Česká rozvojová agentura

ČESKÁ REPUBLIKA POMÁHÁ

Technological terms and conditions

Phytosanitary register in Georgia

Ústřední kontrolní a zkušební
ústav zemědělský
Oddělení komunikace a
zahraniční spolupráce

Hroznová 63/2
656 06 Brno, Czech Republic

Lukáš Plachý
IT architect

lukas@plachy.eu

Šaumannova 3867/1
615 00 Brno, Czech Republic

Document information

Document name:	**Technological terms and conditions**		
Project name:	**Phytosanitary register in Georgia**	**Document version:**	**1.2**
Project phase:	**-**	**Version date:**	**05.02.2021**

Version and revision log:

Version no.	**Version date**	**Author**	**Description**
0.1	20.1.2019	Lukas Plachy	Proposed
0.2	18.2.2019	Lukas Plachy	Proposed, glossary added, phytosanitary/plant passport terms unified, risks format change.
0.3	11.05.2020	Lukas Plachy	Proposed, added as per NFA requirement architectural references and requirements for the use of SitiAGRI platform, other technical references have been checked and updated to match the SitiAGRI platform as closely as possible.
1.0	15. 9. 2020	Lukas Plachy	Finalization as ver. 1.0
1.1	19. 11. 2020	Petr Vaculik Radek Meluzin	Chap. 9.4.3. M2 deadline changed – calendar date omitted (based on law/administratively discussion)
1.2	05. 02. 2021	Lukas Plachy	Pre-release typo corrections

Table of Contents

Terms and abbreviations

Term	Meaning
AAA	Authentication, Authorization and Audit
ACL	ACcess List - a list of users and their authorizations.
AD / MS AD	Microsoft Active Directory - a set of Applications and services used to control users, computers, resources and their roles and access within a so called "Windows Domain" network.
ADSL	Asymmetric Digital Subscriber Line - a data communications technology that enables data transmission over copper telephone lines.
API	Application Programming Interface - An interface designed to be access, deserialised and used by automated electronic means such as computer programs etc.
Application	A software application used to support some process.
AR	Architectural requirement / Architecture requirement
ArchiMate	A notation for software/hardware/orgware inventarization, organization and design.
B/Byte **MB/MegaByte**	Unit for information data size.
ČR	Česká republika - a country in central Europe.
ČRA	Česká rozvojová agentura - Czech development agency - and organizational unit of the Czech government in ČR, funding specific projects.
CVS	Concurrent Version(ing) System – a generic software category of a software system for storage of (usually textual - software high-level language source code) data, handling concurrence access and versioning.
data entity	A data record/sentence, whose attributes relate to the same real-world object.
DB	Database - may refer either to the data itself or an entire RDBMs and its services.
dep.	Department
Deployment	Making some application or entire IT system operational and available for its users to use its functionalities.
E-SSO	Enterprise Single-Sign-On - a (SW) solution for centralized user authentication and authorization management within a network/organization.
EU	European union
Excel	Microsoft Office Excel - software for spreadsheet processing.
file	Set of electronic data belonging together
FR	Functional requirement
Georgia	A country in the Caucasus region of Eurasia. Bounded to the west by the Black Sea, to the north by Russia, to the south by Turkey and Armenia, and to the southeast by Azerbaijan.
GUI	Graphical User Interface - interface of a software designed to be used by humans by displaying information in a graphical form typical for common inter-human understanding.
HW	Hardware
Hyper-V	Virtualization environment of the Windows platform (MS) used to run multiple "virtual" computers on one single computer hardware.
I/O	Input/Output

ID	Identifier / identification number
Internet	An open computer network on a global level, using common protocols, administered by IANA and its sub-accreditees.
IR	Integration requirement
IT	Information technology
LDAP	Lightweight Directory Access Protocol - a protocol used to access tree-organized data structures, typically structures of user account data (such as MS AD).
LOD	Leve-of-detail - how detailed is the subject depicted.
LR	(software) Legal requirement
module	A part of application
MR	(data) Migration requirement
MS	Microsoft - a major global software provider.
MVC	Model-View-Controler - a programming paradigma e.g. how a software application source code is internally organised and understood by it programmers.
Mx	Project milestone
network	Multiple computers and other electronic devices interconnected together by a common electric and logic protocol.
NFA	National food agency - a legal entity under the public law (order No 2-3 of Ministry of Agriculture of Georgia, from the 14th of January 2011), organizationally subsumed to the Ministry of Agriculture of Georgia. Registration code of business subject at the Ministry of justice: 040.030.000.22.032.016.717
OR	Operating / Operational requirement
orgware/OW	Set of organizational measures (administrative decrees, people, their roles and responsibilities etc.) achieving some common goal.
OS	Operating system - basic application to provide user interaction with a computer, especially then the service of loading and launching other applications.
OU / MS AD OU	Organizational unit - a logical data unit for holding data of an MS AD
PA	Project action requirement
PC	Project cooperation requirement/specification
person	A business (business person) or a physical (physical person) entity, usually conducting a kind of business, related to phytosanitary area, thus being registered (e.g. the person's data recorded) in the system.
phytosanitary	(Botany) of or relating to the health of plants, also measures undertaken or intended to ensure that an imported plant or collection of plants is free of specified pests.
PL	(project) Legal requirement
PM	Project management / Project management requirement
PO	Project Output requirement
PR	Performance requirement
px	pixels - a basic unit of graphical display of a computer output on a computer screen
Q/A	Question / Answer
RAM	Random access memory - a fast volatile computer memory holding data and software being currently executed.
RDBMS	Relational database management system - an IT Application used to store and retrieve data entities interconnected together (referenced between themselves).

Register	In the scope of this document: the aggregate of technical, organizational and other measures to provide the record and evidence of the data in question.
script	How various characters/letters are depicted on a computer screen (based on their originating handwriting such as "Latin", "Cyrillic" etc.)
SEA	Sparx Enterprise Architect – modelling software
SLA	Service level agreement - measurable parameters how fast / in what deadlines some service should be delivered.
SQL	Structure Query Language - a language specified and maintained by the appropriate authority used to manipulate data and data structures in RDBMs.
SR	Security requirement
SW	Software
Tender###	Tender proposal description requirement
TR	Testing requirement
UKZUZ (alt.: UZUS)	Ústřední kontrolní a zkušební ústav zemědělský - Central institute for control and testing in agriculture - an organizational unit of the Czech government in ČR.
UML	A notation used for software design.
UR	User requirement (non-functional requirement applicable the entire application).
WS	Web-service - a type of API accessible over a network.
XMI	XML Metadata Interchange – an OMG (Object Management Group consortium) standard for exchanging modelling metadata information

1. System to-be - tender requirements - System overview and purposes

General overview

The implementation of the Phytosanitary register helps to establish a favourable environment for the transition to the information society and to provision of IT support to the competent authorities in order to ensure the trace ability of phyto-products and ensuring their sanity and protection, enabling fast and accurate reaction and proactive sanitization of the plant health in cases of introduction and breakouts of plant diseases.

Taking into account the principles of modularity and expandability of establishment of the system, both subjects handling the phytosanitary material as well the passports issued for them have to be recorded, as well as the number of information objects recorded in the system and the number of data for each of them can be increased.

In order to achieve this, the system has to be integrated into the current environment of the National food agency, especially when concerning the integration with their infrastructure system (concerning user authentication) and the integration with justice register (legal subjects/persons).

Purposes

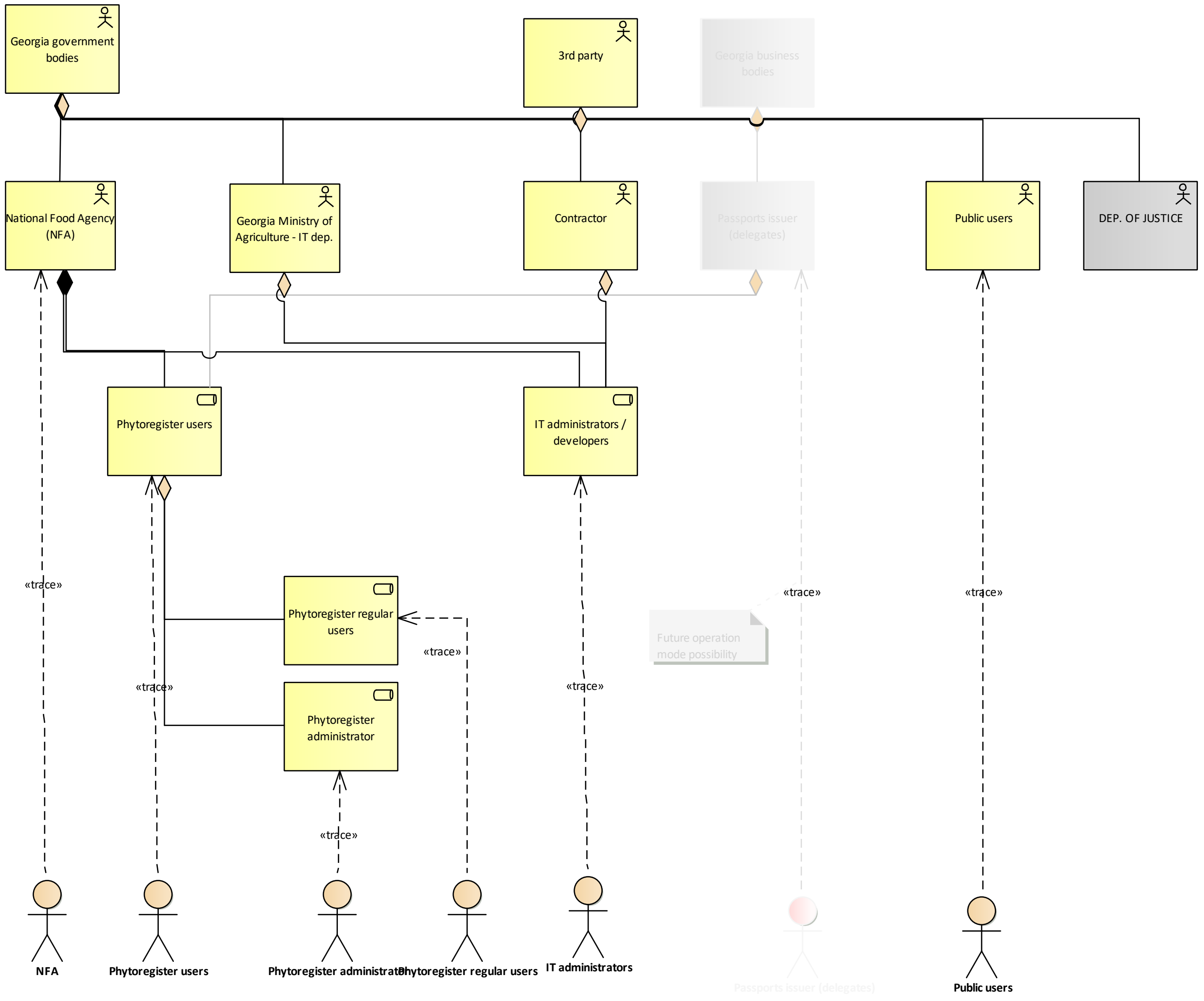
Purposes of the system that have to be addressed within the delivery of the software (including the planned means of fulfilment during the analysis phase for the “Analysis and system design document” etc.):

1. enrolment of persons (subjects) whose business activity contains the handling of phytosanitary material in Georgia;
2. registration of detailed person's activities and production places of them;
3. possibility to keep records and monitor the origins of phytosanitary materials;
4. issuing of the plant passports;

NOTE: In the following diagrams, any elements/concepts grayed out are NOT part of the delivery (or to be supported by the delivered software), however in order to reach mutual understanding and for the sake of completeness and integrity of the diagrams/views, they are part of these diagrams.

2. System to-be - tender requirements - Organization

2.1 Organization

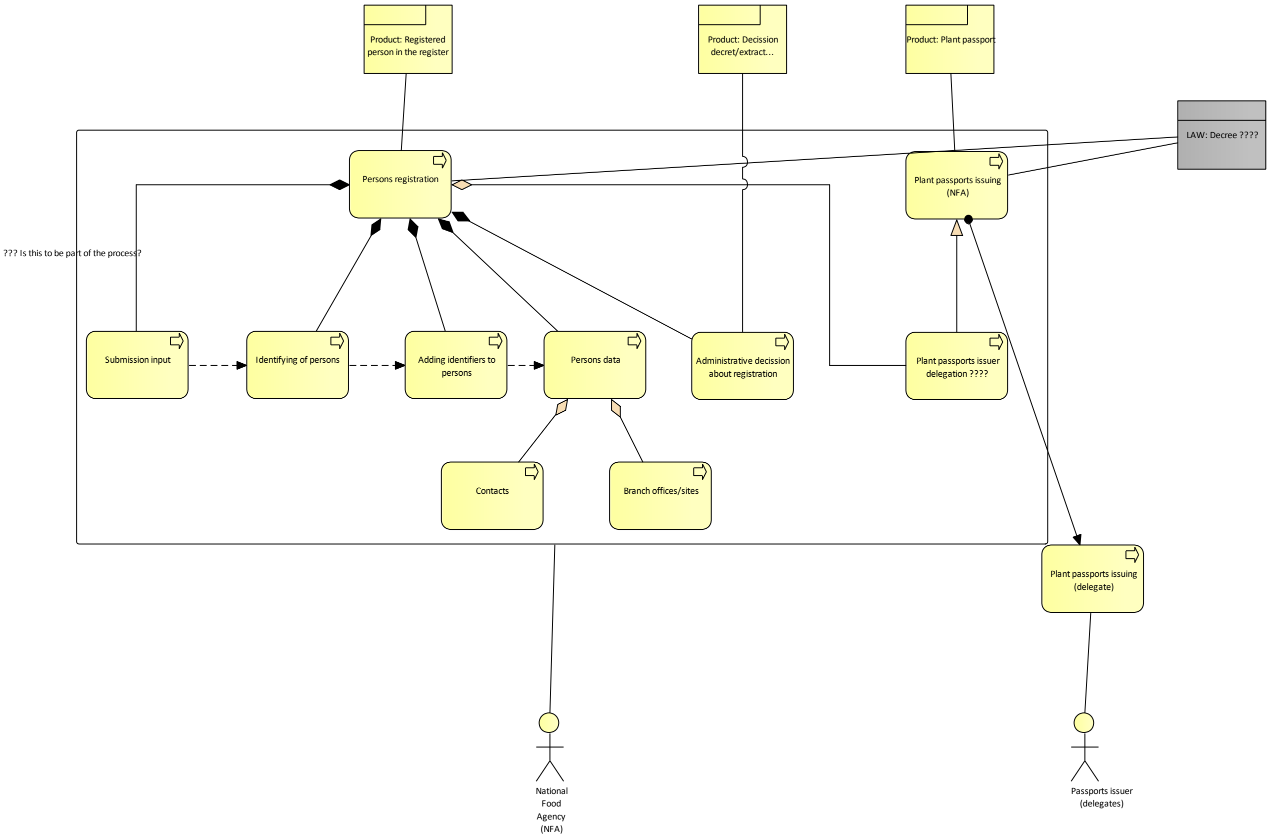

Georgia government bodies
3rd party
Georgia business bodies
National Food Agency (NFA)
Georgia Ministry of Agriculture - IT dep.
Contractor
Passports issuer (delegates)
Public users
DEP. OF JUSTICE
Phytoregister users
IT administrators / developers
Phytoregister regular users
Phytoregister administrator
Future operation mode possibility
«trace»
«trace»
«trace»
«trace»
«trace»
«trace»
«trace»
«trace»
NFA
Phytoregister users
Phytoregister administrator
Phytoregister regular users
IT administrators
Passports issuer (delegates)
Public users

The diagram recapitulates the parties involved in the software usage and specifies their perspective roles (roles registered here are intended for both notations UML and ArchiMate). This might be altered, changed or elaborated in more detail during the analysis within the project.

