

# Mittefunktsionaalsed nõuded arendustele

<p>Versioon: 2.1</p>
<p>Dokument määrab kvaliteedi- ja mittefunktsionaalsed nõuded (MFN) uutele infosüsteemidele ning nende dokumentatsioonile. Dokumenti hoiavad ajakohasena Tervise ja Heaolu Infosüsteemide Keskuse (TEHIK) arhitektid.</p>
<p>Dokumenti tuleb vaadata kui arenduste kvaliteedi- ja mittefunktsionaalsete nõuete põhidokumenti. Põhidokumendi ja viidatud dokumentide erisuste puhul tuleb lähtuda põhidokumendis kirjeldatust. Põhidokumendis viidatud TEHIKu koostatud dokumentide ja kolmandate osapoolte koostatud dokumentide erisuste puhul tuleb lähtuda TEHIKu dokumentides kirjeldatust.</p>
<p>Kui mõnda nõuet ei ole võimalik või otstarbekas täita, tuleb selle mittetäitmise fakt ja põhjendus välja tuua pakkumuse esitamisel.</p>
<p>Nõudeid tuleb järgida ka olemasolevate infosüsteemide versiooniuuendustel nii palju kui versiooniuuenduse käigus võimalik.</p>
<p>Erandid tuleb kooskõlastada TEHIKu vastutava arhitektiga.</p>

Nõude nr	Nõude sisu	Seletused	Koostamise eest vastutaja	Testimise läbi viib või kinnitab
<b>1. Vastavus üldistele standarditele</b>				
1.1	Lahenduse X-tee teenused peavad vastama RIA nõuetele	<a href="https://www.ria.ee/et/ametist/juhendid.html">https://www.ria.ee/et/ametist/juhendid.html</a>	Arendaja	Testija
1.2	Lahendus peab vastama Sotsiaalministeeriumi IT-profiilile	IT-Profiil	Arendaja	Projektijuht Arhitekt Administraator Testija Turvatestija Infoturbe spetsialist Standardija
1.3	Rakendus peab olema kirjutatud arvestades selle rakenduse poolt töödeldavatele andmetele määratud ISKE turvaklassi nõudeid	<a href="https://www.ria.ee/et/kuberturvalisus/infosusteemide-turvameetmete-susteem-iske.html">https://www.ria.ee/et/kuberturvalisus/infosusteemide-turvameetmete-susteem-iske.html</a>	Arendaja	Turvatestija
1.4	Veebirakenduse kasutajaliides peab vastama vähemalt WCAG 2.1 tasemele AA	<a href="https://www.w3.org/TR/WCAG21/">https://www.w3.org/TR/WCAG21/</a>	Arendaja	Testija
1.5	Veebipõhine kasutajaliides peab ühilduma täielikult standarditega HTML 5 ja CSS 3.	Valideerimiseks kasutatakse vastavaid validaatoreid: <a href="http://validator.w3.org/">http://validator.w3.org/</a> Kui on tegu olemasoleva süsteemi edasiarendusega, siis tuleb järgida olemas olevat HTML ja CSS versiooni.	Arendaja	Testija
1.6	Allkirjastamisel tuleb kasutada TEHIKu SiGA/SiVa vahendusteenust.	Vaata ka <a href="https://www.tehik.ee/arendusjuhendid">https://www.tehik.ee/arendusjuhendid</a>	Arendaja	Testija
1.7	Veebirakendus peab probleemiteta läbima OWASP ASVS baasil põhineva testi	Kui pole arenduse eraldi kokku lepitud teisiti, siis on OWASP ASVS tasemeks 2 ( <a href="https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project">https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project</a> ). Kinnise lähtekoodiga komertstoote kasutamisel ei eeldata ligipääsu kinnisele lähtekoodile. Tellija poolset turvatestimist teostab kolmas sõltumatu osapool. Selline esmane kolmanda osapoolte turvatestimine tellitakse tellija finantseeringul. Ilmnunud vigade korral ja peale nende parandamist peab järeletestimise rahaliselt kompenseerima arendaja, kui tellija vastava nõudmise esitab.	Arendaja	Turvatestija
1.8	Krüptoalgoritmide ja räsifunktsioonide kasutamisel tuleb kasutada turvalisi algoritme ja võtmepikkuseid.	Krüptoalgoritmide ja räsifunktsioonide kasutamisel tuleb järgida uusimat RIA kodulehel avaldatud krüptograafiliste algoritmide kasutusvaldkondade ja elutsükli uuringut. Värskema uuringu leiab aadressilt <a href="https://www.ria.ee/et/ametist/uuringud-analuusid-ulevaated.html">https://www.ria.ee/et/ametist/uuringud-analuusid-ulevaated.html</a>  Arendaja loodud lahenduse dokumentatsioon (nt detailanalüüs vms) tuleb välja tuua kasutatavad krüpto- ja räsialgoritmide, nende võtmepikkused, kasutuskohad, sh sertifikaatide kasutuskohad.	Arendaja	Arhitekt Administraator Turvatestija
1.9	Andmete edastus peab olema kaitstud kasutades krüpteeritud ja vajadusel autenditud kanalit.	Autentimist ei ole vaja ainult avalikke andmete edastamisel. Näiteks avaandmed.  Autentimise mehhanism tuleb kokku leppida TEHIKu poolse arhitektiga.	Arendaja	Turvatestija
1.10	Infosüsteem peab kasutama serveri kellaega.	Kõik mahakirjutatavad ja talletatavad kellaajad tuleb salvestada UTC ajatsoonis koos ajatsooni infoga.  Kasutajatele mõeldud kuvades tuleb kasutada sirviku ajatsooni. Aja esitamisel tekstikujul lähtuda standardist ISO 8601.	Arendaja	Administraator
1.11	Süsteemi edasiarendamisel/loomisel peab arvestama selle võimaliku laiendamisega nii andmemahtude, kui ka kasutajate arvu osas.		Arendaja	Arhitekt Testija

1.12	Rakendus peab olema tehniliselt tükeldatud vastavalt loogilisele jaotusele. Saadud osised peavad olema eraldi versiooneeritavad ja paigaldatavad	Lahenduse arhitektuuris kasutada domeenist juhitud disaini ja mikroteenuste põhimõtteid.  Näiteks, kui rakendus on eraldi turvakontekstidega liidese ametnikule ja kodanikule, peab rakendus olema jagatav kaheks eraldi liidese komponendiks ning nende mõlema poolt kasutatavaks andmebaasiks.	Arendaja	Arhitekt
1.13	Avalike e-teenuste loomisel peab arvestama valitsusastutele kehtestatud visuaalse identiteedi stiiljuhiseid	<a href="https://riigikantselei.ee/valitsuse-too-planeerimine-ja-korraldamine/valitsuskommunikatsioon/visuaalne-identiteet">https://riigikantselei.ee/valitsuse-too-planeerimine-ja-korraldamine/valitsuskommunikatsioon/visuaalne-identiteet</a>	Arendaja	Testija
1.14	Avalike e-teenuste loomisel peab arvestama Veera disainisüsteemiga.	<a href="https://zeroheight.com/3d136290e/p/188910-veera-disainisysteem/b/94293b">https://zeroheight.com/3d136290e/p/188910-veera-disainisysteem/b/94293b</a>	Arendaja	Arhitekt
<b>2. Nõuded rakenduse arhitektuurile</b>				
