

7.1.2. Интеграционен слой

- Трябва да бъде разработен и внедрен служебен онлайн интерфейс за машинен обмен на данни и предоставяне на вътрешно-административни електронни услуги към информационни системи и регистри на други администрации, публични институции и доставчици на обществени услуги, съгласно действащите изисквания за оперативна съвместимост. Трябва да бъде предвидена интеграция с първични регистри чрез стандартен междинен слой или чрез националната схема за електронна идентификация – конкретната реализация трябва да бъде одобрена от Възложителя след приключване на етапа на бизнес-анализ.

7.1.3. Технически изисквания към интерфейсите

Приложните програмни интерфейси трябва да отговарят на следните архитектурни, функционални и технологични изисквания:

- Служебните онлайн интерфейси, които предоставят уеб-услуги (Web Services) и да осигуряват достатъчна мащабируемост и производителност за обслужване на заявки в реално време, с минимално време за отговор на заявки под 1 секунда за 95% от заявките, които не включват запитвания до регистри и външни системи. Изпълнителят трябва да обоснове прогнозирано натоварване на Системата и да предложи критерии за оценка на максимално допустимото време за отговор на машинна заявка. Критерият за оценка следва да се основава на анализ на прогнозираното натоварване и на наличния хардуер, който ще се използва. Изпълнителят трябва да представи обосновано предложение за минималното време за отговор на заявка на база посочените по-горе критерии и да осигури нужните условия за спазването му;

- Да бъде предвидено създаването и поддържането на тестова среда, достъпна за използване и извършване на интеграционни тестове от разработчици на информационни системи, включително и такива изпълняващи дейности за други администрации или бизнеса, с цел по-лесно и устойчиво интегриране на съществуващи и бъдещи информационни системи.

7.1.4. Електронна идентификация на потребителите

Неприложимо.

7.1.5. Отворени данни

Неприложимо.

7.1.6. Формиране на изгледи

Потребителите на Системата трябва да получават разрези на информацията чрез филтриране, пренареждане и агрегиране на данните. Резултатът се представя чрез:

- Визуализиране на таблици;
- Графична визуализация на екран;
- Разпечатване на хартиен носител;
- Експорт на данни в един или няколко от изброените формати – ODF, Excel, PDF, HTML, TXT, XML, CSV.

7.1.7. Администриране на системата

Системата трябва да осигурява администриране на потребителите и правата за достъп.

7.2. Нефункционални изисквания към информационната система

7.2.1. Авторски права и изходен код

▪ Всички компютърни програми, които се разработват за реализиране на Системата трябва да отговарят на критериите и изискванията за софтуер с отворен код;

▪ Всички авторски и сродни права върху компютърните програми, които ще бъдат разработени по заданието на Възложителя, техният изходен програмен код, дизайнът на интерфейсите и базите данни, и всичката техническа документация, чиято разработка е предмет на поръчката, трябва да възникват за Възложителя в пълен обем, без ограничения в използването, изменението и разпространението им;

▪ Приложимите и допустими лицензи за софтуер с отворен код са:

- GPL (General Public License) 3.0
- LGPL (Lesser General Public License)
- AGPL (Affero General Public License)
- Apache License 2.0
- New BSD license
- MIT License
- Mozilla Public License 2.0

7.2.2. Системна и приложна архитектура

▪ Системата трябва да бъде реализирана, като разпределена модулна информационна система. Системата трябва да бъде реализирана със стандартни технологии, и да поддържа общо приети комуникационни стандарти, които ще гарантират съвместимост на системата с бъдещи разработки. Съществуващите модули функционалности трябва да бъдат

рефакторирани и/или надградени по начин, който да осигури изпълнението на настоящето изискване;

- Бизнес процесите и услугите трябва да бъдат проектирани колкото се може по-независимо, с цел по-лесно надграждане, разширяване и обслужване. Системата трябва да е максимално параметризирана и да позволява настройка и промяна на параметрите през служебен (администраторски) потребителски интерфейс;

- Трябва да бъде реализирана функционалност за текущ мониторинг, анализ и контрол на изпълнението на бизнес процесите в Системата;

- При разработката, тестването и внедряването на Системата Изпълнителят трябва да прилага наложени се архитектурни (SOA, MVC или еквивалентни) модели и дизайн-шаблони, както и принципите на обектно ориентирания подход за разработка на софтуерни приложения;

- Системата трябва да бъде реализирана със софтуерна архитектура ориентирана към услуги - Service Oriented Architecture (SOA);

- Взаимодействията между отделните модули в Системата и интеграциите с външни информационни системи трябва да се реализират и опишат под формата на веб-услуги (Web Services), които да са достъпни за ползване от други системи в държавната администрация, а за определени услуги – и за гражданите и бизнеса; За всеки от отделните модули/функционалности на Системата следва да се реализират и опишат приложни програмни интерфейси – Application Programming Interfaces (API). Приложните програмни интерфейси трябва да са достъпни и за интеграция на нови модули и други вътрешни или външни системи;

- Приложните програмни интерфейси и информационните обекти задължително да поддържат атрибут за версия;

- За всеки отделен приложен програмен интерфейс трябва да бъде разработен софтуерен комплект за интеграция (SDK) на поне две от популярните развойни платформи (.NET, Java, PHP);