Public users (::«ArchiMate_BusinessActor»)	This understands rather authorized bodies such as customs office etc.
Phytoregister users (::«ArchiMate_BusinessRole»)	(abstract role / roles grouping)

3. System to-be - tender requirements - Processes

3.1 Preliminary process list


Product: Registered person in the register
Product: Decission decret/extract...
Product: Plant passport
LAW: Decree ????
Persons registration
Plant passports issuing (NFA)
??? Is this to be part of the process?
Submission input
Identifying of persons
Adding identifiers to persons
Persons data
Administrative decission about registration
Plant passports issuer delegation ????
Contacts
Branch offices/sites
Plant passports issuing (delegate)
National Food Agency (NFA)
Passports issuer (delegates)

One may observer in this drawing:

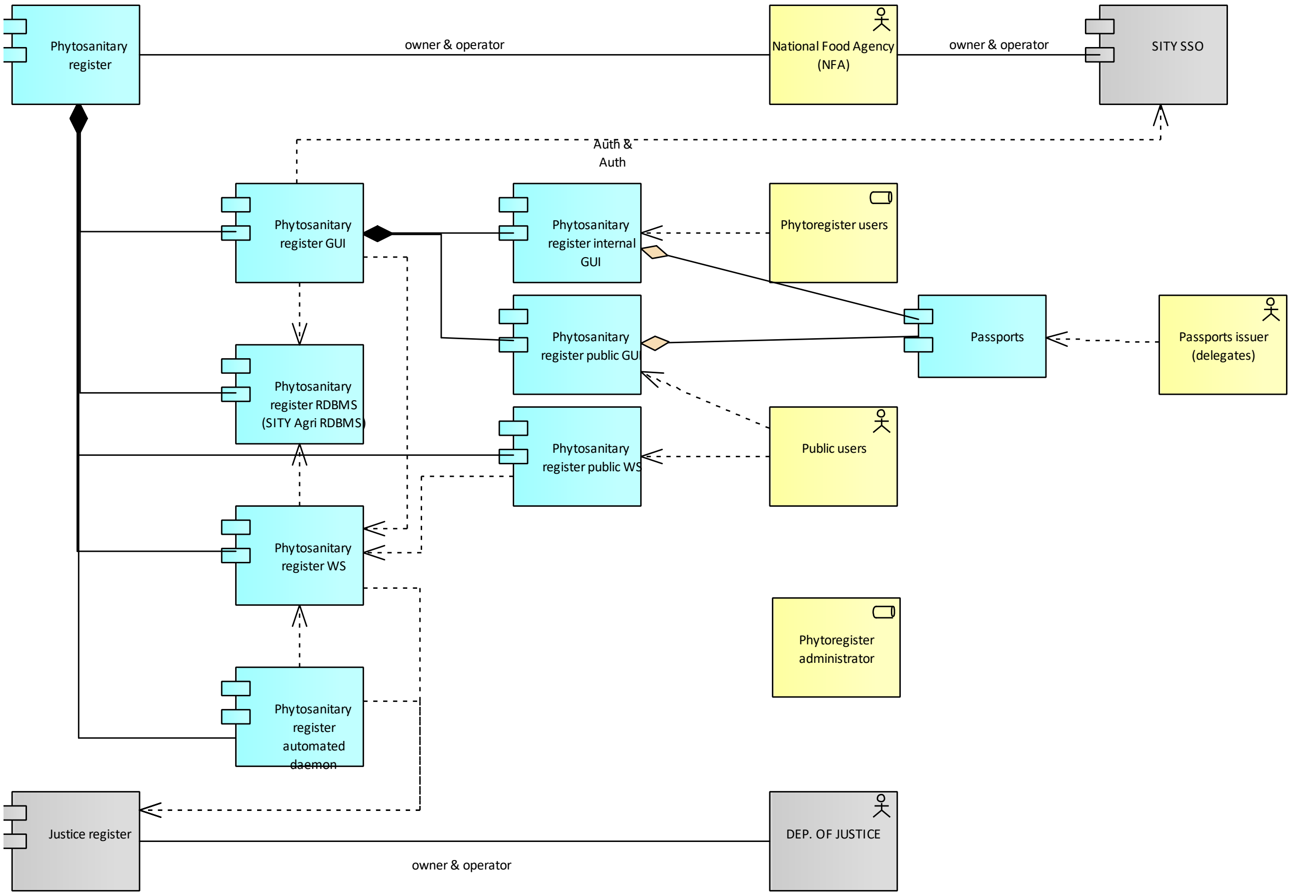
- The structure of the process "persons registration"
- The possible delegation of the passports issuing
- And who (what organization) shall carry out the processes.

This might be elaborated in more detail during the analysis within the project.

Adding identifiers to persons (::«ArchiMate_BusinessProcess»)	NFA is planning to give new number to registered persons, based on crop type and region.
Administrative decission about registration (::«ArchiMate_BusinessProcess»)	Introducing other data as needed
Identifying of persons (::«ArchiMate_BusinessProcess»)	How the persons are to be identified within the register and correlated with the submission for registration (has to be addressed in the analysis and cleared in accordance to the respective Georgian laws as early in the project as possible).
Persons data (::«ArchiMate_BusinessProcess»)	Introducing other data as needed
Persons registration (::«ArchiMate_BusinessProcess»)	The core process being carried out - the registration of persons handling products requiring phytosanitary monitoring.
Plant passports issuing (NFA) (::«ArchiMate_BusinessProcess»)	The process of issuing plant passports for the goods of phytosanitary material.
Submission input (::«ArchiMate_BusinessProcess»)	How the persons are to be identified within the register and correlated with the submission for registration.

4. System to-be - tender requirements - Architecture drafts

4.1 Architecture drafts

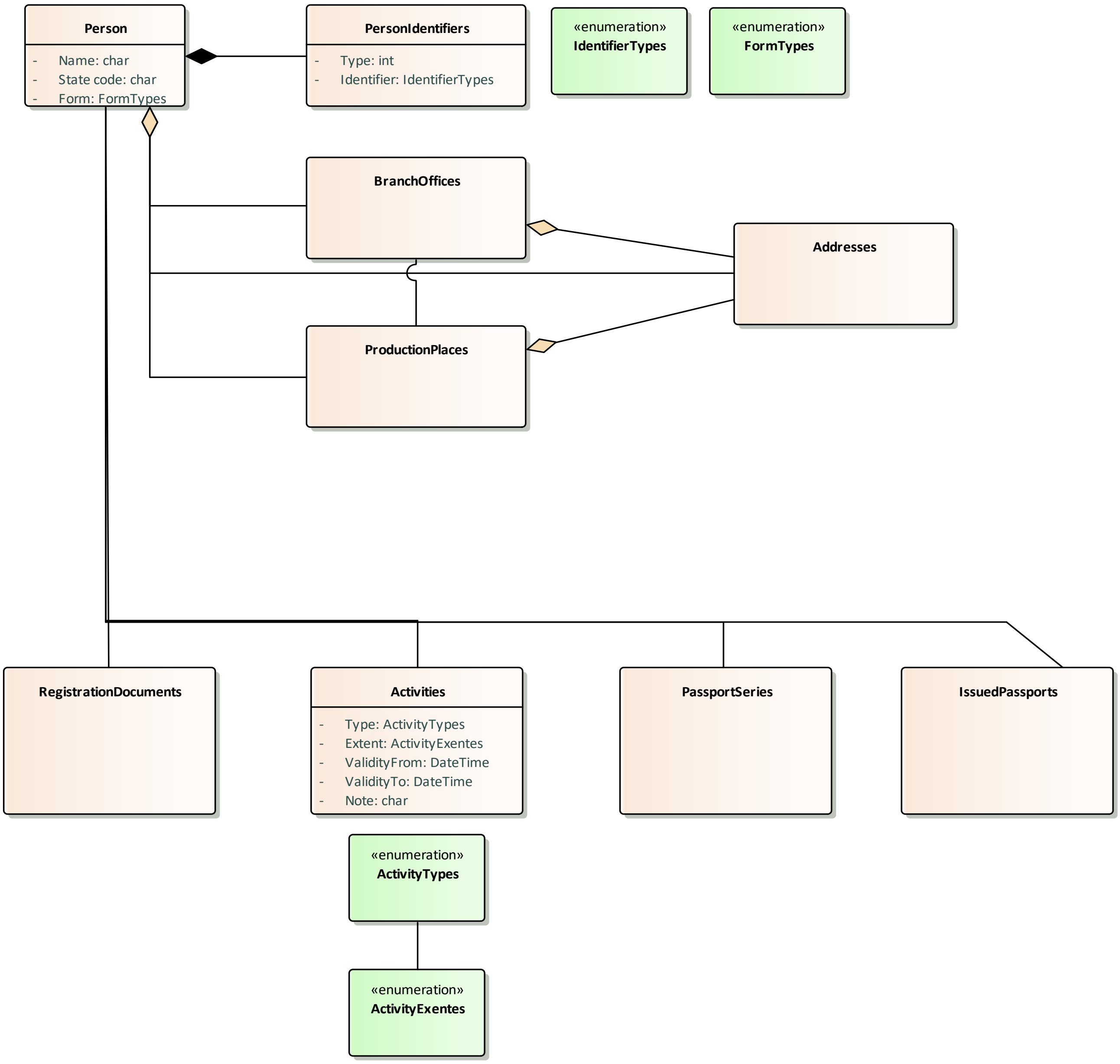

Phytosanitary register
owner & operator
National Food Agency (NFA)
owner & operator
SITY SSO
Auth & Auth
Phytosanitary register GUI
Phytosanitary register internal GUI
Phytoregister users
Passports
Passports issuer (delegates)
Phytosanitary register public GUI
Phytosanitary register RDBMS (SITY Agri RDBMS)
Phytosanitary register public WS
Public users
Phytosanitary register WS
Phytoregister administrator
Phytosanitary register automated daemon
Justice register
DEP. OF JUSTICE
owner & operator

High-level description of modules of which the system should consist and their respective stakeholders roles.
This might be altered, changed or elaborated in more detail during the analysis within the project.

Public users (::«ArchiMate_BusinessActor»)	This understands rather authorized bodies such as customs office etc.
Phytoregister users (::«ArchiMate_BusinessRole»)	(abstract role / roles grouping)
Justice register (::«ArchiMate_ApplicationComponent »)	https://enreg.reestri.gov.ge/main.php?m=new_index&state=search
Phytosanitary register (::«ArchiMate_ApplicationComponent »)	The newly designed and created Information system of Phytosanitary register that is the object of this tender and the resulting delivery.
Phytosanitary register GUI (::«ArchiMate_ApplicationComponent »)	The Graphical user interface used for the users to display, access and manipulate the data records of the register.

5. System to-be - tender requirements - Data drafts (logical classes)

5.1 Data drafts



Main overview of data entities being handled by the system.

FormTypes
Enumeration

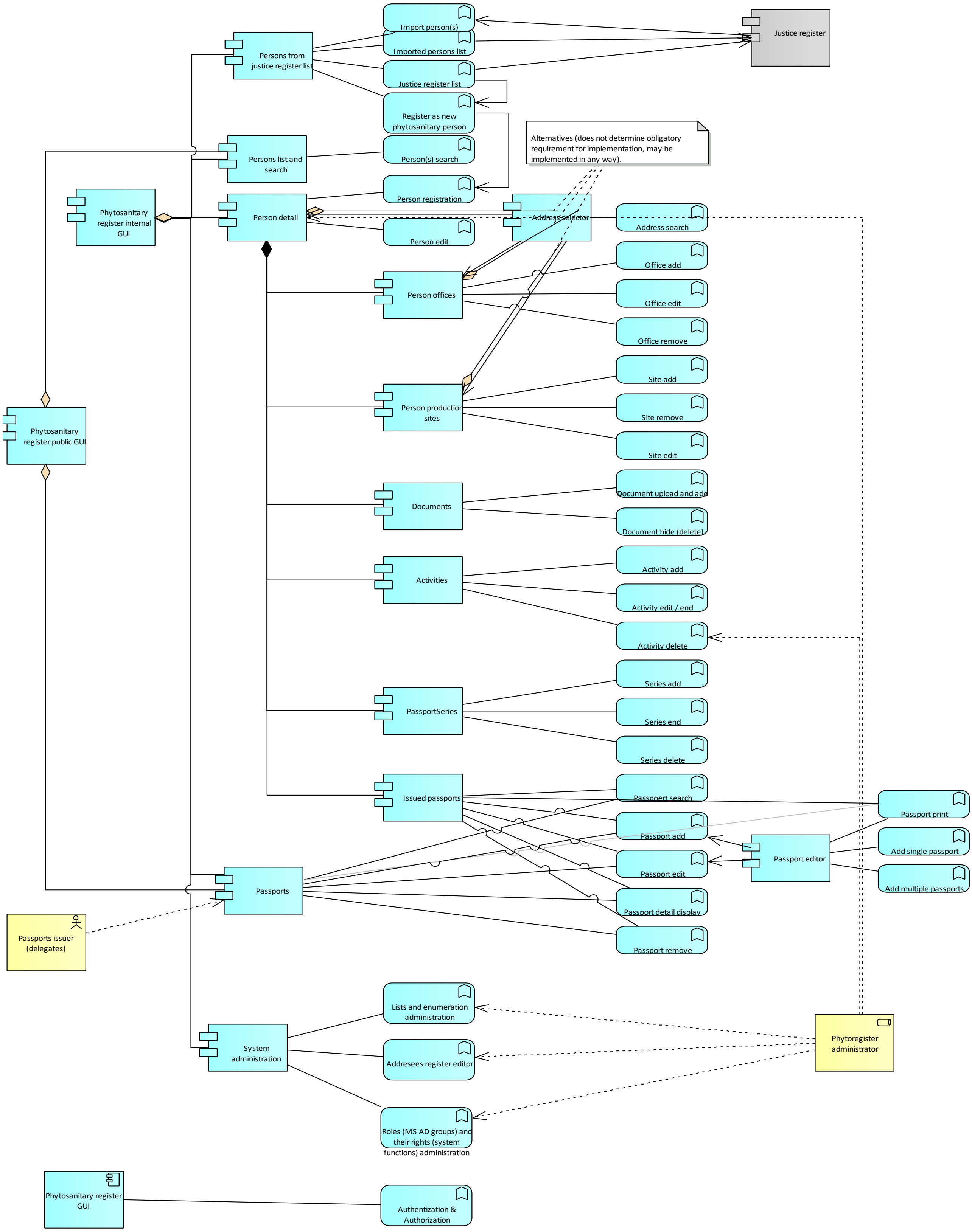
List of identifier types (can be set up dynamically)

IdentifierTypes
Enumeration

List of identifier types (can be set up dynamically)