2.1	Rakenduse, andmebaasi ja kolmanda osapoole komponendid peavad olema sellised, mille eluea lõpp (EOL) pole teadaolevalt vähem kui 2 aasta pärast.	Arendaja loodud lahenduse dokumentatsioon (nt analüüs vms) peab olema välja toodud kasutatavate komponentide nimetused ja versioonid. Versiooni eluea lõppu ei loeta võrdseks terve komponendi eluea lõpuks, st versiooni tugi võib aeguda, kui uus versioon on välja lastud.  Jätkaarenduse puhul tuleb kaardistada eelneva arendusperioodi komponentide kaardistus. EOL komponentide kasutamisest tuleb teavitada TEHKu arhitekti.	Arendaja	Arhitekt
2.2	Tulevase ja olemasolevate infosüsteemide platvormid (rakendusserver, andmebaas, kolmanda osapoole komponendid) ja topoloogia peab olema loodud kooskõlas IT profiiliga.	Süsteemi jõudlus peab vastama kokkulepitud topoloogial eelanalüüsi ja lähteülesande käigus välja toodud jõudlusnäitajatele.	Arendaja	Arhitekt
2.3	Rakendusserver peab võimaldama töötamist andmebaasserverist eraldi serveril.		Arendaja	Administraator
2.4	Rakendusserver peab olema vajadusel klasterdatav aktiivklasteris.	Lahendused peavad toetama rakenduste horisontaalset skaleeruvust.  Kasutaja sessioonid ei tohi olla rakenduserveri klasteri õla põhised.	Arendaja	Administraator
2.5	Rakendust peab saama ilma ümberprogrammeerimata liigutada erinevate domeenide ja domeeni saitide vahel.	Lahendus ei tohi olla sisse kompileeritud absoluutseid URI-sid	Arendaja	Administraator
2.6	Rakenduse komponentide konfiguratsiooni peab olema võimalik ette anda käivitamisel. Konfiguratsiooni muudatus peab olema teostatav ilma rakendust kompilleerimata.	Rakendus peab neid sealt ka kasutama (mitte kopeerima parameetreid käivitamisel kolmandatesse kohtadesse), logimise seaded võivad olla rakenduse konfiguratsioonifailist eraldi ühes lisakonfiguratsioonifailis (näit Log4net). Samuti on väga soovitatav eraldi konfiguratsioonifailis hoida arendaja ja administraatori vastutusala parameetrid. Infosüsteem peab olema seadistatav konfiguratsiooniparameetrite(de) abil. Konfiguratsioonifailiks ei saa lugeda faili, kus hoitakse lisaks konfiguratsioonile ka muud programmikoodi.  Näiteks dockeri konteinerite puhul keskkonnamuutujad.	Arendaja	Administraator Testija
2.7	Rakenduse taaskäivitus, konfiguratsiooni muutmine vms peab toimuma mõistliku aja jooksul.	Tavaline käivitusae ei tohi ületada 1 minutit.  Kui rakendus vajab indekseeritud sisu ja see pole kättesaadav, siis peab rakendus väljastama selle kohta selge teate.	Arendaja	Testija
2.8	Lahenduse väliste osapoole komponentide konfiguratsioonid peavad olema puhverdatud.	Kui rakendusel või mõnel selle komponendil on tihti kasutatav teenus ning sellel teenusel on laetav konfiguratsioon, siis tuleb:  <ul style="list-style-type: none"> <li>• laadida konfiguratsioon ühekordselt ja seda korduvkasutada;</li> <li>• konfiguratsiooni automaatselt ja regulaarselt värskendada; regulaarsuse tarbeks peab saama määra intervalli, millise aja järel või täpsed kellaajad, millal konfiguratsiooni värskendatakse;</li> <li>• luua võimalus värskendada konfiguratsiooni käsitsi.</li> </ul> Näiteks: <ul style="list-style-type: none"> <li>• digidoc4j teegi korral laetakse TSL nimekirja välisvõrgust puhvrise, et vähendada koormust kolmandale osapoolele.</li> <li>• OpenID konfiguratsioon</li> </ul>	Arendaja	Arhitekt Testija Administraator
2.9	Rakendus peab kasutama 64-bitist arvutarhitektuuri.		Arendaja	Administraator Arhitekt
2.10	Kõik andmed, andmebaasid, SQL skriptid, lähtekood ja rakendus peavad kasutama UTF-8 kodeeringut.		Arendaja	Administraator Testija
2.11	Rakenduserveri failisüsteemi ei tohi salvestada midagi püsivaks kasutamiseks.	Näiteks objektide talletuseks kasutada S3 lahendust.	Arendaja	Administraator
2.12	Ühest relatsioonilise andmebaasi andmetabelist teise viitamisel tuleb kasutada väliseid võtmeid (Foreign key).	Erinevate skeemide vahelised ühendused on keelatud. Peab kasutama REST/SOAP/AMQP liidestust.	Arendaja	Arhitekt
2.13	Kõik välised võtmed (Foreign Key) peavad olema indekseeritud.	Andmebaasid peab kasutama indekseid või muid meetmeid, et nõuded rakenduse jõudlusele oleksid täidetud ka tulevikus. (1, 3, 5 või 10 aasta pärast – vastavalt planeeritud kasutusajale).	Arendaja	Arhitekt
2.14	Tuleb kasutada päringumuutujaid (Parameter Binding)	SQL päringute väljakutsumisel väljastpoolt andmebaasi, peab kasutama päringumuutujaid, et vältida SQL vahemälu fragmentseerumist (When calling SQL code from outside the database, Parameter Binding should be used to prevent SQL cache fragmentation)	Arendaja	Arhitekt
2.15	Kõigis andmebaasi tabelites peab olema defineeritud üks primaarvõti.	Kasutada tuleb vastava andmebaasisüsteemi nimetamise parimaid praktikaid.	Arendaja	Arhitekt

2.16	Andmebaasi objektide nimetused peavad olema sisulised ja andma aimu nende ostarbest.	Kasutada tuleb vastava andmebaasisüsteemi nimetamise parimaid praktikaid.	Arendaja	Arhitekt
2.17	Andmebaasis defineeritakse üldjuhul kaks või enam kasutajat: <ul style="list-style-type: none"> <li>Rakenduse peakasutaja, kellena luuakse objektid ja skeemid.</li> <li>Rakenduse piiratud õigustega kasutaja, kellena pöördub rakendusserver/rakendus.</li> </ul> Objektide loomiseks vajalikud õigused ja ressursid on loetletud rakenduse dokumentatsioonis.	Need õigused, mis on vajalikud ainult rakenduse baasi loomiseks, on eraldi välja toodud ja tuleb peale installi ära võtta. Karbitoodete puhul tuleb erisused läbi arutada TEHIK arhitektiga.  Lahenduse puhul, milles kasutatakse andmebaasi versiooneerimist, tuleb kasutada mitut andmebaasi ühendust. Ennem rakenduse käivitumist teostatakse andmebaasi skeemi muudatused eraldi kasutajaga. Rakendus kasutab enda põhitööks kasutajat, kellel puudub õigus andmebaasi skeemis muudatusi teostada.	Arendaja	Administraator