- Системата трябва да осигурява възможности за разширяване, резервиране и балансиране на натоварването между множество инстанции на сървъри с еднаква роля;

- При разработването на Системата трябва да се предвидят възможни промени, продиктувани от непрекъснато променящата се нормативна, бизнес и технологична среда. Основно изискване се явява необходимостта информационната система да бъде разработена като гъвкава и лесно адаптивна, която отчита законодателни, административни, структурни или организационни промени, водещи до промени в работните процеси;

- Изпълнителят трябва да осигури механизми за реализиране на бъдещи промени в Системата без промяна на съществуващия програмен код. Когато това не е възможно, времето за промяна, компилиране и пускане в експлоатация трябва да е сведено до

минимум. Бъдещото развитие на Системата ще се налага във връзка с промени в правната рамка, промени в модела на работа на потребителите, промени във външни системи, интегрирани със Системата, отстраняване на констатирани проблеми, промени в модела на обслужване и т.н. Такива промени ще се извършват през целия период на експлоатация на системата, включително по време и гаранционния период;

- Изпълнителят трябва да проектира, подготви, инсталира и конфигурира като минимум следните среди за Системата: тестова и продукционна;

- Системата трябва да бъде разгърната върху съответните среди (тестова и продукционна);

- Мрежата на държавната администрация (ЕЕСМ) ще бъде използвана като основна комуникационна среда и като основен доставчик на защитен Интернет капацитет (Clean Pipe) за връзките с националните контактни точки и компетентните институции от държавите-членки – изискванията на софтуерните компоненти по отношение на използвани комуникационни протоколи, TCP портове и пр. трябва да бъдат детайлно документирани от Изпълнителя, за да се осигури максимална защита от хакерски атаки и външни прониквания, чрез прилагане на подходящи политики за мрежова и информационна сигурност от Възложителя в ЕЕСМ;

В Техническото си предложение, участникът трябва да опише добрите практики, които ще прилага по отношение на всеки аспект от системната и приложната архитектура на Системата;

- Трябва да бъде създаден административен интерфейс, чрез който може да бъде извършвана конфигурацията на софтуера;

- Всеки обект в системата трябва да има уникален идентификатор;

- Записите в регистрите не трябва да подлежат на изтриване или промяна, а всяко изтриване или промяна трябва да представлява нов запис.

7.2.3. Повторно използване (преизползване) на ресурси и готови разработки

Проектът следва максимално да преизползва налични публично-достъпни инструменти, библиотеки и платформи с отворен код.

За реализацията на системата следва да се използват в максимална степен софтуерни библиотеки и продукти с отворен код.

Подход за избор на отворени имплементации и продукти

За реализацията на дадена техническа функционалност обикновено съществуват множество отворени алтернативни проекти, които могат да се използват в настоящата система. Участникът следва да представи базов списък със свободните компоненти и средства, които възнамерява да използва. Отворените проекти трябва да отговарят на следните критерии:

1. За разработката им се използва система за управление на версиите на кода и е наличен механизъм за съобщаване на несъответствия и приемане на допълнения;
2. Да имат разработена техническа документация за актуалната стабилна версия;
3. Имат повече от един активен програмист, работещ по развитието им;
4. Имат възможност за предоставяне на комерсиална поддръжка;
5. Няма намаляваща от година на година активност;
6. По възможност проектите да са подкрепени от организации с идеална цел, държавни или комерсиални организации;
7. По възможност проектите да имат разработени unit tests с code coverage над 50%, а проектът използва Continuous Integration (CI) подходи – build bots, unit tests run, регулярно използване на статични/динамични анализатори на кода и др.

Препоръчително е преизползването на проекти, финансирани със средства на Европейския съюз, както и такива, в които Участникът има активни разработчици. Използването на closed source и такива с платен лиценз инструменти, библиотеки, продукти и системи става за сметка на изпълнителя, като е допустимо в случаите, когато липсва подходяща свободна алтернатива с необходимата функционалност или тя не отговаря на горните условия.

Изпълнителят трябва да осигури поддръжка от комерсиална организация, развиваща основните отворени продукти, които ще бъдат използвани като минимум за операционните системи и софтуерните продукти за управление на базите данни.

Подход за работа с външните софтуерни ресурси

При използването на свободни имплементации на софтуерни библиотеки е необходимо да се организира копие (fork) на съответното хранилище в общото хранилище за проекти с отворен код, финансирани с публични средства в България (към момента <https://github.com/governmentbg>). Използващите свободните библиотеки компоненти задават за upstream репо хранилищата в областта governmentbg, като задължително се реферира използваната версия/commit identifier.

Когато се налага промяна в изходния код на използван софтуерен компонент, промените трябва да се извършват във fork хранилището на governmentbg в съответствие с изискванията на основния проект. Изпълнителят трябва да извърши необходимите действия за включване на направените промени в основния проект чрез pull requests и извършване на необходимите изисквани от разработчиците на основния проект промени до приемането им. Тези дейности трябва да бъдат извършвани по време на целия проект.

При установяване на наличие на нови версии на използваните проекти се извършва анализ на влиянието върху настоящата система. В случаите, при които се оптимизира използвана функционалност, отстраняват се пропуски в сигурността, стабилността или

бързодействието, новата версия се извлича и използва след успешното изпълнение на интеграционните тестове.