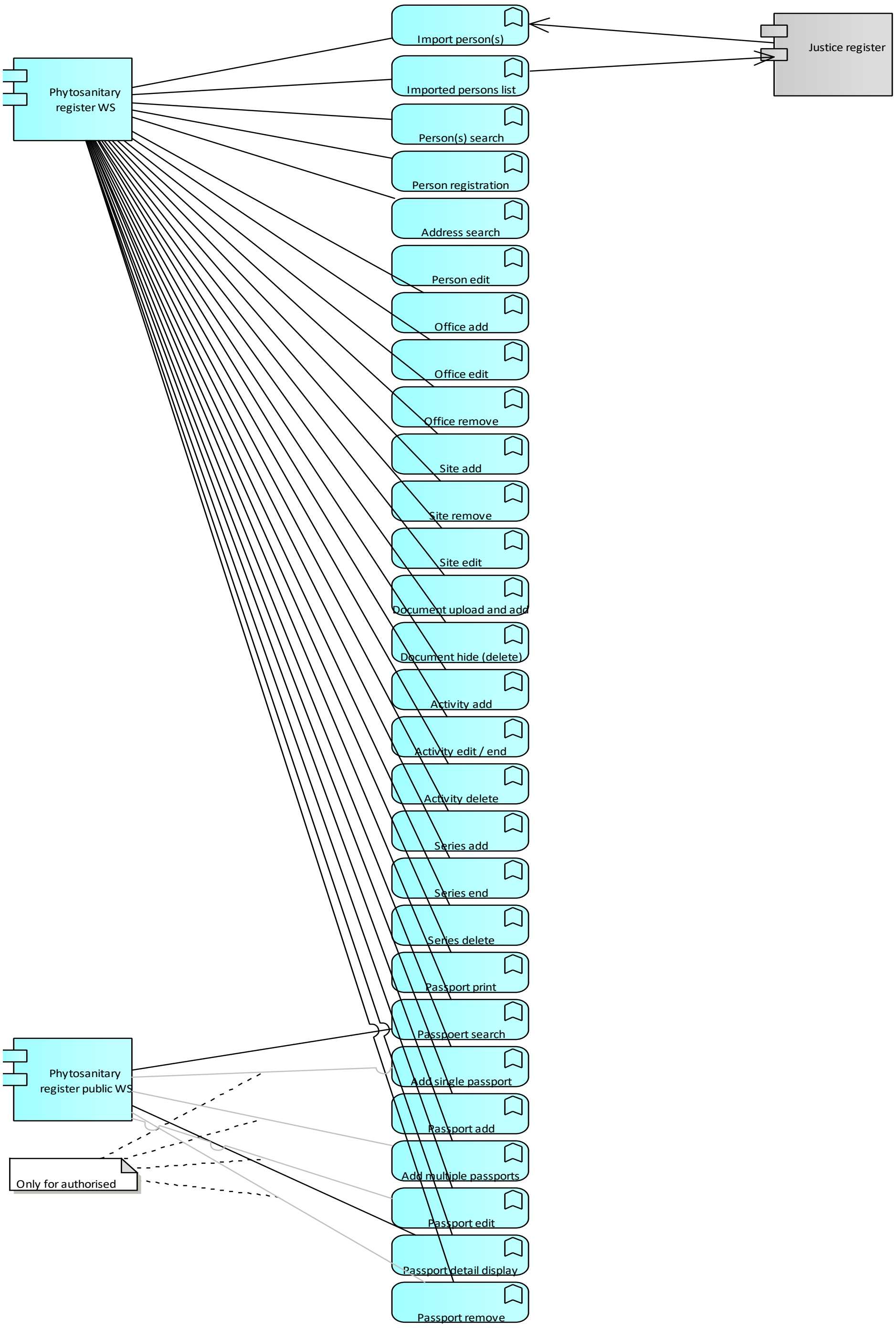
6. System to-be - tender requirements - Functional drafts

6.1 Functional drafts - GUIs



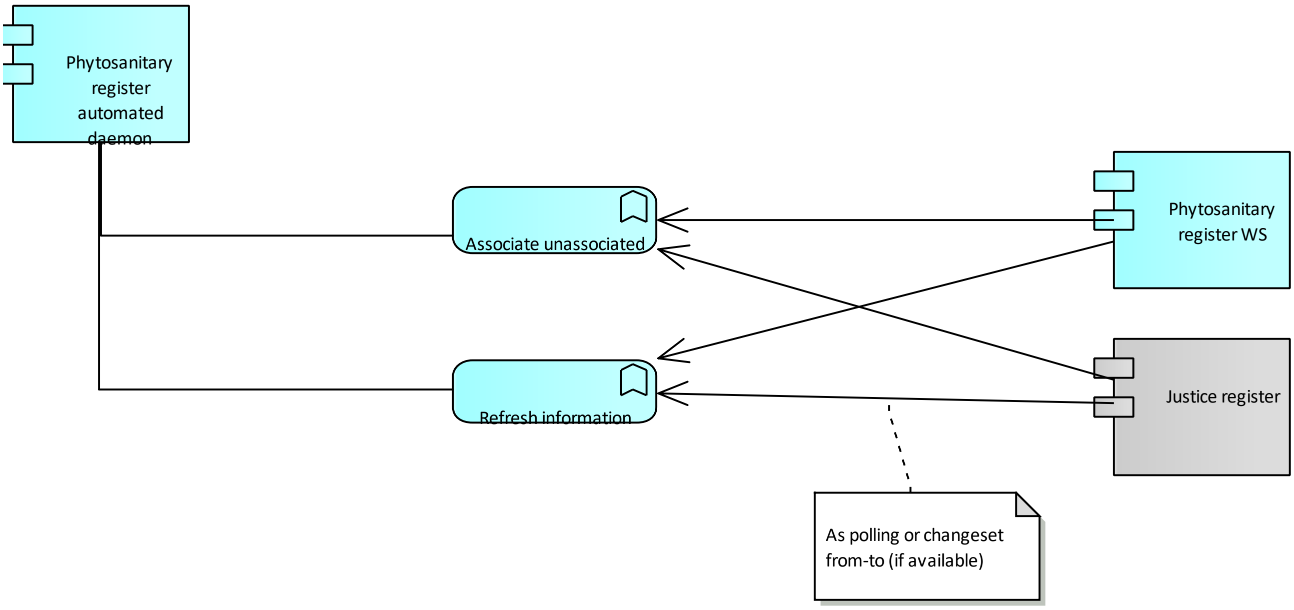
Person registration (::«ArchiMate_ApplicationFunction»)	Shall be able to identify duplicite person based on: • any [ID+IDtype] • Name+state code+Form(+validity) • Seat address (warning only)
Justice register (::«ArchiMate_ApplicationComponent »)	https://enreg.reestri.gov.ge/main.php?m=new_index&state=search
Phytosanitary register GUI (::«ArchiMate_ApplicationComponent »)	The Graphical user interface used for the users to display, access and manipulate the data records of the register.

6.2 Functional drafts - WSs



Person registration (::«ArchiMate_ApplicationFunction»)	Shall be able to identify duplicite person based on: • any [ID+IDtype] • Name+state code+Form(+validity) • Seat address (warning only)
Justice register (::«ArchiMate_ApplicationComponent»)	https://enreg.reestri.gov.ge/main.php?m=new_index&state=search

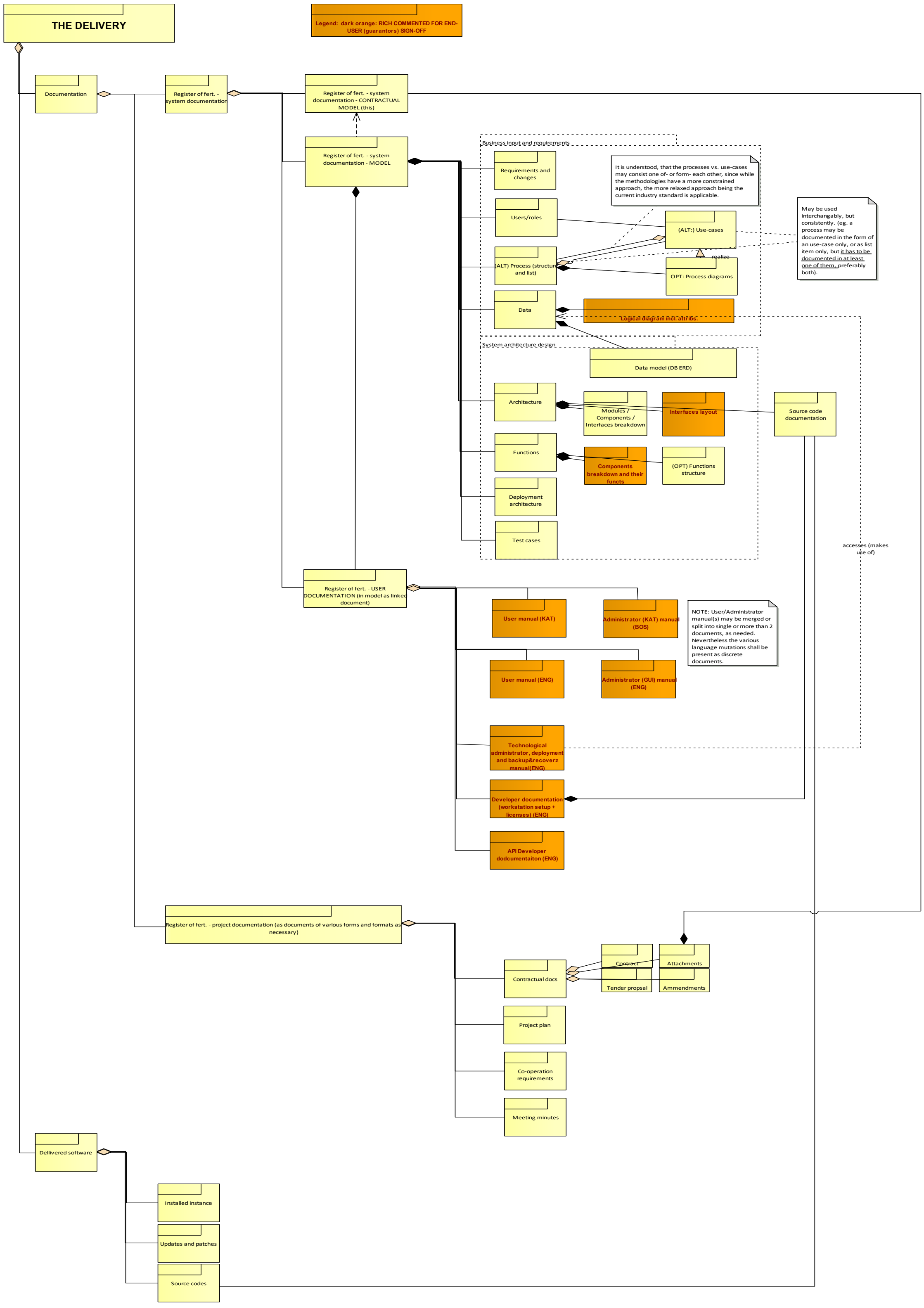
6.3 Functional drafts – automated


Phytosanitary register automated daemon
Associate unassociated
Refresh information
Phytosanitary register WS
Justice register
As polling or changeset from-to (if available)

Justice register (::«ArchiMate_ApplicationComponent»)	https://enreg.reestri.gov.ge/main.php?m=new_index&state=search

7. System to-be - tender requirements - Delivery content (incl. metamodel)

7.1 Documentation structure


THE DELIVERY
Legend: dark orange: RICH COMMENTED FOR END-USER (guarantors) SIGN-OFF
Documentation
Register of fert. - system documentation
Register of fert. - system documentation - CONTRACTUAL MODEL (this)
Register of fert. - system documentation - MODEL
Business input and requirements
Requirements and changes
Users/roles
(ALT) Process (structure and list)
Data
(ALT:) Use-cases
OPT: Process diagrams
realize
Logical diagram incl. attribs.
It is understood, that the processes vs. use-cases may consist one of- or form- each other, since while the methodologies have a more constrained approach, the more relaxed approach being the current industry standard is applicable.
May be used interchangably, but consistently. (eg. a process may be documented in the form of an use-case only, or as list item only, but it has to be documented in at least one of them, preferably both).
System architecture design
Data model (DB ERD)
Architecture
Modules / Components / Interfaces breakdown
Interfaces layout
Source code documentation
Functions
Components breakdown and their functs
(OPT) Functions structure
Deployment architecture
Test cases
accesses (makes use of)
Register of fert. - USER DOCUMENTATION (in model as linked document)
User manual (KAT)
Administrator (KAT) manual (BOS)
NOTE: User/Administrator manual(s) may be merged or split into single or more than 2 documents, as needed. Nevertheless the various language mutations shall be present as discrete documents.
User manual (ENG)
Administrator (GUI) manual (ENG)
Technological administrator, deployment and backup&recoverz manual(ENG)
Developer documentation (workstation setup + licenses) (ENG)
API Developer dodcumentaiton (ENG)
Register of fert. - project documentation (as documents of various forms and formats as necessary)
Contractual docs
Contract
Attachments
Tender propsal
Ammendments
Project plan
Co-operation requirements
Meeting minutes
Delivered software
Installed instance
Updates and patches
Source codes

(ALT) Process (structure and list)	Containing: • Mandatory: Processes (actions/activities ... formal name depending on notation used) being supported by the SW and their respective relations (Composition/Agregation) LOD: • Up to the level of assignable user rights (assignable functions) to the respective roles in the administration. • Note: Should be linkable to "Components breakdown and their functs") Notation: • ArchiMate any ver - Business layer / Application layer; UML any ver: Activity diagram; Structured lists
(ALT:) Use-cases	Containing: • Mandatory: User roles • Mandatory: Use-cases performed by the respective roles • Optional: Use-case structure ("include" etc.) LOD: • Up to the level of assignable user rights (assignable functions) to the respective roles in the administration. • Note: Should be linkable to "Components breakdown and their functs") Notation: • pref.: UML any ver. - Use-case diagram • also possible: ArchiMate any ver. Buisness/Application layer; Structured list.
(OPT) Functions structure	The structure of functions interacting with each other and/or accessing other services. Containing: • Mandatory: Application Components/Services/Functions and their structure and interactions • Opt.: Any other Application layer objects (concepts) LOD: • no specific prefs. Notation: • ArchiMate any ver. Application ev. Technology layer
Administrator (GUI) manual (ENG)	Use-case by use-case description of administrator actions and the ways of performing thereof (HOW-TOs) or the very same based on the modules and functionality tree of the software.
Administrator (KAT) manual (BOS)	Use-case by use-case description of administrator actions and the ways of performing thereof (HOW-TOs) or the very same based on the modules and functionality tree of the software.
API Developer dodcumentaiton (ENG)	(no specific requirements)
Components breakdown and their functs	Links of functions to their respective application modules/parts. Containing: • Mandatory: Application modules (structure) and their respective functions • Opt.: Any other Application / Technology level objects (elements/concepts) LOD: • no pref, should be linkable to either "Use-cases" or "Process (structure and list)" Notation: • pref.: ArchiMate any ver. - Application layer RICH COMMENTED FOR END-USER (guarantors) SIGN-OFF, may be joined into one document / model with Interfaces layout.
Co-operation requirements	Containing required role, its description and schedule as per PC001
Data	IMPORTANT NOTE: With no prejudice to tool and notation used, however any data model featuring relations and having attributes noted within the class/entity objects (such as but not limited to: class diagram, tables and columns, RDBMs generated documentation

	etc.) HAS TO meet following requirement: The relation lines have to start/end (being anchored) on the line of attribute of primary or foreign key being used in that relation.
Data model (DB ERD)	Containing: • Objects/views/procs, their attributes, constraints and relations, all with description of their meaning LOD: • 1:1 to technological (SQL-level view) representation in the RDBMS used. Notation: • no pref. IMPORTANT NOTE: With no prejudice to tool and notation used, however any data model featuring relations and having attributes noted within the class/entity objects (such as but not limited to: class diagram, tables and columns, RDBMs generated documentation etc.) HAS TO meet following requirement: The relation lines have to start/end (being anchored) on the line of attribute of primary or foreign key being used in that relation.
Deployment architecture	Containing: • Architecture components representable as artefacts, e.g. packages deployable on nodes; nodes and infrastructure describing a technologically configurable environment, including network architecture + ports used between components. LOD: • To be usable for administrators/deployment team and infrastructure opponency Notation: • pref.: ArchiMate any ver. - Technology layer
Developer documentation (workstation setup + licenses) (ENG)	• Development prerequisites needed for the code to be run (including 3rd party tools installers where possible or necessary from the point of view of the licensing terms) incl. processes and means of keeping track of changes and patching changes made during the warranty period. • Development environment setup guide (may include virtual machines images, services mock-ups etc., however these do not substitute the step-by-step guide) • License numbers / registration data for components that should require such a number or registration. • May contain also a cloneable virtual image of a developer workstation
Functions	The functionality of the respective modules/components of the system (ArchiMate - app layer (preferred) OR UML Component)
Interfaces layout	Interfaces layout of GUI, WSs, APIS... For webservices: • Methods list • their I/Os attributes • exceptions. For GUI: • rich textual description of the functionalities, • can be also provided in the form of the User / Administrator manual. • Schemas up to the detail of data attributes (depicted as input fields labels and list/table headers) and functionalities, • featuring at least one example representing also the graphical layout (template) for each type of functionality. RICH COMMENTED FOR END-USER (guarantors) SIGN-OFF, may be joined with Components breakdown in one document/model.
Legend: dark orange: RICH COMMENTED FOR END-USER (guarantors) SIGN-OFF	Any of the dark orange elements denotes a deliverable where: • in case of a model rich textual description should be provided, enabling to deliver the information to end-user even when he or she is not familiar with the notation (eg. typically: the diagram description should state the very same information as depicted on the diagram including not only the various objects/elements but the description should convey also the information represented by the various relations between the objects)