2.18	Failide hoidmise asukoht lepitakse iga kord kokku, kuid failid ja failide indeks peavad olema replikeeritavad teise serveriruumi.	Failide hoidmine klassikalises andmebaasis on kulukas ja seab kõrgendatud nõudmised ja piirangud andmebaasiserveritele. Lahenduse dokumentatsioonis tuleb ära tuua failide hoidmise asukoht.  Näiteks objektide talletuseks kasutada S3 lahendust.	Arendaja	Arhitekt
2.19	Peab olema miinimumini viidud vajadus, et haldur teeb haldustoiminguid otse baasis. St rakendusel peab olema haldusliides, mille kaudu rakenduse haldur saab teha tavapäraseid haldustoiminguid.	Halduri haldustoimingud lepitakse tellijaga kokku detailanalüüsi käigus.	Arendaja	Testija
2.20	Rakendus peab olema võimeline kasutama keskkonnamuutujaid (serverinimi, kuu, päev jne).	Näiteks logifailides.	Arendaja	Testija
2.21	Andmebaas peab toetama nii külm- kui ka kuumvaru (peegeldamist) teise serviruumi.	Ei tohi kasutada teenuseid, mis välistavad andmebaasi peegeldamist (nt "MSSQL filestream").	Arendaja	Arhitekt
2.22	Sorteerimisreeglistik peab olema Eesti tähestikule vastav. Tõusutundlikkus peab olema välja lülitatud. Accent peab olema sisse lülitatud.	Näiteks PostgreSQL puhul et_EE.	Arendaja	Testija
2.23	Kui infosüsteemid saadavad e-kirju, peavad nad kasutama välist e-mailserverit. Kirja saatmisel peab rakendus veenduma, et e-posti server võttis kirja vastu. E-kirjade vormindamine peab järgima interneti standardeid (RFC 5322).	Saatja ja aadressaadid, pealkiri ja sisu ei tohi olla rakendusse kodeeritud, vaid on muudetavad konfiguratsioonifaili kaudu. Genereeritud kirjade puhul peab tagama kirjade jälitatavuse (näiteks lisada X-päise kodeeritud kirje, milles on kirjeldatud, mis protsess/skriptifail/kasutaja kirja genereeris jms abistav info).	Arendaja	Administraator
2.24	Konfiguratsiooniparameetrite nimed peavad olema sisulised. Kui see ei ole võimalik, siis peab kõrval olema seletus.	Näiteks : X-TEE-TURVASERVER, mitte XTTS või viitenumber, mitte vk_seb jne	Arendaja	Administraator Testija
2.25	Infosüsteemides on eessüsteemid (front end; presentatsiooni kiht) ja tagasüsteemid (back end; äriloogika kiht) arhitektuuriliselt selgelt lahutatud ja eraldi paigaldatavad.	Koostöövõime raamistik 2011. Punkt 3.1. Tagasüsteemide ülesanneteks on andmete haldamine ja võrguteenuste pakkumine. Tagasüsteemid ei tegele lõppkasutaja autentimise ja autoriseerimisega. Lõppkasutaja autoriseerimise tagavad eessüsteemid. Väliste süsteemi tõrge tohib mõjutada ainult sellest otseselt sõltuvate kasutuslugude toimimist. Väliste süsteemi taastumisel peab süsteem olema suuteline oma tööd jätkama taaskäivitamata.	Arendaja	Arhitekt
2.26	Konfiguratsioonifailid peavad olema vastavalt rakendusserveri tüübile vähimisi kaitstud failid/objektid	Näiteks IIS: *.config , *.resources Apache: *.conf, .htaccess.  Arendaja peab välja tooma konfigifailide listi, kui neid on mitu.	Arendaja	Administraator
2.27	Rakenduse failid, mida kasutaja näha ei tohi, peavad olema vähimisi kaitstud kaustades ja ei tohi olla veebi juurkaustas.	Näiteks: IIS: Bin,App_Code, App_Data, App_Browsers, App_GlobalResources, App_LocalResources, App_Themes, App_WebReferences, .git	Arendaja	Administraator
2.28	Konfiguratsiooniparameetrite taaskasutus. Erinevaid sama sisuga parameetreid ei tohi konfiguratsioonis eksisteerida.	Kõiki parameetreid tuleks konfiguratsioonis kirjeldada vaid korra, mitte nii, et igas lõigus kirjeldatakse samu asju uuesti.	Arendaja	Administraator Testija
2.29	Kõik rakenduse liidesed peavad olema võimelised töötama kõrgkäideldavalt.	Rakendustes tohib kasutada vaid masinapõhiseid teenuseid, mis lubavad kõrgkäideldavaid (klaster) lahendusi. Kõrgkäideldav lahendus on selline, mida saab samaaegselt käitada erinevates masinates.	Arendaja	Administraator
2.30	Klientrakendus ei tohi pöörduda otse andmebaasi poole.	Tuleb kasutada rakendusservereid.	Arendaja	Administraator
2.31	Keskkonnapõhised muutujad peavad olema konfiguratsioonifailist seadistatavad.	Näiteks WSDL ei tohi sisaldada viiteid arendusserveritele.	Arendaja	Administraator Testija
2.32	Eelistada tuleb tsentraalseid autentimislahendusi (nt TEHIK SSO).	Kui rakendus realiseerib ise autentimist, siis peab olema võimalik piirata ebaõnnestunud logimisi ajaühiku kohta (mobiil-ID, paroolid) ühelt IP-aadressilt. <ul style="list-style-type: none"> <li>Eelistama peaks IP-aadressipõhist blokeeringut. Erandina tellijaga kokkuleppel võib kasutada captcha või konto lukustamist. Blokeeringute ajavahemikku ja logimiskatsete arvu peab saama konfiguratsioonifailist muuta.</li> </ul>	Arendaja	Arhitekt Testija
2.33	Rakenduse äriloogika tuleb realiseerida andmebaasist eraldi sõltumatus rakenduskihis.	Andmebaas ei tohi sisaldada äriloogikat, mis muudab andmetabelites olevaid/sinna kirjutatavaid andmeid, va trigerid, mis tekitavad logi.	Arendaja	Arhitekt

2.34	Relatsioonilises andmebaasis võib kasutada vaid ISO/IEC 9075 standardiga kaetud funktsionaalsusi. Lisaks ei tohi kasutada ka sama standardi osas 13 kirjeldatud funktsionaalsusi.	<ul style="list-style-type: none"> <li>Ei ole soovitatav kasutada mingit platvormispetsiifilist lahendust, mille üleviimine mõnele muule andmebaasiplatvormile ei ole võimalik.</li> <li>ISO/IEC 9075 osa 13 spetsifitseerib Javas kirjutatud programmimoodulite kasutamist andmebaasis.</li> </ul>	Arendaja	Arhitekt
2.35	Kui rakendus eeldab eraldi kasutajate, rollide ja õiguste registri pidamist, siis peab rakendus kasutama Tellija tsentraalseid autoriseerimislahendusi.		Arendaja	Arhitekt Testija
2.36	Uniform resource identifier (URI) pikkus ei tohi ületada ühegi IS poolt toetatava brauseri maksimaalset lubatud väärtust.	Harilikult on piiriks 2000 tähemärki, kuid iga IS puhul tuleb seda eraldi järele uurida sõltuvalt IS komponentidest. Asjakohased viited: RFC 3986 ja RFC 7239.	Arendaja	Administraator Testija