7.2.4. Изграждане и поддръжка на множество среди

Изпълнителят трябва да изгради и поддържа минимум следните логически разделени среди:

Среда	Описание
Development	чрез Development средата се осигурява работата по разработката, усъвършенстването и развитието на Системата. В тази среда са налични и допълнителните софтуерни системи и инсталации, необходими за управление на разработката – continuous integration средства, системи за автоматизирано тестване и др.
Testing	чрез Testing средата се извършват тестове, преди разгръщане на нова версия от Development средата върху Production средата. В нея се извършват всички интеграционни тестове, както и тестовете за натоварване. Всички, които трябва да се интегрират към Системата могат да тестват в нея интеграцията си, без да застрашават работата на продукционната среда.
Production	това е средата, която е публично достъпна за реална експлоатация и интеграция със съответните външни системи и услуги.

Управлението на средите трябва да става чрез автоматизирана система за провизиране и разгръщане на системните компоненти. При необходимост от страна на Възложителя, Изпълнителят трябва да съдейства за изграждането на нови системни среди.

Участникът може да предложи изграждането на допълнителни среди според спецификите на предложеното решение.

7.2.5. Процес на разработка, тестване и разгръщане

В случай, че върху част от компонентите, нужни за компилация, има авторски права, те могат да бъдат или в отделно хранилище с подходящия за това лиценз, или за тях трябва да бъде предоставен заместващ „mock up“ компонент, така че да не се нарушава компилацията на проекта.

За всеки един разработван компонент Изпълнителят трябва да покрие следните изисквания за гарантиране на качеството на извършваната разработка и на крайния продукт:

- Документиране на Системата в изходния код, минимум на ниво процедура/функция/клас;

- Покритие на минимум 50% от изходния код с функционални тестове;
- Използване на continuous integration практики;
- Използване на dependency management.

Участникът трябва да опише детайлно подхода си за покриване на изискванията.

Във всеки един компонент на Системата, който се build-ва и подготвя за инсталация (deployment) е необходимо да присъстват следните реквизити:

1. Дата и час на build;
2. Място/среда на build;
3. Потребител извършил/стартирал build процеса;
4. Идентификатор на ревизията от кодовото хранилище на компонента, срещу която се извършва build-a.

7.2.6. Бърздействие и мащабируемост

7.2.6.1. Контрол на натоварването и защита от DoS / DDoS атаки

Неприложимо.

7.2.6.2. Кохерентно кеширане на данни и заявки

Неприложимо.

7.2.6.3. Бърздействие

- При визуализация на уеб-страници, системите трябва да осигуряват висока производителност и минимално време за отговор на заявки – средното време за заявка под 1 секунда, с максимум 1 секунда стандартно отклонение;
- Трябва да бъдат създадени тестове за натоварване.

7.2.6.4. Използване на HTTP/2

С оглед намаляване на служебния трафик, времената за отговор и натоварването на сървърите следва да се имплементира HTTP/2 протокол с включени следните възможности:

- Включена header compression;
- Използване на brotli алгоритъм за компресия;
- Включен HTTP pipelining;
- HTTP/2 Server push, приоритизиращ специфични компоненти, изграждащи страниците (CSS, JavaScript файлове и др.);

- Публичните интерфейси (потребителски и API) трябва да поддържат адаптивни TLS cipher suites според вида на процесорната архитектура на клиентското устройство – AES-GCM за x86 работни станции и преносими компютри (с налични AES-NI CPU разширения) и ChaCha20/Poly1305 за мобилни устройства (основно базирани на ARM процесори);

- Ако клиентският браузър/клиент не поддържа HTTP/2, трябва да бъде предвиден fall-back механизъм към HTTP/1.1. Тази възможност трябва да може да се реконфигурира в бъдеще и да отпадне, когато браузърите/клиентите, неподдържащи HTTP/2 станат незначителен процент.

7.2.6.5. Подписване на документи

Неприложимо.

7.2.6.6. Качество и сигурност на програмните продукти и приложенията

- Да бъде предвидено спазването на добри практики на софтуерната разработка – покритие на изходния код с тестове – над 60%, документиране на изходния код, използване на среда за непрекъсната интеграция (Continuous Integration), система за управление на зависимостите (Dependency Management);

7.2.7. Информационна сигурност и интегритет на данните

- Не се допуска съхранението на пароли на администратори, на вътрешни и външни потребители и на акаунти за достъп на системи (ако такива се използват) в явен вид. Всички пароли на трябва да бъдат защитени с подходящи сигурни алгоритми (напр. bcrypt, PBKDF2, scrypt (RFC 7914)) за съхранение на пароли и където е възможно, да се използва и прозрачно криптиране на данните в СУБД със сертификати (transparent data-at-rest encryption);

- Да бъде предвидена система за ежедневно създаване на резервни копия на данните, които да се съхраняват извън инфраструктурата на системата;