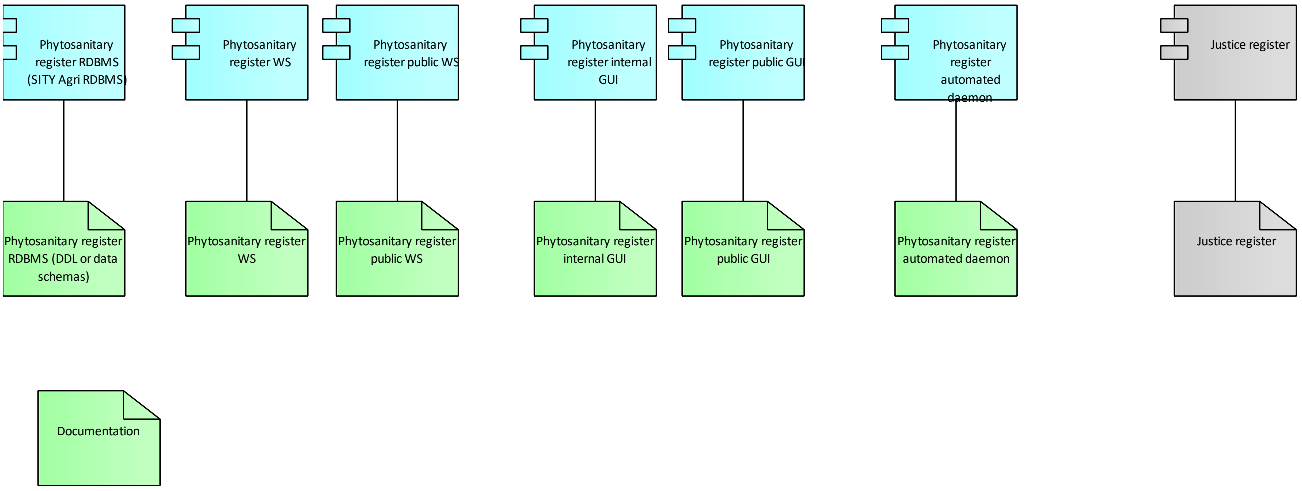
	• in case of textual documents these are to be written for non-professional end-users (in case of manuals) or non-developer/programmer users (in case of document for technological administrators) or (in case of developer/programmer documents) in a friendly and easy-to-understand way.
Logical diagram incl. attribs.	Containing: • A full list of data entities existing throughout the entire system and their relations, including full list of their attributes (logical perspective, not physical) (UML class diagram). • Optionally: data entities may be additionally represented multiple times with specification of variants of each used for persistence, data flow within the I/O of various interfaces etc. (for example: while a person may have the set of 20 attributes, this data entity may be represented not only as "general one" having 20 attributes, but also as a data entity of some say WS or GUI, where only 10 of the attributes are used). LOD: • Data entities on logical level up to their atomic attributes, where the word "atomic" means "representable as some electronically recognized data type" Notation: • pref.: UML any ver. Class diagram RICH COMMENTED FOR END-USER (guarantors) SIGN-OFF • any other - Crowfoot, Coad/Yourdon, generic ERD, tables/lists... IMPORTANT NOTE: With no prejudice to tool and notation used, however any data model featuring relations and having attributes noted within the class/entity objects (such as but not limited to: class diagram, tables and columns, RDBMs generated documentation etc.) HAS TO meet following requirement: The relation lines have to start/end (being anchored) on the line of attribute of primary or foreign key being used in that relation.
Modules / Components / Interfaces breakdown	Architecture of modules/components of the system and their relations Containing: • Mandatory: Components • Mandatory: Composition/aggregation links • Optionally: Services etc., realization/assignment and other links. LOD: • Application parts(modules) up to discrete I/O sets (screens, templates, controls, reports...) See PO006 Notation: • pref.: ArchiMate any ver. - Application layer • alt.: UML any ver - Component diagram
OPT: Process diagrams	Containing: • The processes descriptions/flows, branching, data flows LOD: • no specific requirements Notation: • No specific requirements, try to avoid ArchiMate business layer, rather UML Activity/Sequence/Sync diagrams or BPMN.
Requirements and changes	Containing: • Mandatory: Requirements, issues and the changed/added requirements with "trace" relations in between them. LOD: • All requirements/issues/changes independently on their respective LOD,

	Notation: • SEA Requirements model
Source code documentation	Source codes files / configuration projects and resources structure overview and description Containing: • The source code files + meta data (project files for IDE etc.) including comments • any kind of structured overview of the files groups and their link to respective modules (architecture modules - LOD 1. in PO006)) + their inner logical program design structure (classes, object model)
Technological administrator, deployment and backup&recoverz manual(ENG)	Deployment and technological architecture manual, including 1. full step-by-step manual for installation of the application (including basic schemas of application layout and setup possibilities, if applicable), 2. the documentation of all the configurable variables in any configuration repositories (such as but not limited to: the Microsoft Windows registry, any text/xml/csv/tabbed/json and similar files, the database, configuration repositories of any kind in any proprietary software including the RDBMS being used, templates/reports and the available variables for the report templates configuration; if a 3rd party software is used, then the documentation may be covered by the documentation of the 3rd party software itself, however an addendum has to be made about how the configuration influences the delivered software). 3. Network architecture and ports used between the components (may use the "Deployment architecture" diagram) 4. Backup & recovery plan containing: • Analysis of data assets (may be part of data models) • Architecture of backup solution(s) being employed • Data assets <--> backup solution(s) (which asset is backed up by what solution and with what frequency, type of backup etc.) • Steps to check the backups being done correctly (per typ of backup solution) or (should this require manual operation) the manual for performing these steps. • Steps to recover each data asset (per type of backup solution)
Test cases	• All tests/test cases have to reference either a requirement specified already in this Contract/Tender documents or a requirement provided during the analysis. One requirement may result in multiple tests, and one test may cover multiple requirements, if necessary (eg. the level-of-detail should provide at least such a detail, that each test-case belongs to each function/requirement or small groups there of). (Note: this does not apply, if necessary, for the “global” (non-functional) requirements, which may be tested either 1. within another test case (but this has to be stated as a requirement being tested by the test) 2. separately within an acceptance test or 3. within a test without a scenario: a mere “statement” test, such as “according to chapter XYZ in the documentation the application has 3 tires” or “(…) the system has a software application with functionalities covering each of the 3 modules”) • Each test has to provide a step-by-step manual for performing the relevant test, including all input values and expected outputs. • If a test may fork into multiple variants based on the data input, all possible inputs (or at least their groups – classes of values) have to be tested. • Each test case (especially in the printed version being used as annex for the acceptance M2) has to provide free space, where the result and final resolution (passed/passed with objections/failed) can be noted. • The test cases are to be described in English language (except the labels/buttons/software I/O texts and data I/O). • The first testing will be done by the contractor, but the test scenarios will be accessible and the second times run by the PHPA or departments or UKZUZ as selected during the acceptance procedure.

	The above mentioned does not intend to influence nor it may not replace any internal testing procedures of the contractor.
User manual (ENG)	Use-case by use-case description of actions and the ways of performing thereof (HOW-TOs) or the very same based on the modules and functionality tree of the software.
User manual (KAT)	Use-case by use-case description of actions and the ways of performing thereof (HOW-TOs) or the very same based on the modules and functionality tree of the software.
Users/roles	Structure of the organization affected and the resulting software roles. Containing: • Mandatory: Independently on independently on their LOD, entire list of roles (and their permitted functionalities - see Use-cases/Processes) must be present LOD: • Up to roles (and their assignable functions - see Use-cases/Processes) that shall be set up in the software. Notation: • pref.: ArchiMate any ver. Business layer • also possible: UML - Use-case any ver.; Structured list.

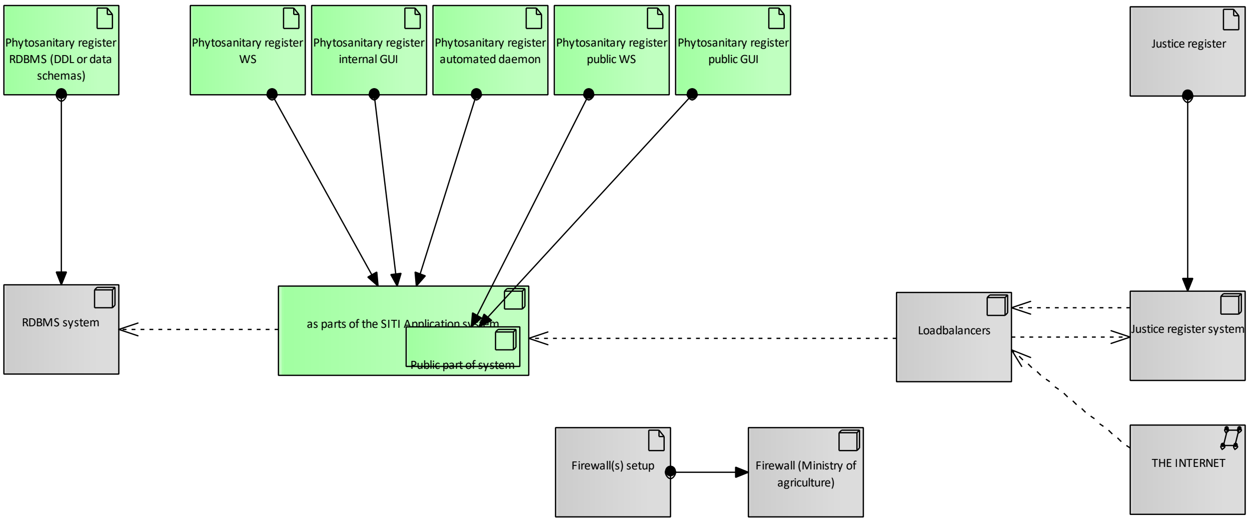
8. System to-be - tender requirements - Deployment drafts

8.1 Component to Artefact transition

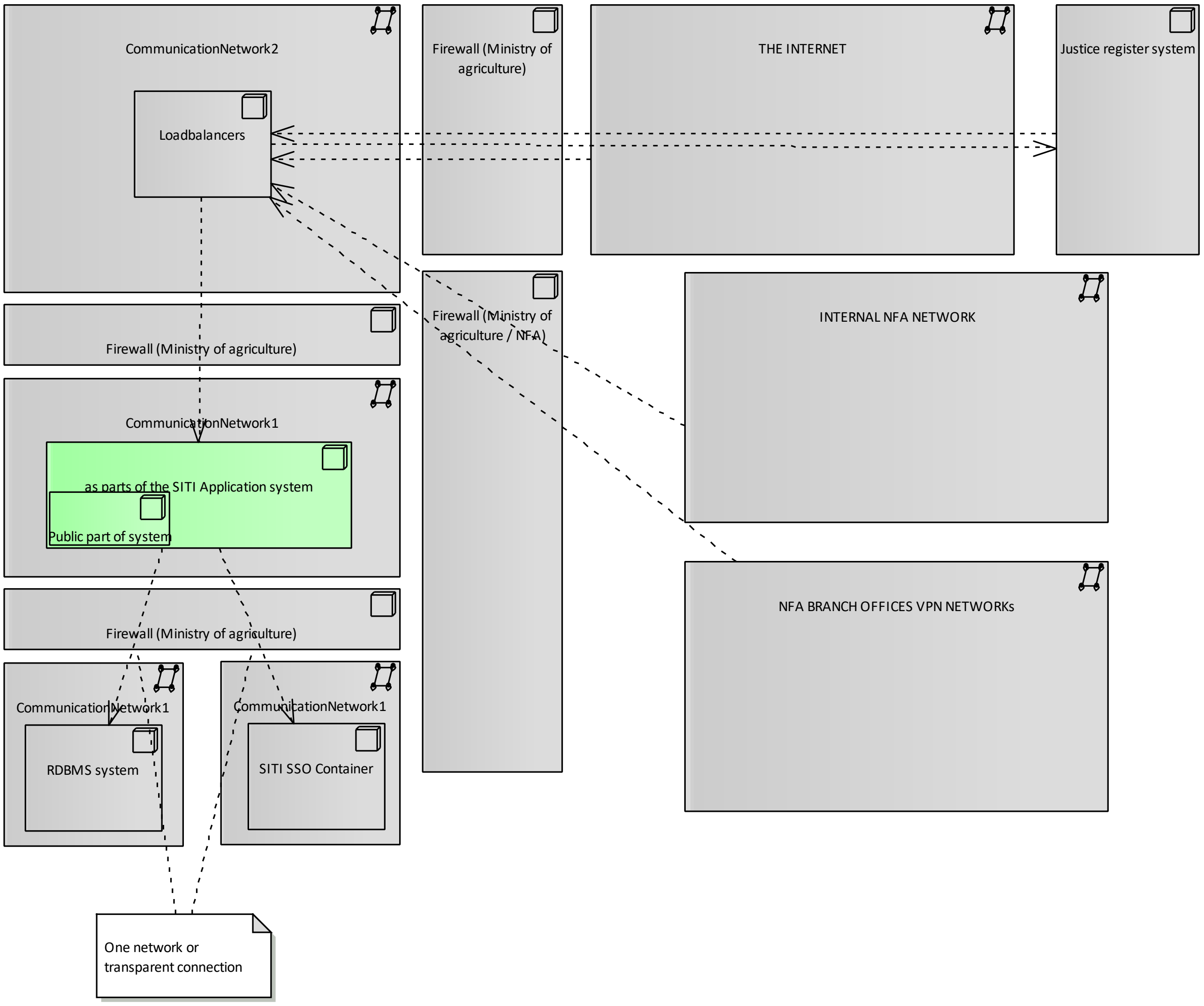

Phytosanitary register RDBMS (SITY Agri RDBMS)
Phytosanitary register WS
Phytosanitary register public WS
Phytosanitary register internal GUI
Phytosanitary register public GUI
Phytosanitary register automated daemon
Justice register
Phytosanitary register RDBMS (DDL or data schemas)
Phytosanitary register WS
Phytosanitary register public WS
Phytosanitary register internal GUI
Phytosanitary register public GUI
Phytosanitary register automated daemon
Justice register
Documentation

Justice register (::«ArchiMate_ApplicationComponent »)	https://enreg.reestri.gov.ge/main.php?m=new_index&state=search

8.2 Deployment draft


Phytosanitary register RDBMS (DDL or data schemas)
Phytosanitary register WS
Phytosanitary register internal GUI
Phytosanitary register automated daemon
Phytosanitary register public WS
Phytosanitary register public GUI
Justice register
RDBMS system
as parts of the SITI Application system
Public part of system
Loadbalancers
Justice register system
Firewall(s) setup
Firewall (Ministry of agriculture)
THE INTERNET

8.3 Deployment draft - devices/network required architecture


CommunicationNetwork2
Loadbalancers
Firewall (Ministry of agriculture)
THE INTERNET
Justice register system
Firewall (Ministry of agriculture)
Firewall (Ministry of agriculture / NFA)
INTERNAL NFA NETWORK
CommunicationNetwork1
as parts of the SITI Application system
Public part of system
NFA BRANCH OFFICES VPN NETWORKs
Firewall (Ministry of agriculture)
CommunicationNetwork1
RDBMS system
CommunicationNetwork1
SITI SSO Container
One network or transparent connection

(gray = not part of the delivery, green = parts of the delivery)

9. System to-be - tender requirements - Requirements catalogue

This part of the documentation is the main catalogue holding the list of requirements as specified for the various part of the system. These requirements are of three main categories:

- Functional requirements - describing the required functional aspects of the system in regard to the processes or agendas the system shall be supporting.
- Non-functional requirements - all other "general" constraints of the system that the system should follow and respect in all it parts.
- Other requirements - dealing rather with the project-based issues, such as requirements for project management, documentation outputs and structure, testing requirements etc. If necessary, the respective parts describing each of the requirements in detail may be found in separate parts of the documentation, or (if not to many), directly within the requirements catalogue.