2.37	Veebiteenuseid (REST, SOAP) pakkuv rakendus peab olema üles ehitatud nii, et see toetaks teenuste versioneerimist URL-i ja/või schema tasemel.	Näiteks WSDL puhul: Alajaotis definitions/types/schema: <ul style="list-style-type: none"> <li>complexType defineerimisel tuleb sellele lisada any element.</li> </ul>	Arendaja	Arhitekt
2.38	Rakendus peab olema võimeline töötama koormusjaoturitega varustatud taristul.	Koormusjaoturi peal kasutatakse järjestikuplaanurit (Round Robin) päringute suunamisel. Samuti võidakse teostada koormusjaoturi TLS ühenduse lahtivõtmist ja uuesti kokkupanemist (SSL offload) .	Arendaja	Administraator
2.39	Sidusinfosüsteemide mitte kättesaadavus ei tohi segada rakenduse töötamist. Sidusinfosüsteemidega andmevahetamisel tekkinud vead logitakse ja kasutajat hoiatatakse.	Sidussüsteemi tõrge tohib mõjutada ainult sellest otseselt sõltuvate kasutuslugude toimimist. Sidussüsteemi taastumisel peab süsteem olema suuteline oma tööd jätkama rakendust taaskäivitamata.  Väliste liidestatud süsteemide tõrke korral ei tohi süsteem hanguda, vaid väljastama mõistliku (võimalikult lühikese) aja jooksul ajakohase veateate. Võimalusel tuleb kasutada asünkroonseid liideseid.	Arendaja	Administraator Testija
2.40	Automaatselt käivituvaid taustatöid peab saama käsitsi (taas)käivitada.	Vajalik juhuks, kui automaatsel käivitumisel on tekkinud viga ja/või taustatöö on pooleli jäänud.  Pärast vea põhjuse korrigeerimist peab saama taustatöö uuesti käivitada.	Arendaja	Testija
2.41	Kui ajastatult käivitav taustatöö, ei ole mõeldud käima paralleelselt, peab selles olema realiseeritud kontrollmehhanism, mis tagab, et sama taustatöö ei ole võimalik käivitada uuesti enne, kui eelmisena käivitatud instants on oma töö lõpetanud.		Arendaja	Administraator Testija
2.42	Uue toote arenduse ja olemasolevate infosüsteemide versiooniuuendustel kasutusele võetavate tehnoloogiate ja standardite valik tuleb kooskõlastada Tellija poolse arhitektiga.		Arendaja	Arhitekt
2.43	Rakenduse ühenduste (s.h. andmebaasi ja sidusinfosüsteemide ühendused) realiseerimisel tuleb kasutada ühenduste puulimist (connection pooling).	Implementeeritud peab olema vähemalt maksimaalsete ühenduste arvu piirang, päringu aegumise aeg (request timeout) ja ühenduse elususe periood (keepalive). Rakenduse ühenduste tõrge tohib mõjutada ainult sellest otseselt sõltuvate kasutuslugude toimimist. Ühenduste taastumisel peab rakendus olema suuteline oma tööd jätkama taaskäivitamata. Tekkinud vead logitakse ja kasutajat hoiatatakse.	Arendaja	Arhitekt
2.44	Rakenduse uuendustega kaasnevad andmebaasi muudatused tuleb automatiseerida.	Näiteks Liquibase või Flyway.	Arendaja	Administraator
<b>3. Turvalisuse tagamisega seotud nõuded</b>				
3.1	Asutusesiseseks kasutamiseks mõeldud rakenduse kasutajate autoriseerimist peab saama teha Active Directory põhiselt.		Arendaja	Turvatestija
3.2	Kliendi ja serveri vahel peab autentitud kasutajasessioonide korral olema sessioon krüpteeritud HTTPS-protokolli kasutades.		Arendaja	Turvatestija
3.3	Rakendus tohib kasutada vaid sessiooni küpsiseid (cookies). Muude küpsiste kasutamine on keelatud.		Arendaja	Turvatestija
3.4	Kui andmebaasis olevate andmete ISKE tervikuse turvaosaklass on 3 (kõrge), siis tuleb kõik andmebaasi kirjed/tabelid versioneerida.	St kõik andmemuudatused peavad baasis säilima. Andmete muutmisel andmeid ei kustutata, vaid tehakse uus kirje uute andmetega. Vana muudetakse kehtetuks. Iga uus kirje peab sisaldama järgmist informatsiooni: <ul style="list-style-type: none"> <li>viide kirjele, mille ta kehtetuks muutis (kui on);</li> <li>kasutaja, kes kirje lõi;</li> <li>kirje loomise aeg;</li> <li>sessiooni-ID (kui on olemas). Iga kehtetuks tunnistatud kirje peab; omama järgmist informatsiooni;</li> <li>kasutaja, kes kirje kehtetuks tunnistas;</li> <li>kirje kehtetuks tunnistamise aeg.</li> </ul>	Arendaja	Turvatestija Arhitekt
3.5	Rakendusega peab kaasas olema lahendus, mis suudab toota toodangu andmetest testandmed, mis ei sisalda konfidentsiaalset informatsiooni.	Testandmed peavad säilitama kõik toodangu andmete omadused (pikkuse, tüübi) ja omavahelised suhted.  Täpsem vajadus ja tegevusplan tuleb koostada TEHIKu arhitekti ja tootemanikuga.	Arendaja	Arhitekt

3.6	Rakendusse ja andmetele tohib olla ligipääs vaid dokumenteeritud ja tellimuses kirjeldatud teid mööda ning dokumenteeritud autentimisprotseduure kasutades.	St rakendustes ega andmebaasides ei tohi olla ligipääsemiseks teisi võimalusi.	Arendaja	Turvatestija
3.7	Kõigi rakenduse poolt salvestatud paroolide ja salaküsimuste vastused peavad olema räsitud+soolatud. Kui räsime asemel valitakse krüpteerimine, siis tuleb dokumenteerida ka krüptovõtme turvalise hoidmise protseduur.	Hetkel kehtivad nõuded saab TEHIKU arhitektilt. Arendaja loodud lahenduse dokumentatsioonis (nt detailanalüüs vms) peab olema ära toodud kasutatavad räsi ja krüptoalgoritm, võtme pikkused ja nende kasutuskohad (vt nõue p 1.10)	Arendaja	Turvatestija
3.8	Rakendus ei tohi teostada X-tee päringut otse kasutajaarvutist.	Kasutajaarvutitest otse x-tee päringute tegemine on arvutivõrgu tasemel kinni.	Arendaja	Turvatestija
3.9	Veebipõhised väliste veebilehete rakendused peavad kasutama vahendeid, kaitsmaks rakendust lubamatute päringute eest.	IIS puhul peab kasutama näiteks URL scan, apache puhul modsecurity või vastavat tööriista. Lubamatud päringud on kõik päringud, mis ei ole detailanalüüsi käigus vastavalt kasutusjuhudele ette nähtud. Kasutama peab whitelisting põhimõtet, mitte blacklisting.	Arendaja Administraator	Turvatestija