- Всички уеб-страници трябва да бъдат достъпни единствено и само през протокол HTTPS. Криптирането трябва да се базира на сигурен сертификат с валидирана идентичност (Verified Identity), позволяващ задължително прилагане на TLS 1.2, който е издаден от удостоверяващ орган, разпознаван от най-често използваните браузъри (Microsoft Internet Explorer, Google Chrome, Mozilla Firefox). Ежегодното преиздаване и подновяване на сертификата трябва да бъде включено като разходи и дейности в гаранционната поддръжка за целия срок на поддръжката;

- При разгръщането на всички уеб-услуги (Web Services) трябва да се използва единствено протокол HTTPS със задължително прилагане на минимум TLS 1.2;

- Програмният код трябва да включва методи за автоматична санитизация на въвежданите данни и потребителски действия за защита от злонамерени атаки, като минимум SQL инжекции, XSS атаки и други познати методи за атаки, и да отговаря, където е необходимо, на Наредбата за оперативна съвместимост и информационна сигурност;

- При проектирането и разработката на компонентите на Системата и при подготовката и разгръщането на средите, трябва да се спазват последните актуални препоръки на OWASP (Open Web Application Security Project);

- Трябва да бъде изграден модул за проследимост на действия и събития в Системата. За всяко действие (добавяне, изтриване, модификация, четене) трябва да съдържа следните атрибути:

- Уникален номер;
- Точно време на възникване на събитието;
- Вид (номенклатура от идентификатори за вид събитие);
- Данни за информационна система, където е възникнало събитието;
- Име или идентификатор на компонент в информационната система, регистрирал събитието;
- Приоритет;
- Описание на събитието;
- Данни за събитието.

- Астрономическото време за удостоверяване настъпването на факти с правно или техническо значение се отчита с точност до година, дата, час, минута, секунда и милисекунда, изписани в съответствие със стандарта БДС ISO 8601:2006;

- Астрономическото време за удостоверяване настъпването на факти с правно значение и такива, за които се изисква противопоставимост, трябва да бъде удостоверявано с електронен времеви печат по смисъла на Раздел 6 от Регламент ЕС 910/2014. Трябва да бъде реализирана функционалност за получаване на точно астрономическо време, отговарящо на горните условия, и от доставчик на доверителни услуги или държавен орган, осигуряващ такава услуга отговаряща на изискванията на RFC 3161;

7.2.8. Използваемост

7.2.8.1. Общи изисквания за използваемост и достъпност

Изискванията за използваемост и достъпност към Системата за електронен обмен на социално-осигурителна информация на НЗОК се отнасят само към разработените от Изпълнителя модули.

- Всички ресурси трябва да са достъпни чрез GET заявка на уникален адрес (URL). Не се допуска използване на POST за достигане до формуляр за подаване на заявление, за генериране на справка и други.

- Функционалностите на потребителския интерфейс на Системата трябва да бъдат независими от използваните от потребителите интернет браузъри и устройства, при условие, че последните са версии в период на поддръжка от съответните производители.

- В екранните форми на Системата трябва да се използват потребителски бутони с унифициран размер и лесни за разбиране текстове в еднакъв стил.

- Всички текстови елементи от потребителския интерфейс трябва да бъдат визуализирани със шрифтове, които са подходящи за изобразяване на екран и които осигуряват максимална съвместимост и еднакво възпроизвеждане под различни клиентски операционни системи и браузъри. Не се допуска използването на серифни шрифтове (Serif).

- Полета, опции от менюта и командни бутони, които не са разрешени конкретно за ролята на влезлия в системата потребител, не трябва да са достъпни за този потребител. Това не отменя необходимостта от ограничаване на достъпа до бизнес логиката на приложението чрез декларативен или програмен подход.

- Всяка екранна форма трябва да има наименование, което да се изписва в горната част на екранната форма. Наименованията трябва да подсказват на потребителя какво е предназначението на формата.

- Всички търсения трябва да са нечувствителни към малки и главни букви;

- Полетата за пароли трябва задължително да различават малки и главни букви;

- Полетата за потребителски имена трябва да позволяват използване на и-мейл адреси като потребителско име, включително и да допускат всички символи регламентирани в RFC 1123 за наименоването на хостове;

- Главните и малки букви на въвежданите данни се запазват непроменени, не се допуска Системата да променя капитализацията на данните въведени от потребителите.

- Системата трябва да позволява въвеждане на данни, съдържащи както български, така и символи на официалните езици в ЕС.

- Наименованията на полетата следва да са достатъчно описателни, като максимално се доближават до характера на съдържащите се в тях данни.

- Системата трябва да поддържа прекъсване на потребителски сесии при липса на активност. Времето трябва да може да се променя от администратора на системата без

промяна в изходния код. Настройките за време за прекъсване на неактивни сесии трябва да включват и възможността администраторите да дефинират стилизирана страница със информативно съобщение, към която Системата да пренасочва автоматично браузърите на потребителите, в случай на прекъсната сесия.

- Дългите списъци с резултати трябва да се разделят на номерирани страници с подходящи навигационни елементи за преминаване към предишна, следваща, първа и последна страница, конкретна страница. Навигационните елементи трябва да са логически обособени и свързани със съответния списък и да се визуализират в началото и края на HTML контейнера съдържащ списъка.

- За големите йерархически категоризации трябва да се предвиди възможност за навигация по нива или чрез отложено зареждане (lazy load).