Note: Functional requirements for shared parts of the systems, that, while being functional ones, however should be valid throughout the entire system are usually parts either of "Architectural requirements" (if concerning management, installation, setup or configuration properties of the system) or "Interfaces requirements" (if concerning typically GUI parts or interfaces concepts valid in all of them such as headers/envelopes etc.).

Please note following important contractual statement: THIS CATALOGUE REPRESENTS THE ONLY AND ENTIRE CONTRACTUALLY BINDING SET OF REQUIREMENTS. THE CONTENT OF PREVIOUS SET OF DIAGRAMS IS CONSIDERED to be rather informative for the contractor in order to gain a faster, better and more comprehensive overview of the ordered system-to-be, yet the diagrams are considered as CONTRACTUALLY BINDING ONLY IN THOSE CASES, WHEN THEY ENUMERATE OR STRUCTURE ANY INFORMATION DIRECTLY OR INDIRECTLY REFERRED FROM WITHIN THE REQUIREMENTS CATALOGUE.

9.1 Requirements catalogue - Functional requirements (FR)

This part of the documentation describes the required functional aspects of the system in regard to the processes or agendas the system shall be supporting.
While one may observe the processes and their order, bindings and logical dependency in the appropriate documentation parts, here the requirements are merely a list of these processes or sub-processes, to ensure their implementation according the abovementioned specification.
Note: This chapter is intended as a preliminary breakdown of the requirements specified further, so this chapter does not intend to replace or support the Functional requirements delivered by the winning contractor within the scope of the project as part of the “Analysis and system design document”.

9.1.1 Functional requirements (FR) - Person enrolment

There are to be several ways how a person (legal subject handling phytosanitary material) can be introduced into the system, where manual input is only one of them.

FR0001	FR0001 Person import - justice register semi-annual export
The system has to allow import data of economic agents from the justice register (who applied for phytosanitary operations there, or, if necessary due to legislation constraints, identified by other type of filter) in any form the semi-annual export is delivered in. In case the data needs any kind of filtering not previously done during the export, based on the data contained in the report, this filtering has to be done.	

<table>
<tr><th>FR0002</th><th>FR0002 Preview of list from justice register semi-annual export</th></tr>
<tr><td colspan="2">Before the persons on the list are stored in the phytosanitary register the list of imports from the list data should be previewed.</td></tr>
</table>

<table>
<tr><th>FR0003</th><th>FR0003 Person load - direct query to justice register</th></tr>
<tr><td colspan="2">The system has to allow import data of economic agents (who applied for phytosanitary operations there, or, if necessary due to legislation constraints, identified by other type of filter) when loading by querying the API of the justice register of economic agents.</td></tr>
</table>

<table>
<tr><th>FR0004</th><th>FR0004 Person list preview - direct query to justice register</th></tr>
<tr><td colspan="2">The phytosanitary register has to enable the search in the justice register (based on such parameters set that corresponds to the set of parameters the API of justice register is exposing) and additionally by date of subject primary registration of phytosanitary business activity, if such parameter is present as input or in the output data. The user has to have the possibility to include/exclude subjects to be enrolled.</td></tr>
</table>

<table>
<tr><th>FR0005</th><th>FR0005 Person manual input</th></tr>
<tr><td colspan="2">When the person cannot be obtained by any other automatic enrolment procedure, then the user (with appropriate role) has to have the possibility to enrol the person by manual input.</td></tr>
</table>

<table>
<tr><th>FR0006</th><th>FR0006 Person enrolment checks</th></tr>
<tr><td colspan="2">When a person is enrolled by any mean, the system has to provide checks concerning its data provided checking if:
• the subject has not already been enrolled in the system under different data attributes (if the subject is being enrolled manually, than this check has to be done latest before its storage in the system)
• the subject does not have different attributes in the central register</td></tr>
</table>

9.1.2 Functional requirements (FR) - Person list and search

After a person (legal subject) is introduced into the system, there are several data sub-entities bound to the person that have to be editable.

<table>
<tr><th>FR0007</th><th>FR0007 Person search</th></tr>
<tr><td colspan="2">The persons stored in the phytosanitary register have to be searchable and details displayable by selecting the required person from the resulting list.</td></tr>
</table>

<table>
<tr><th>FR0008</th><th>FR0008 Person edit</th></tr>
<tr><td colspan="2">The person data has to be editable by user with appropriate role. If the data is being refreshed automatically from justice register than the attribute has to be securable against further automatic refresh, optionally specifying the expiration timeout of the blockage for each attribute.
When being saved, the person enrolment checks have to apply as well.
Specific sub-screens/elements should be present for editing:
• Person metadata (name, address...)
• IDs (business ID and other)
• Contacts (people names and contacts responsible for various roles)
• Offices (addresses of offices of the legal person)
• Production sites (addresses of production sites)
• Documents (files added to the specific person)
• Activities (list of phytosanitary activities the person is doing)
• List of plant passport series assigned to the person
• List of plant passports / passport series issued by/to the person

Person may be deleted/suspended.</td></tr>
</table>

9.1.3 Functional requirements (FR) - Plant passports list and search

Passports are a specific data entity, since while they are bound to a specific person they have been assigned to, they should be still accessible as a stand-alone entity too, since:

- They might be searchable by authorized public subjects
- They might be (in later stages of the project) accessible and editable by the assignees themselves based on the series of the passports only (if the infrastructure allows such type of access also in the terms of enrolled authentication subjects)

FR0009	FR0009 Passport list
Thy system should provide a list for search and select for editing/creating of plant passports issued for batches of phytosanitary material (to a specific person). (Note that this list might be the entry point for public users as well).	

FR0010	FR0010 Passport detail display
Selected plant passport from list may be displayed. (Note that this list might be the entry point for public users as well).	

FR0011	FR0011 Passport edit
The users of the system (the NFA) shall be able to introduce the plant passports issued for batches of phytosanitary material into the system (as series). The users of the system (the NFA) shall be able to introduce the plant passport series reserved for given registered subject into the system (as series of plant passports). The plant passport (series) may be edited, deleted or discarded.	

FR0012	FR0012 Passport print
The formal representation of the plant passport may be printed in specified form.	

9.1.4 Functional requirements (FR) - Global functions

Certain functions of the register are to be expected to have influence on the entire system, while still not being general non-functional requirements. These are listed in this section.

FR0013	FR0013 Addresses register
When any attribute is a geographical address, this should be input using a selector from an internal addresses register. The editing (adding new address records or changing it attributes, as well deleting) of the addresses register has to made possible.	

FR0014	FR0014 Administration
The system shall have a GUI part (or use standard platform tools of the underlying framework) accessible for administrators of the system only, used to maintain: • the integrity and cleanness of the records • editing of enumerations and other registry lists • editing of the register of addresses • editing of AAA settings (roles (=SSO groups) assignment to their rights (=functions within the system); roles creation and user account assignments into roles if the access to the respective users permit)	

FR0015	FR0015 Access to the system
The system should provide all its functionality by means of automated accessibility (API). Some of them may be accessible publicly (for authorized subjects only).	

FR0016	FR0016 Automatic subject data update
When a subject has been enrolled in the system than the central register should be regularly queried to check, if those user data, that have been introduced into the phytosanitary register, did not change. An attempt should be made for	

manually introduced persons to associate them with their records in the justice register (e.g. recording any kind of correlation ID from the justice register into the data of the phytosanitary register).

FR0017	**FR0017 Data entities in functions**
The terms in the various functional requirements and function descriptions refer to the data entities as specified in the "Data drafts (logical classes)" diagrams. Nevertheless it is up to the contractor to re-specify and refine this specification as necessary.	

9.2 Requirements catalogue - Non-functional requirements

This part of the documentation specifies all other "general" constraints of the system, that the system should follow and respect in all it parts.

9.2.1 Non-functional requirements - Architecture (AR)

These requirements specify those needs that are somehow limiting or specifically asking for a concrete design of the modules and their interconnections of the system at various levels of its architecture (SW, HW and even orgware if necessary).
Detailed concepts of the architecture may be specified in separate diagrams in the appropriate parts of the documentation if necessary.

AR001	**AR001 Implementation platform**
Due to the administrative, sustainability and organization-technological requirements of the final beneficiary, it is required, that the system is implemented **within the SitiAGRI platform** (a system by ABACO Group) by the means of the system: • either within the definition of datasets, client webservices, commands, forms and their association to model etc. (by means such as for example SITI Service Manager / DOSSIERs) or • on the top of the basic platform (and with usage of the shared resources/functions of SitiAGRI) formed by JEE Web containers (on Tomcat (preferred)/Jboss/Oracle AS/WebSphere/BEALogic) above Oracle 10g/11g (+Spatial extension) DB NOTE: In case it shows necessary, the contractor has to take part on platform training courses as needed, and has to cover the costs within the fixed price for the delivery, if not agreed otherwise.	

AR002	**AR002 Provided interface**
All the WS functions specified in this document and/or specified later by the detailed analysis done by the contractor have to be made publicly available, if presented as part of the public WS and when possible by the framework of the underlying system.	

AR003	**AR003 Operation environment architecture**
The system should be set up according to given requirements and conditions as specified during analysis, respecting the deployment diagrams in this document (where the analysis requests should prevail), and, if possible, that in at least 2 environments: • testing environment (does not meet response-time requirements as specified, with exception of those moments, when such acceptance tests are to be performed, that are measuring the performance) • production environment (that has to meet response time as specified in appropriate requirement) During the development of the system, the environments shall be regarded as follows: From beginning of the project up to M2 • the testing environment: won't exist • the production environment: will be called "M2 testing environment" and is to be used by the contracting authorities for testing purposes – the acceptance tests will be based on the performance and behaviour of the application being tested on this environment. Up to M2 (or any other moment, when production data shall be filled or the connectivity to production registers established) the contractor shall have free access to the "M2 testing	

environment", after the M2 (or any later moment as described above) the testing environment shall be converted to "production environment" and the access of the contractor may be limited

After the M2 (or any later moment as described above):

- the (newly created) testing environment is to be used for acceptance tests as described above (incl. performance tests, for this, the testing environment virtual infrastructure configuration may be temporally matched to the configuration of the production environment, where the licenses and the required resources configuration have to cover this scenario), while a second instance of the application (or a third, fourth…) for development purposes is possible, if allowed by the application system being used by the contractor and/or as far covered by the licenses, for example those that have been calculated for the coverage of the performance tests (this means also separate virtual servers, if necessary, however sharing the same resources as the rest of the testing environment)
- production environment: shall be already used as described in the project schedule, may be without the possible direct access by the contractor.

Note: The term "environment" may refer either to a virtual infrastructure (Infrastructure as a Service, provided within the infrastructure of Georgia) or any other instance (tenant) of the platform services of the platforms forming the Georgian platform.

AR004	**AR004 Network protocols**

- Every part (of the client application) that is connecting to the application server is to employ secure protocols running on the top of the standard TCP/IP protocols (e.g. is user-friendly securable by certificates and/or means of asymmetric cryptography, routable, VPN tunnel-able etc.). Should any other network/transport/connection/relational/presentation protocol be employed, it has to meet the requirements specified by the features of the TCP/IP stack or these features have to be ensured programmatically within higher levels of protocols including the application one (if possible by the specification of the app. protocol, if not, such protocol cannot be used as substitute for the requirements above))

AR005	**AR005 Connection reliability**

Client - Server connection must not rely on stable network connections (especially in case of dial-up VPNs).

AR006	**AR006 Software licenses**

- Should the solution proposed by the candidate require any licenses, these licenses have to be calculated within the delivery price and provided as a part of the project.
- The licenses calculated and provided within the delivery have to have NO limitation when regarding the migration on different hardware, since this process is under automatic control of the underlying virtualization platform.
- These licenses provided have to be life-longs. Should they have a validity period specified (such as maintenance/assurance or extended support), then it is understood, that the calculation for a license provided within the delivery has to cover at least the entire time frame of the project, including the warranty period and + 1 year afterwards. If there is such a license, then this license has to be enlisted separately in the proposal and in the contract including its complete (commercial) specification.
- If any of the licenses is not a life-long licence, then the candidate has to provide also the list and costs for further maintenance/renewal of the license for 2 years period (based on the current prices, as if the license renewal/maintenance would be purchased (directly from producer or for a price not lower or higher as recommended by the producer) to the date within one month before the proposal without any specific promotional and other similar benefits). This applies also for the licenses for developers (libraries, third party tools, VDEs etc.) needed.
- The licenses have to be able to cover the requirements of both environments in accordance with requirement "Operation environment architecture".

This does not apply for those **SITIAgri** licenses that are already held by the beneficiary and/or Ministry of Agriculture.

AR007	**AR007 Software licensing**

The entire solution being part of the delivery is foreseeing the handover of the license for any kind of use to the final user of the software (the government of Georgia and its agencies). Thus:

- any tool necessary for the development and/or build or deployment of the solution has to meet requirements as specified in AR006
- AND the output of the work of the contractor that might form a licensable piece of work must be licensed to the final user.

AR008	AR008 Backup solution
The backup solution shall use the current means of backup of the SITIAgri platform as they exist.	

9.2.2 Non-functional requirements - Integration (IR)

These requirements specify the list of other parts/systems that the application(s) should integrate with. If this integration is not only process-based, but also software based, then the appropriate architectural concepts and/or interfaces specifications can be found in the appropriate part of the documentation.