3.10	Kasutaja peab saama soovi korral veenduda, kas keegi pole tema nime all vahepeal sisse loginud.	Rakendus peab sisemenisel näitama pärast õnnestunud sisselogimist eelmise õnnestunud sisselogimise aega. Kui on toimunud ebaõnnestunud sisselogimise katseid, siis peab kuvama ka, millal need toimusid, mitu neid oli ja mis IP-aadressilt pöörduti.  Ebaõnnestunud logimiste katsete kuvamise nõue kehtib juhul, kui autentimine ja autoriseerimine lahendatakse rakenduses lokaalselt.	Arendaja	Turvatestija
3.11	Kõigil rakendustel peab olema konfigureeritav kasutajasessiooni aegumise aeg.	Aeg peab olema muudetav koos teiste konfiguratsiooniparameetritega.	Arendaja	Turvatestija
3.12	Autentitud sessiooni tunnus peab olema krüpteeritud.	Sessiooni ei tohi olla võimalik üle võtta sessioonitunnuse kopeerimisega ühest arvutist teise.	Arendaja	Turvatestija
3.13	LDAP lahenduse (nt AD) kasutamisel peab rakendus kasutama kontoga kaasnevaid piiranguparameetreid.	Näiteks: konto on lukus, parool aegunud, konto aegunud, paroolipoliitika jne.	Arendaja	Turvatestija
3.14	Tagada tuleb rakenduse rollide lahusus.	Peakasutajal ja tavakasutajal on erinevad tööülesanded. Rollide/õiguste kirjeldus peab lähtuma detailanalüüsist ja kasutusjuhudest.	Arendaja	Turvatestija
3.15	Arendus peab olema orienteeritud toodangukeskkonnas toimimiseks.	Toodangukeskkonna rakendus ei tohi sisaldada osiseid, mis toodangu keskkonnas on ebavajalikud või segavad (näiteks kasutuse funktsionaalsus ja komponendid, mõeldud testimiseks testkeskkonnas, arendusabi arenduskeskkonnas jne).	Arendaja	Turvatestija
3.16	Rakendus ei tohi lubada ühe kasutajaga mitut samaaegset sessiooni.	Juhul kui lähteülesanne spetsiaalselt ei ütle teisiti.	Arendaja	Turvatestija
3.17	Kui rakenduse tervikluse turvaosaklass on T3, peavad tõestusväärtust omavad andmed olema kas ajatembeldatud, digiallkirjastatud või digitembeldatud.	Milline lahendus valitakse tuleb kokku leppida tellijaga.  Täpsustuseks vt ISKE nõue HT.34. Vt ka nõuet 3.28.	Arendaja	Turvatestija
3.18	Kui rakenduse tervikluse turvaosaklass on T3, peavad tõestusväärtust omavad andmed olema krüptoheldatud, et tagada et tõestusväärtusega andmeid ei saaks märkamatuks kustutada.	Täpsustuseks vt ISKE nõue HT.10. Krüptoahela kasutamise vajadus lepitakse eraldi kokku Tellija infrastruktuuri juhiga ja infoturbejuhiga. See sõltub Tellija keskse krüptoahela kasutamise võimalustest.	Arendaja	Turvatestija
3.19	Kui rakenduses on S3 salastuse astmega andmeid, peavad need olema nii transpordi ajal ja ka salvestatult alati krüpteeritud.	Täpsustuseks vt ISKE nõue HT.37.	Arendaja	Turvatestija
3.20	Paksu kliendi korral ei tohi rakendus kasutaja tööjaama jätta maha krüpteerimata kujul ajutisi faile, mis sisaldavad või võivad sisaldada konfidentsiaalse või kõrget terviklust nõudvat informatsiooni.	Kui paks klient kasutab ajutisi faile, tuleb tagada nende perioodiline kustutamine tagamaks, et ei koormata liigselt kasutaja arvutit.  Nõude eesmärk on tagada, et rakenduse sulgemisel ei jääks kasutaja arvutisse maha informatsiooni, mida sinna jääda ei tohiks, sh pääsuandmeid, isikuandmeid, andmekogu sisulisi andmeid jms	Arendaja	Turvatestija
3.21	Rakendus ja selle komponendid peavad võimaldama kasutada keskkondade lahusust	Arendaja arendab arenduskeskkonnas ja annab tarne üle tellijale paigalduspakkidena. Tellija paigaldab selle testkeskkonda ja testib ning seejärel paigaldab tarne toodangu keskkonda. Reaalseid andmekogu andmeid tohib töödelda üksnes toodangu keskkonnas.  Üldjoones on kõik keskkonnad majutatud TEHIKU majutuses.	Arendaja	Turvatestija
3.22	Rakendus peab võimaldama hõlpsalt välja vahetada aegunud ja ebaturvalise krüptoalgoritmi.	Krüptograafiat kasutatav rakendus ei tohi nimeliselt välja kutsuda krüptograafilisi algoritme, vaid peaksid seda tegema vahendavate vaheteekide kaudu üldiste funktsioonide järgi (nt krüpteerimine, dekrüpteerimine, signeerimine, signatuuri verifitseerimine jne). Dokumentatsioon peab kajastama üldist kirjeldust, kuidas vajadusel ebaturvaline krüptoalgoritm välja vahetada.	Arendaja	Turvatestija
3.23	Rakenduse andmebaasi krüpteerimisega seotud andmeväljad peavad olema muudetava pikkusega.	Andmebaasides kasutatavad krüpteerimisfunktsioonid tingitud lisaväljad peaksid olema muudetava pikkusega, et formaati muutmata saaks kasutada teistsuguste parameetritega krüpteerimisalgoritme.	Arendaja	Turvatestija
<b>4. Logimine</b>				
4.1	Logimiseks tuleb kasutada standardseid komponente kogu logiahela ulatuses.	Näiteks Java raamistikku log4j, SLF4J, logback, transpordiks syslog, fluentd; logi formaadiks JSON. Logi peab olema loetaval tekstilisel kujul, et logikirjeid saaks töödelda masinloetavalt ja inimloetavalt.	Arendaja	Arhitekt Administrator
4.2	Logikomponent peab võimaldama rakenduse administraatoril määratleda ja muuta logide väljundit, logimise taset ja logimise formaati.	Näiteks logide saatmiseks syslog, fluentd või muu kokkulepitud meetod. Logimisparameetreid peab saama muuta rakendust taaskäivitamata.	Arendaja	Arhitekt Administraator Testija

4.3	Logid peavad olema jaotatud loogiiselt.	Auditlogi (Seansilogi, tegevuslogi) - info sisselogimiste, väljalogimiste ja seansi aegumiste kohta. Vigased sisselogimise katsed. Info õiguste suurendamise kohta. Peab olema logitud ka tühja või puuduvate parameetritega logimise katsed. Kogu informatsioon kasutajate tegevuste kohta koos tegevuse tüübi, seansi parameetrite (korreleerimaks seansi- ja tegevuslogi) ja kasutaja poolt esitatud sisendparameetritega (sh. väliste ressursside kasutamise kohta). Logida tuleb nii õnnestunud kui ka ebaõnnestunud tegevusi. Tehniline logi - rakendusserveri poolt loodud logi Vealogs - erinevate vealokordade info Silumislogi - arendajate jaoks vajalik debug info	Arendaja	Arhitekt Administraator
4.4	Logimine peab olema optimeeritud.	Informatsiooni dubleerimist loiges tuleb vältida, kui ei ole nõutud teisiti.	Arendaja	Arhitekt Testija