7.2.8.2. Интернационализация

- Системата трябва да може да съхранява и едновременно да визуализира данни и съдържание, което е въведено/генерирано на различни езици;

- Всички софтуерни компоненти на Системата, използваните софтуерни библиотеки и развойни комплекти, приложните сървъри и сървърите за управление на бази данни, елементите от потребителския интерфейс, програмно-приложните интерфейси, уеб-услугите и пр. трябва да поддържат стандартно и да са конфигурирани изрично за спазване на минимум Unicode 5.2 стандарт при съхранението и обработката на текстови данни, респективно трябва да се използва само UTF-8 кодиране на текстовите данни.

7.2.8.3. Изисквания за използваемост на потребителския интерфейс

- Електронните форми за подаване на заявления и за обявяване на обстоятелства трябва да бъдат реализирани с AJAX или аналогична технология, като по този начин се гарантират следните функционалности:

- Контекстна валидация на въвежданите данни на ниво "поле" от форма и контекстни съобщения за грешка / невалидни данни в реално време;
- Възможност за избор на стойности от номенклатури чрез търсене в списък по част от дума (autocomplete) и визуализиране на записи, отговарящи на въведеното до момента, без да е необходимо пълните номенклатури да са заредени в браузъра на клиента и потребителят да скорлира дълги списъци с повече от 10 стойности;

- В електронните форми трябва да бъде реализирана валидация на въвежданите от потребителите данни на ниво "поле" (in-line validation). Валидацията трябва да се извършва в реално време на сървъра, като при успешна валидация, данните от съответното поле следва да бъдат запазени от сървъра;

- Системата трябва да гарантира, че въведени, валидирани и запазени от сървъра данни, остават достъпни за потребителите, дори за процеси, които не са приключили, така че при волно, неволно или автоматично прекъсване на потребителската сесия поради изтичане на периода за допустима липсва на активност, потребителят да може да продължи съответния процес след повторно влизане в системата, без да загуби въведените до момента данни и прикачените до момента електронни документи;

- Трябва да бъде реализирана възможност за добавяне и редактиране от страна на администраторите на системата, без да са необходими промени в изходния код, на контекстна помощна информация за:

- всяка електронна форма или стъпка от процес, за която има отделен екран / форма;
- всяка група полета за въвеждане на данни (в случаите, в които определени полета от формата са групирани тематично);
- всяко отделно поле за въвеждане на данни;

- Трябва да бъде разработена контекстна помощна информация за всички процеси, екрани и електронни форми, включително ясни указания за попълване и разяснения за особеностите при попълване на различните групи полета или отделни полета;

- Контекстната помощна информация, указанията към потребителите и информативните текстове за всяка електронна административна услуга не трябва да съдържат акроними, имена и референции към нормативни документи, които са въведени като обикновен текст (plain-text). Всички акроними, референции към нормативни документи, формуляри, изисквания и пр. трябва да бъдат разработени като хипер-връзки към съответните актуални версии на нормативни документи и/или съответния речник / списък с акроними и термини;

- Достъпът на потребителя до контекстната помощна информация трябва да бъде реализиран по унифициран и консистентен начин, чрез подходящи навигационни елементи, като например чрез подходящо разположени микро-бутони с икони разположени до/преди/след етикета на съответния елемент, за който се отнася контекстната помощ или чрез обработка на "Mouse Hover / Mouse Over" събития;

- Потребителският интерфейс следва да бъде достъпен за хора с увреждания, съгласно изискванията на чл. 48, ал. 5 от ЗОП.

7.2.8.4. Изисквания за използваемост в случай на прекъснати бизнес процеси

- Системата трябва да съхранява перманентно всеки започнал процес / процедура по подаване на заявление или обявяване на обстоятелства, текущия му статус, всички

въведени данни и прикачени документи, дори ако потребителят е прекъснал волно или неволно потребителската си сесия;

- При вход в системата потребителят трябва да получава прегледна и ясна нотификация, че има започнати, но недовършени / неизпратени / неподписани заявления и да бъде подканен да отвори модула за преглед на историята на транзакциите;

- Модулът за преглед на историята на транзакциите трябва да поддържа следните функционалности:

- Да визуализира списък с историята на подадените заявления, като минимум със следните колони – дата, входящ номер, код на тупа формуляр, подател, статус на заявлението;
- Да предлага видни и лесни за използване от потребителите контроли / инструменти:
 - за филтриране на списъка (от дата до дата, за предефинирани периоди като "последния 1 месец", "последната 1 година");
 - сортиране на списъка по всяка от колоните, без това да премахва текущия филтър;
 - свободно търсене по ключови думи по всички колони в списъка и метаданните на прикачените / свързани документи със заявленията, което да води до динамично филтриране на списъка.

7.2.8.5. Изисквания за проактивно информиране на потребителите
Неприложимо.

7.2.9. Системен журнал

Изгражданото решение задължително трябва да осигурява проследимост на действията на всеки потребител (одит), както и версия на предишното състояние на данните, които той е променил в резултат на своите действия (системен журнал).

Атрибутите, които трябва да се запазват при всеки запис трябва да включват като минимум следните данни:

- дата/час на действието;
- модул на системата, в който се извършва действието;
- действие;
- обект, над който е извършено действието;
- допълнителна информация;
- IP адрес и браузър на потребителя.