The Phytosanitary register information space is required to rely on interaction with automated information systems, which form the following information resources:

IR001	IR001 Register of economic agents
Register of economic agents of the Justice house (Ministry of justice) for: • Searching of subjects of economic agents in order to download the metadata of the subject (when searching during introducing of the subject into the system manually through GUI as well through WS). • Searching of subjects of economic agents in order to evaluate if an economic agent solely present in the Phytosanitary register has already been introduced into the register of ministry of justice. • Download of the metadata during refresh/update of the subject in Phytosanitary register based on a subject in register of economic agents of ministry of justice.	

IR002	IR002 SITI SSO
SITI SSO services (using JDBC or internal application mechanisms of the system)): • For authentication of users accessing the Phytosanitary register • For download of respective groups the user is part of	

IR003	IR003 Means of access
The external systems will be accessible using methods according to documentation given to the contractor after signing of the contract and relevant documents and use them in scenarios as defined above and/or during detailed analysis. The scenarios may include not only basic / automated / batch identification and data completion on the subjects/entities, but also automatic updates, resulting integrity cross-checks and warnings etc. based on the use-cases of the respective registers and their possibilities/requirements.	

IR004	IR004 Integration fall-back
If any of the (integrated) registers should not perform according to requirements needed by the (implemented) phytosanitary register, following measures are to be taken by the contractor and other parties involved (one after another or in parallel, as the situation requires): • Specification of the under-performing system and negotiation for diagnosing bottlenecks and other issues in the implementation or in the underlying infrastructure that might cause the problems. For this negotiation the presence (preferably on-site) of the contractor is required. • Rectification of the state using optimization on the side of the phytosanitary register (such as for example but not limited to asynchronous calling with the possibility of listing the tasks for (user) reaction on the returned asynchronous calls, limitation of the minimal set or range of input parameters etc.) • Rectification of the state using batch processing (again with user-accessible lists of errors / ambivalences to be decided/fixed) and/or the possibility to input these values as “text-only” (without the synchronous verification on the respective register) • Cancelling of the integration of register with the respective external register – as a written amendment to the contract only – including the delivery of any changes needed to ensure the operation of the system according to all other specification within the scope and original budget of the project up to a degree the contractor can be fairly asked.	

9.2.3 Non-functional requirements - Interfaces (GUI/WS..) and user requirements (UR)

These are general requirements for the structure and layout of the interfaces the software shall be employing to communicate with its outer world, including the users.

UR001	**UR001 Web application**

- For any parts/modules used by more than 3 users or more than 2 users on geographically distant location or more than 2 users in different organization or environments it is required, that the client application is a web application (providing a web interface to the users)

(Note: The client-side application MAY contain any security/validation and other relevant business-functionalities especially for interactivity reasons, provided that the very same are being employed on the server side.)

UR002	**UR002 Supported browsers**

Supported browsers shall be:

- Chrome

and

- IE/Edge or their successors

(all in current version and one previous major version, as long as supported by the manufacturer to the date of M2/M3).

UR003	**UR003 Multiple instance**

The software should allow for each user to open and run more the one instance, without the instances interfering with each other, as far as the underlying framework platform allows.

UR004	**UR004 Multilingualism**

The system, including the prototype/pilot/testing semi-products, delivered at any stage of the project, has to have a bi-lingual user interface (implemented in a manner allowing easy localization of UI texts into next language (XML, DLL...)).
The minimum required languages for the delivery shall be featuring an "English" and "Georgian" ("Kartuli ena" / KAT) version of all descriptions, labels, short texts and context help tool-tips (mock-up prototype delivered by M1 may be in English only), if not agrreed otherwise within contract amendment. This also applies for selection lists.

UR005	**UR005 Autocompleting**

Any value being edited that is based on an enumeration list has to provide an auto-completion feature when being edited in a graphical user interface as far as the underlying platform allows.

UR006	**UR006 Display size and resolution - workstation screens optimization**

The GUI has to be fully working (e.g. fully visible without horizontal scrolling (unless necessary as given by functional requirements) and/or any kind of additional magnification and/or objects layering/overlapping and design-responsive to any resolutions as stated here) on the monitors of client PC workstations:

- Physical screen sizes: 15" and above
- Resolution: starting at 1600px horizontally and 900px vertically.
- Aspect ratios: 4:3, 16:9, 16:10, 1:1, 3:4, 9:16 and 10:16

UR007	**UR007 Display size and resolution - mobile clients functionality**

The GUI shall not prevent the usage (e.g. visible and allowing to carry out all actions without objects layering/overlapping and design-responsive to any resolutions as stated here) of portable mobile client devices (such as tablet, smartphone; incl. *dpi* dependency):

- Physical screens size: from 7’’ up to any (even hand-luggage sized)
- Resolution: any
- Aspect ratio: any
- Shaped/Cut-out displays: fully rectangular displays optimization only, w.o. cut-outs

UR008	**UR008 Credits**

The logo of the “Czech development agency” has to be visible on all GUI input screens (excl. dialogues; it may be placed in header or footer), on at least ~6% of surface of splash screens/title pages/graphical materials and at least ~4%

<table>
<tr><td colspan="2">of documents accompanying the software, all of them in accordance to the „Graphical manual“ as published on the http://www.czechaid.cz/wp-content/uploads/2016/09/Graficky-manual-CR-pomaha-a-loga.zip</td></tr>
</table>

<table>
<tr><th>UR009</th><th>UR009 Externally loaded 3rd party references</th></tr>
<tr><td colspan="2">When the application uses any 3rd party components, such as but not limited to cascading style sheets, scripts, fonts, images etc. then these have to be bundled together on the same installed instance provider as the application. Referencing from 3rd party external sources is not permitted. The bundled components have to allow such bundling and redistribution within their license.</td></tr>
</table>

9.2.4 Non-functional requirements - Performance (PR)

List of required performance measures.

<table>
<tr><th>PR001</th><th>PR001 Number of users</th></tr>
<tr><td colspan="2">Expected no. of concurrent users:
• internal min.: 60, max 100.
• external users (from the groups of Revenue service, other EU countries): may be expected at maximum 10 concurrent users (however this has to be expected for the worst case (longest response) scenario when overlapping</td></tr>
</table>

<table>
<tr><th>PR002</th><th>PR002 Number of workstations</th></tr>
<tr><td colspan="2">PCs that could be eligible to run the system:
• internal users: 500pcs+.
• external users: any</td></tr>
</table>

<table>
<tr><th>PR003</th><th>PR003 Network bandwidth usage</th></tr>
<tr><td colspan="2">• Working without any uncomfortable delay (see "Response time") on ADSL lines from 1MBit (latencies may exceed 200ms) download upwards up 4MBit (60% of clients may reach the highest theoretical download speed;)
(This delay should be solved also for including documents (files) uploads - implemented as non-blocking asynchronous GUI background batch tasks if necessary.)
• 40% of clients are employing broadband and fibre channel connections (10-20Mbit)
• Administration (NFA central office) have direct optical link.</td></tr>
</table>

<table>
<tr><th>PR004</th><th>PR004 Response time</th></tr>
<tr><td colspan="2">• It is required, that the software response to the user actions is immediate (= such a short time, that an average human cannot tell the time period, thus under approx 40 msec).
• The response for a user action requiring any data loading (does not apply for enumerations that can be reasonable preloaded) shall be either within 3 seconds (incl. GUI rendering, excl. those factors the contractor objectively could not influence in a direct or indirect manner) or has to be followed by a blocking waiting dialog with cancellation possibility or implemented as a non-blocking batch system with possibility (GUI) of management of the tasks in a cue.
• Any process/action/requirement resulting in processing time longer than tens of seconds (e.g. 1-2 minutes or more, such as report generating) has to be defined and implemented as an automated batch process with a GUI enabling the user control of the queue (manual start, scheduling, log/output lists for obtaining the results (if not agreed otherwise) and a list of running batches with the possibility of termination), as proposed and agreed in the “Analysis and system design documentation”. This does not necessarily apply for reports where appropriate.
• The abovementioned requirements have to be followed as far as the underlying platform framework allows, by using all possible means of such a platform or framework.</td></tr>
</table>

9.2.5 Non-functional requirements - Security (SR)

What threats and risks should the system address and how, including ACLs etc.

<table>
<tr><th>SR001</th><th>SR001 Information integrity and authenticity</th></tr>
<tr><td colspan="2">The requirement of information integrity means a condition of data, when they preserve their content ensuring its expected level of integrity, accessibility and efficiency and are interpreted unambiguously despite of- and protected</td></tr>
</table>

from- unpredictable influences causing their loss, denaturation, distortion and unauthorized use, where and up to the extent the contractor may be held liable for (esp. but not limited to the design of transactions, data manipulation, storage and backup solutions and their configuration design and operation during any kind of maintenance tasks).
The requirement of information veracity is the degree of data compliance of data stored in computer memory to the actual condition of the objects in a specific area of the system that they represent. This means especially the fact that the system has to ensure not to distort and/or omit/loose data being committed into it without corresponding reporting of the error or condition preventing to keep the data authenticity.

<table>
<tr><th>SR002</th><th>SR002 Information confidentiality & non-repudiation.</th></tr>
<tr><td colspan="2">Means the system has to be designed in such a way that it is providing means and resources for:
• enabling the personal liability (under the laws in force) of employees working in the system for unauthorized use and dissemination of personified confidential information (e.g. access control, logging)
• and preventing them to extent or execute their authorization beyond its original designation.</td></tr>
</table>

<table>
<tr><th>SR003</th><th>SR003 AAA</th></tr>
<tr><td colspan="2">Authentication and authorization of the system shall be done against the NFA/Ministry of agriculture SITI SSO.</td></tr>
</table>

<table>
<tr><th>SR004</th><th>SR004 Authorization groups</th></tr>
<tr><td colspan="2">Groups aggregating the various Phytosanitary register functionalities (allowed for the specified group) shall be mirrored and created in the respective SITI SSO as far as it alows.</td></tr>
</table>

<table>
<tr><th>SR005</th><th>SR005 Data record access management</th></tr>
<tr><td colspan="2">All records have to be accessible only based on the organizational structure that means, that any operator may see only those records created (and “owned”) by him, his colleagues from the same department or of his subordinate departments, if not specified otherwise for his/her user-role.</td></tr>
</table>

<table>
<tr><th>SR006</th><th>SR006 Network-level security prerequisites expectations</th></tr>
<tr><td colspan="2">The application might not be set up in any secure and consistent enterprise environment (the operators for the various organizations may not be using any kind of unified infrastructure) – this applies also for example to the local storage of configuration (if applicable) of the client application etc.
The public part of the application may be set up against the open public non-restricted access network (on technological level) thus any part of the application reachable before and also after performing authentication has to be secured against all known attacks against- and weaknesses of- all the platform whose services are exposed to the public, directly or indirectly. A suitable segregation of the computing nodes (DMZs, multi-layer networks separated by FWs etc.) should be done where appropriate (especially when protecting either personal/user data and those data entities/attributes that form the meaningful information the register is about to supervise/authorise).</td></tr>
</table>

9.2.6 Non-functional requirements - Operation and maintenance (OR)

These requirements specify what limitations and constraints the system should respect concerning its further operation and maintenance. This includes but is not limited to:

- warranty terms
- SLAs
- administration/configuration files/possibilities
- etc.

<table>
<tr><th>OR001</th><th>OR001 Operating environment</th></tr>
<tr><td colspan="2">The contractor is obliged to use the platform (Infrastructure as a service) of virtual servers provided by the NFA services, operated by the Georgian Ministry of agriculture and integrate into them as required by the platform and its operators.
Preliminary structure of the deployment may be observable from the deployment diagram earlier in this document.</td></tr>
</table>

<table>
<tr><th>OR002</th><th>OR002 Network architecture</th></tr>
<tr><td colspan="2">All inner functions of the software shall be accessible through the internal NFA network, including branch-offices using VPN (provided by the NFA)</td></tr>
</table>

OR003	**OR003 Required HW resources specification**
The contractor has to specify the expected required hardware resources consumed by its application: • as part of the tender proposal • as part of the “Analysis and system design document” at M1 • with latest update 1 calendar year before production environment start planned in the schedule The request has to be corresponding to the conditions and requirements specified by the Georgian party; there is no limitation to the total amount of resources required. • It is expected, that the specification gives the total amount of servers/CPU/RAM/storage and their respective breakdown on single servers. • The contractor is entitled to require any changes to his virtual data center at any time provided it does not exceed the total amount of resources reserved and the frequency rate of such changes remains in a reasonable limit per month. • The required configuration of resources has to be able to cover the requirements of both environments in accordance with Operation environment architecture requirement.	

OR004	**OR004 SLA**
This contract does not expect any operation, thus it does not define any SLAs with the exception of warranty SLA.	

OR005	**OR005 Warranty SLA**
(See linked document)	

OR006	**OR006 After-warranty support**
1 year after the end of warranty period a free support for the Georgian programmers shall be present (remote only).	

OR005 Warranty SLA

During the warranty period, the contractor is obliged to provide solutions in following time frames:

Error grade	Reaction time (using el. communication)	Commence of rectification - a. event occ.:	Solution time (or providing an temporary solution) - after event occurrence:
A	2 hours	*not specified*	5* hours
B	2 hours	*not specified*	1* working day (8 working hours)
C	2 hours	*not specified*	5 working days or longer, if agreed by the contracting authority

Error grade “A” means such an error that halts the operation of the system completely or prevents the execution of any of the key processes (use-cases) of the system within any step of these processes. A temporary solution of an A-grade error is regarded as a (reported) B-grade error.

Error grade “B” means such an error that either allows the operation of the system however with difficulties or using alternate or more complicated methods/approaches or prevents the execution of support processes (use-cases) within any step of these processes.

When a B-grade error preventing the execution of typically administrative processes results in the impossibility of execution of any of the key processes by any party, it is regarded as an A-grade error.

A temporary solution of a B-grade error is regarded as a (reported) C-grade error.

Error grade “C” means any other errors or the claiming of other agreed and not delivered features etc.

*) Even in case that an error may require the physical presence of the contractors representative on-site, these times may be not prolonged unless agreed in advance as a contract amendment.

9.3 Requirements catalogue - Other (software) requirements

The project-based requirements.

9.3.1 Other (software) requirements - Software legal requirements (LR)

LR001	LR001 Legitimacy and respect for human rights
The creation and operation of the system should be in accordance with the national end European legislation (specifically as enlisted in respective requirements below), international treaties and agreements, where and up to the extent the contractor may be held liable (specially concerning design and the execution of functionality of the system).	

LR002	LR002 Control
Stands for the system's ability to support and provide means for the totality of organizational and technical measures for the system, ensuring high quality of state information resources, high reliability and correctness of their use in accordance with the law and ensuring operational and convenient access to information for the user, according to its access level;	

LR003	LR003 State identification of objects (persons) of registration
It is required, that the system fully supports the state identification of objects of registration, which provides for existence of a single identification index for each of them (if not required otherwise by respective exceptions as defined in the functional requirements).	

9.3.2 Other (software) requirements - Testing requirements (TR)

How (using what tools and techniques and what processes and organization) shall the resulting product be tested. May also contain acceptance criteria as whole.