4.5	Logides peab olema maksimaalselt üks sündmus ühel real.	Mitme realiste sündmuste puhul kasutada kodeerimist või JSON formaati. Mitme realiste sündmused võivad olla tehnilises-, vea- või silumislogis kui rakendusserver ei oska seda kodeerida.	Arendaja	Administraator Testija
4.6	Logikirje peab olema JSON formaadis, kui ei lepita kokku teisiti.	Kui logikirje ei ole JSON formaadis, siis ühe ja sama sündmuse väljad peavad olema erinevates logikirjetes samas kohas.	Arendaja	Arhitekt Testija
4.7	Logiväljade nimed peavad olema normaliseeritud.	Võimalusel tuleb aluseks võtta Elastic Common Schema. Näiteks samatüübilised logiväljade nimed peavad olema ühtsed üle logi.	Arendaja	Arhitekt Infoturbespetsialist
4.8	Rakendus peab logima kasutaja edukat ja ebaedukat autentimist ja sessiooni lõpetamist, kasutaja IP ja autentimismeetodit (ID-kaart, mobiil-ID, Smart-ID vms), eduka autendi puhul tuleks logida ka kasutaja isikukood ja mobiil-ID ning Smart-ID puhul telefoninumber.	Logima peab ka autentimise ebaõnnestumise koos põhjusega (vale juurdepääsumandaat, aegunud konto jne). Logida tuleks IP-aadress, meetod ja kui võimalik kasutajatunnus (mobiil-ID puhul telefoni number, ID-kaardi puhul isikukood).  Kui rakendus kasutab kasutajate autentimiseks välist autentimise/autoriseerimise vahendit, siis leppida eraldi kokku autentimise detailsus ehk mida kajastatakse autentimise/autoriseerimise vahendis ja mida rakenduses.	Arendaja	Arhitekt Testija
4.9	Üle terve logi peab olema kasutaja sessiooni käigus tehtud tegevusi või sama sündmust võimalik siduda loogiiselt kokku.	Tegevuste sidumiseks peab olema võimalik logikirjeid siduda ühise välja abil. Selleks ei sobi kellaaeg, IP ega isikukood. Sobib näiteks unikaalne ID, mis ei tohi olla sessiooni ID, sest seda saaks logist välja lugeda ja rünnakuks ära kasutada. Võib olla sessiooni ID räsi koos transaktsiooni ID'ga.  Konkreetne lahendus tuleb valideerida TEHIKu arhitektiga.	Arendaja	Arhitekt Testija
4.10	Andmete loomise/vaatamise/muutmise /kustutamise tegevused peavad olema kajastatud logides. Logida tuleb ka päringud, mille vastus on puhverdatud.	Logikirjes peab sisalduma piisavalt informatsiooni, et vastata küsimustele kes, mida, kus, kust, millal, kuidas ja tulemus.	Arendaja	Testija Turvatestija Infoturbespetsialist
4.11	Administraatorite ja haldurite poolt tehtavaid andmete vaatamised, muutmised sh kustutamised (ka otse baasis) tuleb logida. Muutmise puhul tuleb logida nii uus kui ka vana väärtus.	Lahendus peab tagama, et administraatorid/haldurid ei saa andmete vaatamise, muutmise logimist ise (ka tavakasutajate logimist) deaktiviseerida või logisid kustutada/muuta.	Arendaja	Infoturbespetsialist Testija
4.12	Kui parameetri väärtus on tühi, tuleb see logis märkida asendusväärtusega.	Näiteks NULL	Arendaja	Testija
4.13	Logis tuleb kõik mitte kuvatavad (non-printable) sümbolid kodeerida.	Näiteks reavahetused -> \n, non-printable sümbolid - 0x00..0x1f, 0x7f..0xff.	Arendaja	Testija
4.14	Rakendus peab logima kõiki rakenduses tekkivaid tehnilisi vigu.	Logi sisaldab minimaalselt vea tekkimise aega, veakoodi, veakirjeldust (stack trace, traceback vms), võimalusel kasutaja andmeid, HTTP-, GET- ja POST-parameetrid ja nende väärtusi. Logimise detailsusrežiimi (info, warning, error, debug) peab saama muuta.	Arendaja	Administraator Testija
4.15	Rakendus peab suutma logida kõiki X-tee teenuste kaudu liikuvaid andmeid. Peab olema võimalus logimist sisse-välja lülitada.	Vajalik eelkõige silumiseks ja toodangu keskkonna probleemide lahendamiseks.	Arendaja	Administraator Testija
4.16	Rakenduse funktsionaalsuse kirjeldusega tuleb luua ka logimise dokumentatsioon ja loginäidised. Koos funktsionaalsuse arendamisega tuleb luua ka loodava funktsionaalsuse logimine ja selle dokumentatsioon. Dokumentatsioon peab sisaldama logis kasutatud klassifikaatorite kirjeldusi.	Mida logitakse, kuidas sündmused on logis jagatud, logiridade näited.	Arendaja	Arhitekt Administraator Infoturbespetsialist
<b>5. Testimine</b>				
5.1	Rakenduse kõik üleantavad versioonid peavad enne tellijale üle andmist olema testitud	Testitulemused tuleb edastada tellijale koos rakenduse üleandmisega. Vaata lisaks nõuet 5.2 ja 5.3.	Arendaja	Testija
5.2	Lahendus peab olema minimaalselt 75% ulatuses kaetud automaatsete komponentidega (unit test).	Käivitatakse tellija pideva integreerimise (CI) keskkonnas (nt Gitlab) ja kaetust raporteeritakse lähtekoodi analüsaatoris (nt SonarQube).	Arendaja	Arhitekt Testija
5.3	Lahendus peab olema minimaalselt 50% ulatuses kaetud automaatsete vastuvõtutestidega.	Käivitatakse tellija pideva integreerimise (CI) keskkonnas (nt Gitlab) ja kaetust raporteeritakse lähtekoodi analüsaatoris (nt SonarQube).	Arendaja	Arhitekt Testija
5.4	Rakendusega peab olema kaasas skript jõudlustestide tegemiseks.	Jõudlustestide täpne kirjeldus tuleb kokku leppida detailanalüüsi käigus. Arendaja peab koos rakendusega tarnima skripti ja vajalikud tarkvaralised vahendid kokkulepitud jõudlustestide läbiviimiseks. Jõudlustestide läbiviimine ei tohi nõuda tellijalt omapoolset tarkvara arendamist, skriptide kirjutamist või litsentside ostmist.	Arendaja	Arhitekt Administraator Testija

6. Monitooring				
6.1	Rakendusel peab olema masinloetav testleht ( <i>health check</i> ) nt JSON või XML formaadis.	Testlehte kättesaadavus erinevatest arvutivõrkudest peab olema konfigureeritav. Testleht peab uuendama ennast lehe pärimisel. Testleht peab sisaldama custom built rakenduse versiooni numbrit, standardsed komponendid (veebiserver, andmebaas, CMS'id jms) ei tohi oma versioone reeta. Samuti peab testlehel olema infot rakenduse (vajadusel tema erinevate osade) ja tema kõigi väliste liideste staatuse kohta (töötab, ei tööta). Rakenduse, andmebaasi ja liideste töökorda kontrollitakse testpäringute teel, mis tuleb tellijaga kokku leppida eelanalüüsi käigus. Testleht peab oma konfiguratsiooni võtma rakenduse üldisest konfiguratsioonist (baasistring, välised ühendused).	Arendaja	Arhitekt Administraator