Размерът на журнала на потребителските действия нараства по време на работа на всяка система, което налага по-различното му третиране от гледна точка организация на базата данни:

- по време на работа на системата потребителският журнал трябва да се записва в специализиран компонент, който поддържа много бързо добавяне на записи; този подход се налага, за да не се забавя излишно работата на Системата;

- специална фоновая задача трябва да акумулира записаните данни и да ги организира в отделна специално предвидена за целта база данни, отделна от работната база данни на системата;

- данните в специализираната база данни трябва да се архивират и изчистват, като в специализираната база данни трябва да бъде достъпна информация за не повече от 2 месеца назад; при необходимост от информация за предишен период администраторът на системата трябва първо да възстанови архивните данни.

7.2.10. Дизайн на бази данни и взаимодействие с тях

При използване на база данни (релационна или нерелационна(NoSQL) следва да бъдат следвани добрите практики за дизайн и взаимодействие с базата данни, в т.ч.:

- дизайнът на схемата на базата данни (ако има такава) трябва да бъде с максимално ниво на нормализация, освен ако това не би навредило сериозно на производителността;

- базата данни трябва да може да оперира в клъстер; в определени случаи следва да бъде използван т.нар. sharding;

- имената на таблиците и колоните трябва да следват унифицирана конвенция;

- трябва да бъдат създадени индекси по определени колони, така че да се оптимизират най-често използваните заявки; създаването на индекс трябва да е мотивирано и подкрепено със замервания;

- връзките между таблици трябва да са дефинирани чрез foreign key;

- периодично трябва да бъде правен анализ на заявките, включително чрез EXPLAIN (при SQL бази данни), и да бъдат предприети мерки за оптимизиране на бавните такива;

- задължително трябва да се използват транзакции, като нивото на изолация трябва да бъде мотивирано в предадената документация;

- при операции върху много записи (batch) следва да се избягват дългопродължаващи транзакции;

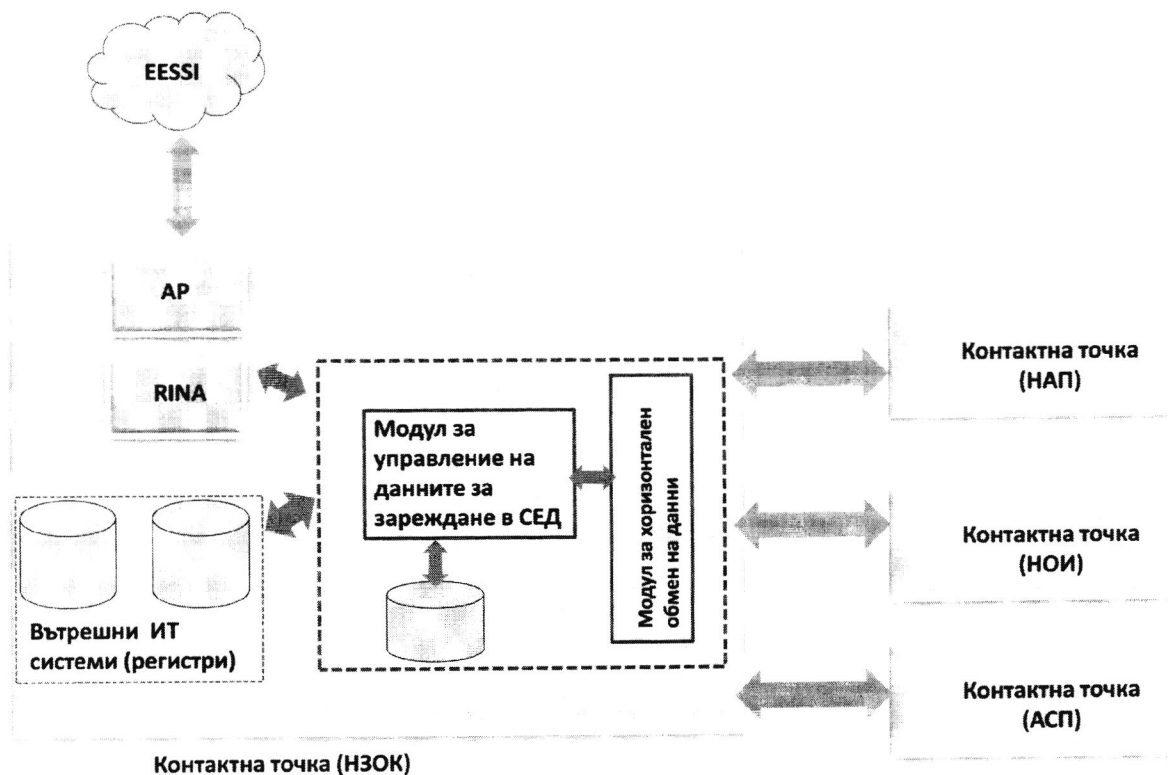
- заявките трябва да бъдат ограничени в броя записи, които връщат;

- при използване на ORM или на друг слой на абстракция между приложението и базата данни, трябва да се минимизира броят на излишните заявки (т.нар. n+1 selects проблем);

- при използване на нерелационна база данни трябва да се използват по-бързи и компактни протоколи за комуникация, ако такива са достъпни.