	TR001 Test cases requirements
• All tests/test cases have to reference either a requirement specified already in this Contract/Tender documents or a requirement provided during the analysis. One requirement may result in multiple tests, and one test may cover multiple requirements, if necessary (e.g. the level-of-detail should provide at least such a detail, that each test-case belongs to each function/requirement or small groups thereof). (Note: this does not apply, if necessary, for the “global” (non-functional) requirements, which may be tested either 1. within another test case (but this has to be stated as a requirement being tested by the test) 2. separately within an acceptance test or 3. within a test without a scenario: a mere “statement” test, such as “according to chapter XYZ in the documentation the application has 3 tires” or “(…) the system has a software application with functionalities covering each of the 3 modules”) • Each test has to provide a step-by-step manual for performing the relevant test, including all input values and expected outputs. • If a test may fork into multiple variants based on the data input, all possible inputs (or at least their groups – classes of values) have to be tested. • Each test case (especially in the printed version being used as annex for the acceptance M2) has to provide free space, where the result and final resolution (passed/passed with objections/failed) can be noted. • The test cases are to be described in English language (except the labels/buttons/software I/O texts and data I/O). • The first testing will be done by the contractor, but the test scenarios will be accessible and the second times run by the NFA as selected during the acceptance procedure. The above mentioned does not intend to influence nor it may not replace any internal testing procedures of the contractor.	

	TR002 User acceptance tests

Based on the test cases the system will be tested by UKZUZ and/or NFA during the acceptance procedure for M2 and M3.

9.3.3 Other (software) requirements - Migration requirements (MR)

What data/knowledge from previous instances of the product (application) shall be maintained and migrated into the new version (if any).

MR001	MR001 No migration
The project at this development stage does not foresee any data migration from no previous software versions or other data sources. This is without prejudice to (this shall not affect): • any requirements for altering design or project actions in such a manner to make this possible • or any requirements emerging in later stages of the project or later follow-up projects.	

9.4 Requirements catalogue - Other (project) requirements

9.4.1 Other (project) requirements - Project legal requirements (PL)

PL001	PL001 Contractor subject requirement
The contractor has to be an active legal subject (company) or self-employed physical person, established with accordance with its country's legislation, being able to fulfil the liabilities represented by this documentation and not undergoing any winding-up, liquidation or bankruptcy petition process.	

PL002	PL002 Contractors behaviour
Any project-related action or step (including any kind of business communication) of the contractor has to meet: • Czech law • Georgia law • International laws and contracts • Polite and respectful manners of business relations	

9.4.2 Other (project) requirements - Project management requirements (PM)

What managerial approaches and processes should be followed and what results achieved within the project of:

- development
- maintenance and operation
- testing
- implementation and setup
- handover
- etc.

Including DR&CP (Disaster recovery & Contingency planning) scenarios etc.

PM001	PM001 Project management responsibility
The contracting authority expects that the project is being managed and coordinated by the successful candidate, using any of the currently recognized formal project methodologies.	

PM002	PM002 Project documents repository

It is expected, that the candidate will also create and maintain in a central repository all the documents, that shall be output of the project, as prescribed by the list of documents (according to templates in POxxx requirements) and as prescribed by this document.

Any documentation containing information of or about the following authorization classes must not be stored in repositories publicly or 3rd party accessible (this includes also data storage services such as Google Drive, Microsoft OneDrive, Drop Box etc.):

- "secret" or "top secret"
- "confidential"
- "protected"

where:

- secret or top secret = information of data or data being classified to the specified degree by national legislation, additional requirements for the subjects handling such data may apply, however this project does not foresee handling of any of such data
- confidential = information of or data, whose disclosure may directly reveal substantial information that might lead to any kind of abuse, gaining unjustified profit or predominance of one party above another one, this concerns also private (personal) data of physical persons including sensitive data about them. Data giving hint to-, forming or directly representing the meaning for whose control and record keeping the system is mainly designated are also considered confidential.
- protected = information being not publicly available, yet their knowledge gives or might give either direct or substantial indirect opportunity, facilitation or aid for gaining access to information with higher level of protection (e.g. confidential, secret or top-secret).
- non-public = information (or data) not belonging to either of the previous categories, however requiring protection and non-disclosure since they are not meant for publication (e.g. the scope of their intended addresses or other persons meant to deal with them is enumerable by a finite list of persons, contacts, groups, roles etc.). If not belonging into any of the previous categories, the "business know-how" and similar belong to this category.
- public = information or data belonging to neither of the previous categories, that can be or are being made available or accessible to anyone without preceding knowledge or designation of the scope of those subjects.

9.4.3 Other (project) requirements - Output and actions schedule milestones

In linked document:

During the execution of the contract, following milestone deadlines have to be met:

ID	**Deadline**	**Deliverables present**
M0	(date of contract signature)	Signed contract UKZUZ + contractor
M1	M0 + 7 months	1. Acceptance protocol of system design and architecture signed by the representatives of UKZUZ and NFA (ENG/KAT) 2. Analysis and system design document – in the latest version, ENG. 3. Prototype / mock-ups (ENG) **NOTE: The deadline for M1 is the latest one. It is expected, that at least a half of functionalities shall have an earlier (4 months!) deadline. The handover and the "fixation" or "freezing" of the requirements shall be done by a partial acceptance protocol. This deadline or number of "fixed" functionalities/requirements may however be postponed or altered contractually.**
M2	M1 + 12 months	4. Acceptance protocol (ENG/KAT) 5. Analysis and system design documentation – including all annexes in their respective languages.

		+ Filled test cases by UKZUZ and/or NFA (not every test case is necessarily to be evaluated by all of them) 6. Manuals (KAT/ENG lang. where appropriate) 7. Software - installed in production environment as ready-to-manufacture version and the same in testing environment - commented and prepared source codes 8. Project documentation pack version to the date M2 minus 2 weeks. 9. Training presence lists (signed by all attendees) – ENG/KAT language (may be postponed +3 months if necessary). NOTE: This acceptance is being done on-site in Tbilisi, Georgia with at least one representative of the contractor (authorized to fully represent the contractor at least in the entire scope of the project – including contractual agreements) present during the formal handover procedures (does not necessarily apply on testing).
M3	M2 + 24 months	10. Handover protocol 11. Analysis and system design documentation – warranty claims resulting changes applied + changelog documents including all annexes in their respective languages. 12. Software + Source codes with changes applied 13. Manuals and other user documentation with updates (ENG/KAT) 14. Test protocols relevant to the changes.

Please note, that the abovementioned deadlines M1, and M2 reflect the **final date** of the acceptance procedure of the outputs, however some of the outputs (especially the installed software and "Analysis and system design document") have to be present in their latest version for the respective milestone certain time earlier, as required by the testing and acceptance procedure defined within the Contract (that means at least 2 comments iterations, each in the duration of 10 working days/2 weeks at least, the mock-ups for at least one iteration of comment).

Based on the test results the delivery may be accepted, accepted with objection or not accepted at all. When a single test fails in such a manner, which makes the usage of the system or one of its use-cases or vital functionalities impossible, the system will not be accepted. However, in such a case (repeated acceptance) only that test, that have failed and have been corrected, shall be re-done. Those test results that are not influenced by the bug being corrected, may be reused from previous acceptance testing.

9.4.4 Other (project) requirements - Project outputs requirements (PO)

A note on the structure and list:
The meta-model diagram of documentation as well its descriptions are featuring the expected obligatory structure and minimum content of each of the items (minimum information given, e.g. elements and their relations used or minimum table-of-contents outline).

Note on the mandatory / optional items:
While an item in the following text or in the referred model may be labelled as "Optional" (or "OPT"), its inside textual description describing its outline may bear the tags "Mandatory". This is

no way contradictory, it only means, that "SHOULD the documentation item be used (if necessary) THAN (and only then) it shall follow at least those outlines or mandatory elements/links required in its description."
The contractor may append any other diagrams/models/notation recognized as necessary during the project.

9.4.4.1 Project outputs requirements (PO) - General

<table>
<tr><th>PO000</th><th>PO000 Project outputs</th></tr>
<tr><td colspan="2">The structure and requirements for project outputs refer to the appropriate diagram "Delivery content (incl. metamodel) - Documentation structure"</td></tr>
</table>

<table>
<tr><th>PO001</th><th>PO001 Output forms - methodical</th></tr>
<tr><td colspan="2">Any diagrams or schemes being part of the project output in neither technological representation should be created:
• Using a formal notation such as the preferred UML (1.1 and above) or BPMN, EPC or ArchiMate (version 3.x preferred).
OR
• (especially for smaller/high-level schemas or schemas intended for end users) Using any notation, supposing all the elements used within the schema are covered in the appropriate legend.</td></tr>
</table>

<table>
<tr><th>PO002</th><th>PO002 Output form - technological</th></tr>
<tr><td colspan="2">The system documentation described as <b>"MODEL"</b> shall be delivered
• as a shared model (no important what technique, if a mailed/shared eapx/feap file, XMI imports/exports including those realized using the CVS integration or shared SEA cloud or database)
• as a model report (DOC(X) or PDF with reasonable size, split into multiple documents and reasonable comments period)
with an exception of those marked in dark orange colour in the diagram (Data-logical diagram, Interfaces layout schemes/designs and Components breakdown and their functions), that are to be delivered as readable by non-it personnel (a specific SEA report template fulfilling this requirement is also acceptable).

The system documentation described as <b>"USER DOCUMENTATION"</b>, while it may be stored in the model as empty packages containing linked documents only, shall be delivered as:
• Human written and non-IT specialist readable Office format documents (PDF, DOCX, XLSX, PPTS, JPEG/PNG/BMPs or open software variants thereof).

The documentation described as <b>"PROJECT DOCUMENTATION"</b>, being stored separately, shall be delivered in those formats the contractor sees fit, if not able to export / preview in software without a licence or free of charge, the contractor has to provide at least 4 licenses to the contracting authority for the entire duration of the project including its warranty period.</td></tr>
</table>

<table>
<tr><th>PO003</th><th>PO003 Trackback</th></tr>
<tr><td colspan="2">For all SEA model elements applies:
All these have to be traced back either to any meeting or other kind of dialogue within the analysis phase or to the original requirements within this Tender annex. Should any chapter/requirement of this annex prove as obsolete, this has to be mentioned explicitly in the “Analysis and system design document”.</td></tr>
</table>

<table>
<tr><th>PO004</th><th>PO004 Language</th></tr>
<tr><td colspan="2">All project outputs shall be in English language, with exception:
• Software I/O on GUI as per UR004
• User and administration manuals, same multilingual rules apply as UR004</td></tr>
</table>

9.4.4.2 Project outputs requirements (PO) - Analysis and system design documentation

This document has to address following aims:

- The “central point” for capturing of requirements, architectural concepts and decisions
- The reference (may be auto-generated) for (public as well private/protected etc.) modules, classes/source files, methods, variables and data entities (persistently stored as well used inside the run-times)

It is required, that the “Analysis and system design document” exists in the following versions, e.g. is created and updated (maintained) at least for following stages of the project:

- Initial writing – the version is to be embodied by the accepted document after the analysis&design phase (= delivered by M1)
- Finalization for the project acceptance (delivered by M2) – this version should incorporate seamlessly all the changes done during the project implementation & testing and has to reflect exactly the state of the software/system as accepted during the acceptance phase.
- Finalized version + differential change lists (annexes) AND entirely updated document if any warranty corrections should be done (delivered by M3) – Documents recording the changes that have resulted from any warranty claims. Change lists should refer to relevant warranty claims or have their own acceptance clause, the updated document(s) (incl. source code) are to be delivered only as electronic version with all the factual changes resulting from changelogs have to be reflected.

The minimum content is as follows:

<table>
<tr><th>PO005</th><th>PO005 (updated until M1, M2, M3) Business input and requirements</th></tr>
<tr><td colspan="2">Consisting of (horizontal spec.):
• Requirements catalogue, issues and the changed/added requirements (SEA "Custom diagram"),
• and thereof resulting users/roles (UML or ArchiMate)
• carrying out their respective use-cases and/or process(es) on the data, (UML/ArchiMate, BPMN where necessary)
• enlisted as a full list of data entities existing throughout the entire system and their relations, including full list of their attributes (logical perspective, not physical) (UML class diagram).
Optionally: data entities may be additionally represented multiple times with specification of variants of each used for persistence, data flow within the I/O of various interfaces etc. (for example: while a person may have the set of 20 attributes, this data entity may be represented not only as "general one" having 20 attributes, but also as a data entity of some say WS or GUI, where only 10 of the attributes are used).

Minimum level-of-detail (LOD - vertical spec.):
• All requirements/issues/changes independently on their respective LOD
• user/roles independently on independently on their LOD, entire list of roles (and their permitted functionalities) must be present
• data entities on logical level up to their atomic attributes, where the word "atomic" means "representable as some electronically recognized data type"</td></tr>
</table>

<table>
<tr><th>PO006</th><th>PO006 (updated until M1, M2, M3) System architecture design</th></tr>
<tr><td colspan="2">Consisting of (horizontal spec.):
• Architecture (breakdown) of modules/components of the system and their respective functionality (ArchiMate - app layer (preferred) OR UML Component)
• Interfaces design/layout of GUI, WSs, APIS containing: For webservices: methods list, their I/Os attributes and exceptions. For GUI: rich textual description of the functionalities, schemas up to the detail of data attributes (depicted as input fields labels and list/table headers) and functionalities, featuring at least one example representing also the graphical layout (template) for each type of functionality.
• Detailed data documentation for each of the module (data structure in the database, data structure of I/O messages etc. - note: the diagram may contain only the DB persistence, the other may be and are encouraged to be part of the</td></tr>
</table>

respective documentation of WSes, system classes (such as MVC - Model) etc.) (ERD / UML Data modelling diagrams)

- Deployment diagram (how the respective modules/components are deployed and where (ArchiMate technol. layer (preferred) OR UML deployment diagram)
- Test cases (UML testing)

Note: As one may reason from the text above, the architecture design documentation is required only in the form of a "static" perspective. However, a dynamic (process-oriented) documentation has to be added where the collaboration of multiple parts of the system may not be easily deduced. However, this dynamic view documentation does not have to be comprehensive (done for each of) for all the functions/functionalities/cases/processes intended within the system.

Minimum level-of-detail (LOD - vertical spec.):

- Architecture of modules/components of the system = a "module" is considered up to a component having the size of single I/O screen/report or its part devoted to a single or prevailing data entity, proposed minimum LODs:

1. General overview of modules (= system and its main parts forming different binaries to be setup on different servers)
2.(or up to 3.) Modules structure (what pages/templates/ function groups/WSs/robots or automated agents triggered events etc. the part consists of)
OPT: 3.(or up to 4.) Inside parts structure (control sets/custom control/ custom sub-templates/WS methods) does each page/function group/WS consist of.

- their respective functionality = full list of functions that the part (on level 1.-2. above, optionally: 3.-4.) provides, the LOD of the functions is let up to the discretion of the contractor
- Data (physical representation) up to their atomic attributes, where the word "atomic" means "representable as some electronically recognized data type"
- Deployment diagram = components representable as artefacts, e.g. packages deployable on nodes, nodes and infrastructure describing a technologically configurable environment
- Test cases (UML testing) = at least in such a detail, that each test-case belongs to each function/requirement or small groups thereof.

9.4.4.3 Project outputs requirements (PO) - Manuals

As annexes to the Analysis and system design document, following system manuals are to be delivered as addition to the abovementioned documentation (each in all languages as indicated by the abbrev. in braces):

PO007	**PO007 (until M2, updated if necessary until M3) The user manual (ENG/KAT)**
Fully describing all of the functionalities and processes covered by the software and the ways of their usage designated for all roles of end-users, excl. user-side administrators.	

PO008	**PO008 (until M2, updated if necessary until M3) The user manual for administrator**
Fully describing all of the functionalities and processes covered by the software and the ways of their usage designated for the user-side administration tasks (such as for example the editing of registry or selection lists if not covered by a (manual for a) specific GUI).	