6.2	Rakendus peab pakkuma monitooringu lehte, kus leidub informatsioon rakenduse funktsionaalsuse toimimise kohta.	Näiteks Prometheus formaadis monitooringu leht, kus näeb infot päringu kestuste kohta, veakoodide ja nende hulka. Rakenduse mälu kasutamist jne.	Arendaja	Arhitekt Administraator
7. Nõuded rakenduse lähtekoodile				
7.1	Lähtekoodi kommentaarid peavad kõigis lahenduse kihtides (rakenduse enda kood, andmebaas jne) olema kirjutatud inglise keeles.	NB! Nõuet ei arvestata arendustarkvara poolt automaatselt genereeritavate koodilõikude puhul – neid ei ole vaja tõlkida. Samuti ei rakendata nõuet kolmandate osapoolte poolt toodetud lähtekoodile – nt igasugu erinevad lahtise koodiga koodilõigud jms.  Kui on tegu olemasoleva süsteemi edasiarendusega, siis peaks kommentaarides kasutama eelnevalt kasutatud keelt.	Arendaja	Arhitekt
7.2	Lähtekoodi kommentaarid peavad olema selged, arusaadavad ja sisuliselt kirjeldama vastavat koodi, mille juures nad on ning moodustama vähemalt 20% koodi mahust.	Rakenduse kood peab olema piisavalt hästi kommenteeritud, et erialast haridust omav tarkvaraarendaja on võimeline süsteemile jätkuarendusi teostama.	Arendaja	Arhitekt
7.3	Muutujate, tüüpide ja funktsioonide nimed peavad olema sisulised ja andma aimu nende otstarbest.	Parim praktika	Arendaja	Arhitekt
7.4	Koodis kasutatavad konstandid ja lühendid tuleb kirjutada suurte tähtedega, lähtudes kasutatava programmeerimiskeele parimast praktikast.	Nt Javas identifikaator --> ID	Arendaja	Arhitekt
7.5	Koodis kasutatavaid konstante ei tohi selle kasutamise kohta väärtusena hardcodeida – need tuleb defineerida muutujatena ja kasutada läbi nende.		Arendaja	Arhitekt
7.6	Koodis defineeritud andmetüübid peavad olema nimetava käände ainsuses. Kõik andmemassiivid tuleb nimetada nimetava mitmuses (st igasugu collectionid, arrayd, jms).	N:Isik; Menetlus; jne. Andmebaaside struktuurikirjeldustes/andmemudelid ei tohi kasutada täpitähti.	Arendaja	Arhitekt
7.7	Andmetabelites sisalduvad võõrvõtmed peavad nime järgi seostuma tabeli ja väljaga millele need viitavad.	Kasutada tuleb konkreetse andmebaasisüsteemi nimetamise parimaid praktikaid. Nt kui on tegu tabelitega 'Isikud' ja 'Autod', siis seos 'isiku autod' oleks: Isikud.ID=Autod.Isik_ID	Arendaja	Arhitekt
7.8	Andmebaasi väljade pikkused tuleb kirjeldada sümbolites, mitte baitides.	Selle asemel, et eraldada väljale x baiti, tuleb eraldada x tähemärki. (Instead of allocating x bytes of storage for the field, x chars of storage must be allocated).	Arendaja	Arhitekt
7.9	Kui kokku pole lepitud teisiti, siis JAVA rakenduse kood peab olema kirjutatud vastavalt "Oracle Java Code convention" dokumendile.	Checkstyle ( <a href="http://checkstyle.sourceforge.net/">http://checkstyle.sourceforge.net/</a> ) ei tohi käivitamisel järgmise konfiguratsioonifailiga väljastada ühtegi viga.	Arendaja	Arhitekt
7.10	Kui kokku pole lepitud teisiti, siis Python rakenduse kood peab olema kirjutatud vastavalt "Style Guide for Python Code" dokumendile	<a href="http://www.python.org/dev/peps/pep-0008/">http://www.python.org/dev/peps/pep-0008/</a>	Arendaja	Arhitekt
7.11	Kui kokku pole lepitud teisiti, siis .NET rakenduse kood peab olema kirjutatud vastavalt "NET Framework Developer's Guide - Design Guidelines for Developing Class Libraries".	<a href="http://msdn.microsoft.com/en-us/library/ms229042.aspx">http://msdn.microsoft.com/en-us/library/ms229042.aspx</a> Valideerimiseks kasutatakse 'StyleCop' vaikimisi konfiguratsiooni	Arendaja	Arhitekt
7.12	Koodi valideerimiseks kasutatakse minimaalselt TEHIK'u Sonarqube teenust.	<a href="https://sonar.tehik.ee/">https://sonar.tehik.ee/</a> ( <a href="https://www.sonarqube.org/">https://www.sonarqube.org/</a> )	Arendaja	Arhitekt
7.13	Kasutuses mitteolev kood tuleb rakenduse lähtekoodist kõrvaldada.		Arendaja	Arhitekt
7.14	Arendamisel kasutatakse DRY ja SOLID printsiipe	<a href="http://en.wikipedia.org/wiki/Don%27t_repeat_yourself">http://en.wikipedia.org/wiki/Don%27t_repeat_yourself</a> <a href="http://en.wikipedia.org/wiki/SOLID_(object-oriented_design)">http://en.wikipedia.org/wiki/SOLID_(object-oriented_design)</a>	Arendaja	Arhitekt
7.15	Üleantavas koodis ei tohi olla paroole, mida on kasutatud arenduse käigus	Kehtib ka siis, kui need on välja kommenteeritud. Kõik sellised paroolid tuleb asendada fraasiga "<password>".	Arendaja	Arhitekt
7.16	Rakenduste lähtekoodi tasemel ei tohi olla ühtegi sisse kodeeritud parameetrit, veateadet.	Eelpoolmainitu haldamine toimub failis või andmebaasis.	Arendaja	Arhitekt
8. Andmekvaliteet ja standardid				



8.1	Adressiandmete sisestamisel, kuvamisel ja hoidmisel tuleb lähtuda Vabariigi Valitsuse määrusest "Adressiandmete süsteem".	Liidestatakse Maa-ameti ADS teenusega. Esituslohis on lubatud liidestada In-ADS teenusega otsingu tarvis. Taustsüsteemides toimub liidestamine x-tee teenusega <a href="https://www.rigiteataja.ee/akt/103102017005?leiaKehtiv">https://www.rigiteataja.ee/akt/103102017005?leiaKehtiv</a>	Arendaja	Arhitekt Standardija Testija
8.2	Rakendus peab automaatselt eeltäitma kõik võimalikud andmeväljad, kui need andmed on varem riigile esitatud või kui nende väärtused on võimalik automaatselt arvutada.	Välja arvatud logimisvormi lahtirid autentimisel Näiteks: kirje sisestamise kuupäev, kasutaja nimi, sünnikuupäev jne	Arendaja	Testija
8.3	Tegevusalade andmete sisestamisel, kuvamisel ja hoidmisel tuleb lähtuda Vabariigi Valitsuse 10. Jaanuari 2008. a määrusest nr 11 "Klassifikaatorite süsteem" ja kasutada EMTAK infosüsteemis kehtivat klassifikaatorit.		Arendaja	Standardija Testija