8. ИЗИСКВАНИЯ КЪМ ИЗПЪЛНЕНИЕТО НА ДЕЙНОСТИТЕ ПО ПРОЕКТА

Следващата фигура показва общата архитектура на бъдещата Система за електронен обмен на социално-осигурителна информация на НЗОК.



8.1. Дейност 1 Внедряване на национална точка за достъп (AP) и RINA в НЗОК

8.1.1. Описание на дейността

Поддейност 1.1. Инсталиране на софтуера на точката за достъп (AP)

Инсталиране на софтуера на AP според спецификациите и препоръките на Европейската комисия и интеграция в инфраструктурата на НЗОК. Изпълнителят трябва

да инсталира софтуера AP, който е свързан с продукционна, тестова и развойна среда на RINA.

Софтуерът се осигурява от Европейската комисия и се предоставя от Възложителя, когато е наличен.

Изпълнителят трябва да извърши конфигуриране на софтуера на AP. Софтуерът трябва може да работи с RINA, другите точки за достъп и ако е необходимо, с централна точка за обмен.

Поддейност 1.2. Инсталиране на RINA

В поддейност 1.2. Изпълнителят трябва да извърши:

- Анализ на работните процеси и ИТ инфраструктурата на НЗОК с оглед инсталирането на RINA.
- Инсталиране на RINA според спецификациите и препоръките на Европейската комисия. Изпълнителят трябва да инсталира RINA в продукционна, тестова и развойна среда при използване на специфичната функционалност, описана в EESSI Architecture pack. В същото време поради ролята на НЗОК като точка за достъп, някои от компетентните институции, свързани към точката за достъп следва да могат да използват функцията за RINA портал. При такива сценарии трябва да се реализират възможности за множество инсталации на RINA.
- Конфигуриране на RINA за работа със системите на НЗОК и подготовка за свързване с точката за достъп AP с цел извършване на обмен по EESSI. RINA следва да бъде интегрирана със софтуерните решения по Дейност 2. Разработване на модули за комуникация и обмен на данни между националните контактни точки и Дейност 3: Разработване на модул за управление на данните за зареждане в СЕД.

Софтуерът RINA се разработва и поддържа от Европейската комисия и се предоставя от Възложителя, когато е налична негова продукционна версия.

8.1.2. Изисквания към изпълнение на дейността

Изпълнението на Дейност 1 трябва да се координира с изпълнението на Дейност 2 и Дейност 3.

Възложителят ще предостави на изпълнителя необходимите софтуерни пакети за извършване на инсталация и конфигурация на софтуер на точката за достъп и RINA.

8.1.3. Очаквани резултати

- Инсталиран софтуер на точката за достъп (НЗОК).

- Инсталирана и конфигурирана RINA при използване на специфичната функционалност, описана в EESSI Architecture pack.

8.2. Дейност 2 Разработване на модули за комуникация и обмен на данни между националните контактни точки за достъп и НЗОК

8.2.1. Описание на дейността

Поддейност 2.1. Подготовка на данните и информационните системи в НЗОК за обмена между точките за достъп.

Поддейност 2.1. включва:

- анализ на информацията, която е необходимо да се зарежда в RINA за целите на формиране на документите по EESSI от компетентността на НЗОК;
- анализ на информацията и информационните системи в НЗОК, от които е необходимо да се извличат данни за целите на предоставянето им на останалите точки за достъп и информацията, която е необходимо да се получават от останалите точки за достъп за целите на използването им от НЗОК при формиране на документите по EESSI от компетентността на НЗОК;
- изготвяне на спецификация на данните, необходими за обмена между НЗОК и контактните точки.

Поддейност 2.2. Разработване и внедряване на уеб услуги.

Поддейността включва Разработване и внедряване на на уеб услуги за комуникация и обмен на данни между НЗОК и националните точки за достъп чрез разработеното софтуерно решение по Дейност 3. Не се предполага директен достъп до данните в ИТ системите на НЗОК за трети страни. Изпълнителят трябва да направи анализ на нуждите и да предложи на Възложителя какви типове данни е необходимо да се обменят данни между НЗОК и националните точки за достъп.

Изготвяне на справочна част за проследяване на статусите и готовността на данните за формиране на документите по EESSI от компетентността на НЗОК.

Следва да се извърши зареждане на данни в RINA и за целите на формирането на документите по EESSI от компетентността на НЗОК.

Необходимо е да се извърши 3-дневно обучение за 5 (петима) ИТ специалисти на НЗОК за инсталацията, конфигурирането и администрирането на уеб услугите за обмен на данни между националните точки за достъп.

8.2.2. Изисквания към изпълнение на дейността

Изпълнението на Дейност 2 е в пряка зависимост от изпълнението на Дейност 3, тъй като обменът на данни ще се осъществява чрез разработеното решение по Дейност 3.

Изпълнителят е длъжен да спазва изискванията на ЕК за обмен на социално-осигурителна информация, които са валидни към периода на изпълнение на проекта.

8.2.3. Очаквани резултати

- Извършена подготовка на данните и информационните системи в НЗОК за обмена между точките за достъп.
- Разработени и внедрени веб услуги за обмен на данни между НЗОК и националните точки за достъп.
- Проведено 3-дневно обучение за 5 ИТ специалисти на НЗОК.

