device interacting with it. It should ensure that the application satisfies the following high-level session management requirements:</p> <ul style="list-style-type: none"> Sessions are unique to each individual and cannot be guessed or shared. Sessions are invalidated when no longer required and timed out during periods of inactivity.
NF-SEC-16	The solution architecture should have local high-availability features where no component within the main data center should be a single point of failure.
NF-SEC-17	The solution architecture for WAN connectivity should be redundant to void any network disruption between any branch and the main data center.
NF-SEC-18	The solution architecture for should cater for on-line real-time asynchronous replication between the main data center and the DRS.
NF-SEC-19	Steps to follow for the fail-over and fall back between the main data center and the DRS should be clearly described and easy to follow.

10.2 METHODOLOGY REQUIREMENTS

The new applications should abide by the following **Methodology Requirements**:

Requirement ID #	Requirement Description
NF-MET-01	The Company shall provide a clear description of the methodology and technical approach for performing the assignment and implementing the tasks which will deliver the expected outputs.
NF-MET-02	The Company shall demonstrate, through their proposal, their understanding of the objectives of the assignment and how it is planned to achieve the same.

Requirement ID #	Requirement Description
NF-MET-03	The Company shall outline the work plan for the implementation of the main activities/tasks foreseen in the assignment, their content and duration, phasing and interrelations, milestones.
NF-MET-04	The warranty period for delivered solutions/components/systems shall last for a period of one year from project go-live.
NF-MET-05	Warranty shall include repair of detected bugs, security patches and any verified problems/issues identified in the solutions.
NF-MET-06	The Company shall define and deliver a quality control process to ensure and maintain the quality of the deliverables.

10.3 ARCHITECTURE REQUIREMENTS

The new applications should abide by the following **Architecture Requirements**:

Requirement ID #	Requirement Description
NF-ARCH-01	The solutions and any included components, as necessary, shall support the possibility of system installation on two or more servers , allowing extension of existing system, scalability, and better availability.
NF-ARCH-02	The solutions shall be scalable and support possibility of increasing number of users and functionalities, without diminishing overall performance.
NF-ARCH-03	The solutions and accompanying components must ensure Audit trail functionality , as electronic chronological records representing documented records of activities which have been carried out (operations, events, transactions, loggings, etc.).
NF-ARCH-04	The solution should be web based and support future mobile applications in secure and reliable way.

10.4 TECHNOLOGY REQUIREMENTS

The new applications should abide by the following **Technology Requirements**:

Requirement ID #	Requirement Description
NF-TECH-01	All technical components and products should be based on the latest technologies of Microsoft Solutions and Platform .
NF-TECH-02	Solution Design can be based on specific development or available in the market ready-made applications with their customization.
NF-TECH-03	APIs should be based preferably on Secure RESTful API technology .

10.5 CHANGE REQUESTS MANAGEMENT REQUIREMENTS

The new applications should abide by the following **Change Requests Management Requirements**:

Requirement ID #	Requirement Description
NF-CHREQ-01	A Change Requests Management will start after the acceptance of the BRD (Business Requirements Management) of each applications or group of applications.
NF-CHREQ-02	A Change Request is defined with one of the following options: <ul style="list-style-type: none"> Enhancements of the existing processes, existing procedures, existing functionalities, and/or business rules. New processes, procedures, functionalities, and/or business rules.
NF-CHREQ-03	A Change Request is prepared by the end users (NSSF Directorates), validated by the NSSF Management, and sent to the Company for analysis.
NF-CHREQ-04	Each Change Request is analyzed by the Company and sent back to the NSSF Management. The analysis contains the understanding of the required changes and if it is part of the current scope of the project or not.