PO009 (updated until M2, M3) Technological administrator and deployment manual	• Deployment and technological architecture manual, including 1. full step-by-step manual for installation of the application (including basic schemas of application layout and setup possibilities, if applicable), 2. the documentation of all the configurable variables in any configuration repositories (such as but not limited to: the Microsoft Windows registry, any text/xml/csv/tabbed/json and similar files, the database, configuration repositories of any kind in any proprietary software including the RDBMS being used, templates/reports and the available variables for the report templates configuration; if a 3rd party software is used, then the documentation may be covered by the documentation of the 3rd party software itself, however an addendum has to be made about how the configuration influences the delivered software) and 3. Network architecture and ports used between the components (may use the

	"Deployment architecture" diagram). 4. Backup & recovery plan containing: • Analysis of data assets (may be part of data models) • Architecture of backup solution(s) being employed • Data assets <--> backup solution(s) (which asset is backed up by what solution and with what frequency, type of backup etc.) • Steps to check the backups being done correctly (per typ of backup solution) or (should this require manual operation) the manual for performing these steps. • Steps to recover each data asset (per type of backup solution)

PO010	**PO010 Developer documentation (workstation setup + licenses)**
• Development prerequisites needed for the code to be run (including 3rd party tools installers where possible or necessary from the point of view of the licensing terms) incl. processes and means of keeping track of changes and patching changes made during the warranty period. + Development environment setup guide (may include virtual machines images, services mock-ups etc., however these do not substitute the step-by-step guide) - this does not apply for those (parts of) developer seats corresponding to standard administration tools for SITIAgri plaform. • Source codes files / configuration projects and resources structure overview and description (UML: any kind of structured overview of the files groups and their link to respective modules (architecture modules - LOD 1.)) + their inner logical program design structure (classes, object model)	

PO011	**PO011 API Developer documentation**
Public and internal API (each as separate document!) developer documentation, one for each type of an endpoint (SOAP/DLL/COM, header files, network endpoints etc.) describing: 1. Methods exposed (including their link to relevant requirements and the processes it belongs to) 2. Their I/O data and their attributes up to atomic items (primitive datatypes), including their allowable ranges and content types/forms; in case of numbers or lists influencing the flow of the handling of the data also with documentation on the specific values and the consequences on the process for each of them. 3. Their exceptions and errors that can be thrown/returned, their possible reasons (as a short textual fault analysis guide) and possible solutions.	

9.4.4.4 Project outputs requirements (PO) - Project documentation

PO012	**PO012 (continuous updates) Project documentation**
The project (management) documentation shall consist of: • Contractual docs (e.g. the contract, its attachments, amendments and contractor proposal in the tender) NOTE: While the electronic copy of these document shall be accessible within the repository, the contractor does not have to provide it back to the awarding authority when delivering the delivery at any milestone. • Cooperation requirements plan (required role + schedule) - may be done also as a set of (outlook or other similar PIM) invitations with at least one month in advance • Repository of meeting minutes. • Project plan as per PO013 including schedule	

PO013	**PO013 (continuous updates) Project plan**
A project plan has to be created and further maintained by the candidate (contractor), containing at least the following information: A) An organizational a managerial structure of the project, consisting of: • Roles of project participants • and their mutual relations • and their competencies within the project and outside the project (towards the broader scope of stakeholders around the project). • The name of physical persons acting within these roles (employees or equivalent of the contracting authority as well the contractor). B) The project schedule (based on the required schedule as described in these terms and requirements), including the milestones of the project, that have been identified and set during the preliminary as well main analytic phases.	

It is required, that at least following phases and their items are scheduled within the schedule (the following list does not cover those phases/items, which arise or might arise from other requirements):
1. (phase) Analysis: (item) physical presence for analysis phase on-site
2. (phase) Analysis: (item) creation of the “Analysis and system design documentation”
3. (phase) Analysis: (item) creation and opponency of schemas/mock-ups or beta-version of the GUI of the final system
4. (phase) Analysis: (item) opponency & acceptance process of the “Analysis and system design document”
5. (phase) Analysis: (milestone) Acceptance of the “Analysis and system design document”
6. (phase) Testing: (item) physical presence for supporting the (acceptance) testing phase on-site
7. (phase) Testing: (item) Acceptance testing by the contracting authority
8. (any appropriate phase) training periods
9. (phase) Support: (item) Support (remote or on-site in Georgia, when needed) for warranty claims that might emerge during the production usage of the system within the warranty period agreed within the contract.

C) List and contents (and preferably templates) of relevant project documents

D) Project meeting types and schedules

E) Other project-dependent processes and their agendas including the required cooperation by the contracting authority

9.4.4.5 Project outputs requirements (PO) - Software

As annexes to the Analysis and system design document, following software are to be delivered:

PO014	**PO014 (continually) Software - installed instance**

As a part of the delivery, the software has to be installed (and covered by all licenses required):
- (until M1) Mock-ups /beta-version prototype in any remote-access capable environment provided by the contractor, based on the “Analysis and system design document” initial writing version
- (until M2) Set up in designated testing/production environment as beta version / launch version.
- (until M3) Code changes applied and delivered according to the warranty period changes. + testing protocols for those (regression) tests that are relevant to these changes.

PO015	**PO015 (continually, optionally) Software - updates and patches**

It may be, that by recommendation, no 3rd party employees will have access to the production environment. In such case following documents and sources have to be provided for each patching/update of the production software (typically bugs correction during the warranty period), this also applies for client-side software if necessary:
- Patch binaries/install packages in a form best ready for the installation with the smallest number of steps required (for example: source code is all right, if the installation means the copying of scripts into the web-app directory or compilation with heavy optimization for the target platform that cannot be done in the development environment or is required due to security reasons or considerations; however pre-compiled binaries for Copy&Paste or registration or one-click install – such as but not limited to installation scripts/projects - are preferred)
- Patch manual – step by step guide for installing the patch, including the editing of configuration files etc.
- Patch tools – any tools necessary for the execution of the steps described in the manual
- Client-side software only: Installation tool suitable for user-side install done by operator with little or no IT-related knowledge or suitable for remote/bulk deployment on the target platform.

The delivery mentioned above may be omitted if agreed otherwise with the NFA during the analysis.

PO016	**PO016 (updated until M2, M3) Software - source codes**

- As a part of the delivery, the entire source codes created or being used within the development of the software in any way, unless licensed separately as 3rd party closed binaries, have to be handed over during handover/acceptance (M2, M3).

All source codes have to meet following requirements:
- Commented in such a level of detail, that at least blocks of code performing complex functions are being described. The comments have to be exclusively in English language.
- They all meet the same form of indenting, brackets usage etc.

- All variables (including any persistent data such as DB structure description elements and including runtime data such as method parameters or API attributes) have to be named in English language (when the variable name is representing any meaningful information, this does not apply for indexing variables, technological filed/stack/pointer counters etc.)

This does not apply for those packages/scripts/objects and other entities of the systems, where such commenting is not possible or some of the concepts (such as variables, bracketing/indenting etc.) are not being used.

- Any changes made during the warranty period have to be handed over as well (as changelog, differential patches or similar), including one last handover of the entire (patched) code at the end of the warranty period or the whole set of patches once more.
- During the warranty period of at least 2 years the contracting authority (or any 3rd party in contracted by them) is/are not entitled to make any changes of the software without providing the relevant differential patch being shared and accepted by the contractor by means as described and approved in the “Analysis and system design document” by the contractor.
- The contractor is then required to patch his version of source codes (or similar packages/projects/configurations) in order to be able to provide warranty changes.
- The contractor is not to be held liable for any bugs resulting from within these patches. For clarity the word “resulting” means in the context of this paragraph that either the patched lines/segments of code/configuration entities either throw an unexpected/unhandled error or they alter the data of the application in such a way, that any other lines/segments/configuration entities, even the original ones, throw an unexpected/unhandled error due to this concrete data value. “Data” means not only persistent ones (files, database etc.), but also the runtime data (variables, temporary files, calling parameters of functions/methods, network transported data etc.) within any part of the software application system.
- The above also applies proportionally for various backups/exports etc. of in-system specifications of datasets, forms, commands/actions etc. as far as the underlying platform allows.

9.4.5 Other (project) requirements - Minimum project actions (tasks) requirements (PA)

<table>
<tr><th>PA001</th><th>PA001 Project actions generel</th></tr>
<tr><td colspan="2">The contractor shall ensure and perform all steps necessary to reach the goal and aim of the project - the delivery of the system as required by the requirements specified in this document. Above this, following mandatory steps cannot be omitted:</td></tr>
</table>

<table>
<tr><th>PA002</th><th>PA002 Analysis on-site</th></tr>
<tr><td colspan="2">It is required, that the contractor provides at least one physical presence on-site (Tbilisi, Georgie) for performing the process of analysis within the proposed scope.</td></tr>
</table>

<table>
<tr><th>PA003</th><th>PA003 Analysis in Czech Republic</th></tr>
<tr><td colspan="2">It is required, that the contractor performs at least one physical presence in Czech Republic (Prague) for performing the process of analysis within the proposed scope.</td></tr>
</table>

<table>
<tr><th>PA004</th><th>PA004 Acceptance testing on-site</th></tr>
<tr><td colspan="2">It is required, that the contractor provides at least one physical presence on-site (Tbilisi, Georgia) in order to support process of testing within the proposed scope.</td></tr>
</table>

<table>
<tr><th>PA005</th><th>PA005 User training on-site</th></tr>
<tr><td colspan="2">It is required, that the contractor performs an on-site training in English language for following target audience:
• 5 system administrators if at least 1 nominated by the Georgian side (IT administrators/developers as well data/user administrators of total max. 5 persons)
• 20 users if at least 1 nominated by the Georgian side
in the total length of at least 2 days (1 day if no users or no administrators are nominated, 0 days if no users nor administrators are nominated), covering at least following themes:

<u>IT administrators/developers:</u>
• Source code structure and blueprints/patterns used and employed</td></tr>
</table>

- System architecture and design (incl. DB entities and relations and key sets of namespaces/packages/classes)
- Internal API of the application (incl. DB objects such as stored procedures, triggers etc.)
- Application installation and setup parameters (configuration, registry, command line parameters)
- Trainees questions

(All the above-mentioned themes especially in such a manner, so that is explaining combination and influence between the various parts of the system.)

Administrators – data/app:

- Explanation of the admin GUI
- Administrative tasks and the performing their of incl. the demonstration of the impacts in the user-side
- Trainees questions

Users:

- The application GUI and generic tasks/procedures - handling all the processes as per functional requirements.
- The use-cases and the performing thereof; incl. the methodology and methodical remarks
- Advanced tasks accessible to the users but not necessary for everyday usage
- Demonstration of admin GUI and the impact on the user-side of the application

9.4.6 Other (project) requirements - Cooperation requirements (PC)

The candidate shall indicate in his proposal, what kind and range of cooperation he expects from the contracting authority. This should be done for the following parts of the project:

PC001	**PC001 Project management**
Cooperation needed for project management	

PC002	**PC002 System design, implementation, testing, acceptance**
Cooperation needed for system design, implementation, testing and acceptance	

PC003	**PC003 Production deploy & operation**
Cooperation needed for system production run within the warranty period	

PC004	**PC004 Risks**
Cooperation needed for acceptance, mitigation, transfer or prevention of risks (in such case a risk analysis has to be performed, listing all threads possible, the likelihood of their occurrence, impact analysis and the proposed type of reaction).	

PC005	**PC005 Other cooperation**
Other cooperation needed incl. training	

9.4.7 Other (project) requirements - Risks to be addressed (RR)

Note: Issues and requirements in this chapter are here for:

- information to the candidate
- as cooperation advice, since the solutions used by 3rd party to address or solve these risks / issues may result into other requirements for the software. The chosen candidate has to cover these solutions as far as they can be fairly held liable within the contract.

RR001	**RR001 GE legislation**
Currently, the state of the legislation in Georgia does not provide any mandatory requirement for the relevant subjects to be registered in the phytosanitary register. The legislation has to ensure, that there is mandatory registration (either by solving RR002 or other means).	

RR002	RR002 Ensuring data flow from justice register
Currently, the justice register: • does not provide list of new "unknown" registrations (the ID of registration has to be known to obtain further details of the subject being registered) • might overflow the phytosanitary register with subjects, that have applied for the business of handling of phytosanitary material only for precaution, but are not handling any material yet. The data from justice register have to be filtered and acquired in a reasonable and useful manner.	

RR003	RR003 Determining the type of registered subjects
The current legislation does not address the terms and conditions for determining what subjects shall have / will have the duty to be registered. The legislation has to provide this clearly and a simulation and/or an estimate has to be made for determining the number of business subjects registration numbers (no. of subject registered per given time frame). This also has impact on the RR004.	

RR004	RR004 Evaluation of number of users
The administration processes for the phytosanitary register are not entirely clear at the present moment of writing. The administration/government processes of the phytosanitary register have to be layed down clearly, especially to determine: • the final number of registered subjects (per time frame, see RR003), • the final number of users using the system • and eventually some details concerning the deployment architecture (branch offices etc.).	

9.5 Requirements catalogue - Tender proposal requirements - Cooperation requirements (PC)

(recapitulation of proposal requirements identical as specified in 9.4.6.)

The candidate shall indicate in his proposal, what kind and range of cooperation he expects from the contracting authority. This should be done for the following parts of the project:

PC001	PC001 Project management
Cooperation needed for project management	

PC002	PC002 System design, implementation, testing, acceptance
Cooperation needed for system design, implementation, testing and acceptance	

PC003	PC003 Production deploy & operation
Cooperation needed for system production run within the warranty period	

PC004	PC004 Risks
Cooperation needed for acceptance, mitigation, transfer or prevention of risks (in such case a risk analysis has to be performed, listing all threads possible, the likelihood of their occurrence, impact analysis and the proposed type of reaction).	

PC005	PC005 Other cooperation
Other cooperation needed incl. training	

10. Requirements catalogue - Tender proposal requirements

The technical part of the contractor (tender contestant) proposal should contain:

Tender001 Preliminary architecture

Preliminary architecture (as per PO006 - modules breakdown) (may be based on the diagrams being part of this document) specifying concrete platforms, techniques, concepts and 3rd party tools the contractor is expecting to use in order to deliver the various parts/modules of the solution.

Tender002 Preliminary data architecture

Preliminary data architecture (as per PO005 or PO006 - data documentation/entities) (may be done as non-contractually obligatory database data model) refining as observed and analysed from the data diagram and functional requirements.

Tender003 Interface schemes

3-5 schemas of selected interfaces (as per PO006 - Interfaces design) with statement what parts/characteristics are typical for the platform used and which may be customized as part of the project, featuring at least one (general) graphical layout example of either previously done applications or designed specifically for the contract presenting the proposed graphical design.

Tender004 Project management and testing

Project procedures and testing (internal) the contractor is intending to employ within the project.

Tender005 Patching cooperation

Project procedures the contractor is intending for the patching and cooperation with the developers during the warranty period (when both sides - the users, developers and the contractor - may alter the code).

Tender006 Project plan

Preliminary project plan (as per PO013) featuring yellow "TODO" in those places the contractor cannot be yet sure of and additionally including:

1. gaps and questions currently open from the point of view of the contractor,
2. cooperation requirements as per PCxxx (including preliminary risk analysis if applicable).

Tender007 Preliminary HW requirements

As per OR003 the required HW for deploying the solution has to be specified.

(optional)Tender008 License pricing and maintenance/assurance

If any additional licenses are to be acquired for the solution, the tender proposal has to enumerate:

- the price for such a licence
- any kind for a maintenance / assurance for 2 years