8.3. Дейност 3 Разработване на модул за управление на данните за зареждане в СЕД

8.3.1. Описание на дейността

В рамките на тази дейност ще бъде разработено софтуерно решение, което да автоматизира процесите по работа с ВУС/СЕД в НЗОК.

ЦУ на НЗОК и РЗОК обработват следните ВУС и съставните им структурирани електронни документи:

ВУС - НЗОК	SED-ове от ВУСs	Контактна точка, която участва в обработването на всеки SED
AW_BUC_01a - Certification right benefit in Kind – Member of stay/residence request confirmation of rights	DA001, DA002, DA003	НЗОК/РЗОК
AW_BUC_01b - Certification of the right for benefits in kind - Competent Member State sends the certification	DA002, DA003	НЗОК/РЗОК
AW_BUC_02 - Cancellation Certification right benefit in Kind	DA003	НЗОК/РЗОК
AW_BUC_03 - Request for Reimbursement rates	DA004, DA005	ЦУ НЗОК
AW_BUC_04a - AWOD benefits in kind - Authorise Scheduled Treatment	DA006, DA007	ЦУ НЗОК

BUC - НЗОК	SED-ове от BUCs	Контактна точка, която участва в обработването на всеки SED
AW_BUC_04b - AWOD benefits in kind - Information about supplement scheduled treatment	DA007, DA008	ЦУ НЗОК
AW_BUC_04c - AWOD benefits in kind - Information about vitally necessary treatment	DA009	ЦУ НЗОК
AW_BUC_05 - Claim for Reimbursement of Benefit in Kind	DA010, DA011, DA012, DA014, DA015, DA016, DA017, DA018, DA019	ЦУ НЗОК
AW_BUC_07c - Benefits in kind - Detailed report	DA042 - DA044	ЦУ НЗОК
AW_BUC_08 - Contestation of Application of Legislation	DA048, DA049	ЦУ НЗОК/ПЗОК
AW_BUC_14 - Cost of Transport	DA062, DA063	ЦУ НЗОК
AW_BUC_15 - Claim Interest Reimbursement of Benefits in Kind	DA020, DA021, DA022, DA024, DA025, DA026	ЦУ НЗОК
AW_BUC_23 - Refund of Overpayment	DA071, DA073, DA074	ЦУ НЗОК
Ad_BUC_01_Subprocess - Close Case	X001	ЦУ НЗОК/ПЗОК
Ad_BUC_02_Subprocess - Reopen Case	X002, X003, X004	ЦУ НЗОК/ПЗОК
Ad_BUC_03_Subprocess - Add Participant	X005	ЦУ НЗОК/ПЗОК
Ad_BUC_04_Subprocess - Remove Participant	X006	ЦУ НЗОК/ПЗОК
Ad_BUC_05_Subprocess - Forward Case	X007	ЦУ НЗОК/ПЗОК
Ad_BUC_06_Subprocess - Invalidate SED	X008	ЦУ НЗОК/ПЗОК
Ad_BUC_07_Subprocess - Reminder	X009, X010	ЦУ НЗОК/ПЗОК
Ad_BUC_08_Subprocess - Clarify SED Content	X012, X013	ЦУ НЗОК/ПЗОК

BUC - НЗОК	SED-ове от BUCs	Контактна точка, която участва в обработването на всеки SED
Ad_BUC_09_Subprocess - Reject SED	X011	ЦУ НЗОК/ПЗОК
Ad_BUC_10_Subprocess - Update SED	-	ЦУ НЗОК/ПЗОК
Ad_BUC_11_Subprocess - Business Exception	X050	ЦУ НЗОК/ПЗОК
Ad_BUC_12_Subprocess - Change of Participant	X100	ЦУ НЗОК/ПЗОК
H_BUC_01 - Ad-hoc Exchange of Information	H001, H002	ЦУ НЗОК/ПЗОК
H_BUC_02a - Determine Residence, Request for information	H005, H006	НЗОК/ПЗОК
H_BUC_02b - Determine Residence, Disagreement procedure	H003, H004, H005, H006	НЗОК/ПЗОК
H_BUC_02c - Determine Residence, Notification on Residence	H003	НЗОК/ПЗОК
H_BUC_04 - Reimbursement of Admin Check or Med Costs	H020, H021	ЦУ НЗОК
H_BUC_05 - Exchange of PIN	H061, H062	ЦУ НЗОК/ПЗОК
H_BUC_06 - Exchange of Claim	H065, H066	ЦУ НЗОК/ПЗОК
H_BUC_08 - Exchange of Medical Data	H120, H121	ЦУ НЗОК
H_BUC_09 - Notification of Medical Report	H121	ЦУ НЗОК
H_BUC_10 - Request for Administrative Check	H130, H131	ЦУ НЗОК
H_BUC_01_Subprocess - Ad-hoc Exchange of Information	H001, H002	ЦУ НЗОК/ПЗОК
H_BUC_02_Subprocess - Determine Residence, Disagreement, Procedure, Notification of Residence, Request for Information	H003, H004 H005, H006	НЗОК/ПЗОК
H_BUC_04_Subprocess - Reimbursement of Administrative Check or Medical examination	H020, H021	ЦУ НЗОК