8.4	Tervishoiuteenuste osutajate andmete kasutamisel peab kontrollima nende andmeid, sh tegevuslubade kehtivust Terviseameti Tegevuslubade registrist	Liidestatakse Terviseameti x-tee teenusega	Arendaja	Testija
8.5	Tervishoiutöötajate andmete kasutamisel peab kontrollima nende andmeid Terviseameti tervishoiutöötajate registrist	Liidestatakse Terviseameti x-tee teenusega	Arendaja	Testija
8.6	Meditsiiniandmete vahetamiseks tuleb kasutada meditsiiniandmete andmevahetuse standardit HL7	<a href="http://www.hl7.org">www.hl7.org</a> Vajadusel on lubatud ka teiste standardite kasutamine, kuid see tuleb tellijaga eraldi kokku leppida	Arendaja	Standardija Testija
8.7	Tervise valdkonna klassifikaatorid peavad olema kooskõlas Tellija OID-ide ja publitseerimiskeskusega	Võimalusel tuleb kasutada olemasolevaid OID-e ja klassifikaatoreid, mida vajadusel täiendatakse või luuakse uued ning publitseeritakse samuti publitseerimiskeskuses <a href="http://pub.e-tervis.ee">http://pub.e-tervis.ee</a>	Arendaja Tellija	Standardija
<b>9. Kasutajaliides</b>				
9.1	Kasutajaliidese kõik disainiotsused peavad olema kooskõlastatud tellijaga enne nende realiseerimist		Arendaja	Projektijuht
9.2	Veebipõhine kasutajaliides peab olema kasutatav enamlevinud veebibrauseritega, sh nutiseadmetel (Android, IOS)	Minimaalselt Microsoft Edge, Mozilla Firefox, Chrome ja Safari arenduse testimise hetkel tootja poolt toetatud versioonid. Täpsemad nõuded dokumendis "Front-end arendusreeglid"	Arendaja	Testija
9.3	Rakenduse värviskeem ja logo kasutamine peab vastama Tellija ametlikule visuaalsele identiteedile (CVI) ja disainijuhistele (UIG).	Kui tegemist on struktuurifondide projektiga on lisaks nõutud ka vastav SF sümboolika. Tellija ametlikud CVI esitluspõhjad, logo kasutusjuhend ja kõik logod (ka jpg-na) küsida tellijalt.	Arendaja	Testija
9.4	Kasutajaliidese kõik osad ja teated peavad olema eestikeelseid.	Kui soovitakse juurde eraldi ka muid keeli, siis see on spetsifitseeritud hankedokumentides	Arendaja	Testija
9.5	Avaliku kasutamiseks tehtav rakendus peab olema graafiliselt skaleeruv ja mugavalt kasutatav kõigi enamlevinud arvutite monitoride resolutsioonidega.	Toetatud peavad olema vähemalt resolutsioonid: 1920x1200, 1920x1080, 1680x1050, 1600x1200, 1440x900, 1360x768, 1280x1024, 1280x960, 1280x800, 1280x768, 1152x864, 1024x768, 1024x600. Ühegi nimetatud resolutsiooni korral ei tohi tekkida horisontaalset kerimisriba.	Arendaja	Testija
9.6	Sisemiseks kasutamiseks tehtav rakendus peab olema graafiliselt skaleeruv ja mugavalt kasutatav TEHIK'u töökohaprofiilis loetletud resolutsioonides.	Toetatud peavad olema töökohaprofiilis loetletud resolutsioonid. Ühegi nimetatud resolutsiooni korral ei tohi tekkida horisontaalset kerimisriba.	Arendaja	Testija
9.7	Hüpikaknaid (pop-up) ei tohi kasutada.	Silmas on peetud uusi veebilehtiseja aknaid avavaid hüpikaknaid	Arendaja	Testija
9.8	Kasutajaliidese toiminguni (põhi- ehk enamkasutatavad tegevused) navigeerimiseks peab kehtima 3 kliki printsiip, väljalogimiseks 1 kliki printsiip.	Kõik rakenduse kasutajaliidese tehtavad toimingud tohivad üksteisest olla maksimaalselt 3 hiirekliki kaugusel. Toimingut ei pea nende 3 klikiga tehtud saama. Väljalogimise nupp/link peab olema ühe kliki kaugusel ja arusaadavas/intuiitvses kohas.	Arendaja	Testija
9.9	Kasutajaliides peab alati küsima kinnituse andmete kustutamise ja massmuutmiste kohta kui pole kokku lepitud teisiti.		Arendaja	Testija
9.10	Rakenduse kasutamisel tekkinud veale peab kasutajaliides vastama kasutajale eestikeelse kasutajasõbraliku veateatega, mis sisaldab soovituslikult ka vea koodi. Veateated peavad olema hallatavad.	Veateated peavad olema sellised, mis võimaldavad IT-abil võimalikult lihtsalt tuvastada vea olemuse ja asukoha.	Arendaja	Testija
9.11	Kasutajaliides peab olema ilma rakenduse koodi muutmata tõlgitav teise keelde, v.a kui ei ole kokkulepitud teisiti.	Uue keele lisamine peab olema teostatav konfiguratsiooni failist või administreerimisliidestest.	Arendaja	Testija
9.12	Rakenduse tausta, menüüde ja teksti värvid peavad olema ilma rakenduse koodi muutmata vahetatavad.		Arendaja	Testija



9.13	Rakenduse kasutajaliides peab teavitama kasutajat ette sessiooni aegumisest.	Ette teavitamise aeg peab olema konfigureeritav.	Arendaja	Administraator Testija
9.14	Kui vormile sisestatakse mahukaid andmevälju peab kasutajaliides kokku lepitud ajavahemike järel salvetama välja sisu, et sessiooni aegumisel või võrgu katkestuse korral juba sisestatud andmed ei kaoks.	Kui vorm koosneb paljudest väiksest andmeväljadest (nt taotlus), siis jagatakse vorm etappideks ning salvestatakse vastava etapi lõpus.	Arendaja	Testija
9.15	Sisestusvormidel andmete sisestamisel peab saama väljade vahel vastavalt ärioloogikale liikuda klaviatuuri abil tabulaatoriga.		Arendaja	Testija
9.16	Interaktiivsete vormide puhul (näiteks faili üleslaadimine), ei tohiks lehe värskendamisega tegevust korrata (faili taas üles laadida, andmeid saata, avaldust esitada).		Arendaja	Testija
9.17	Kui päring võtab aega kauem kui 3 sekundit, peab kasutaja saama visuaalse teate, et süsteem tegeleb päringu läbiviimisega.	Ikoon peab muutuma liivakellaks ja/või kuvatakse teade: päringut sooritatakse või muu tellijaga kokkulepitud indikaator.	Arendaja	Testija
9.18	Esilehel (sisselogimata) ja ka pärast kasutaja sisselogimist peab olema lihtne võimalus teavitada kasutajat muudatustest või probleemidest. Teavitus peab olema halduri poolt lihtsalt lisatav ja olema kasutajale märgatav.	Näiteks võimalikud teavitused: mingi süsteemi osa on vigane, tuli mingi uus funktsionaalsus, vahetage oma parool, uuendage isikuandmeid jne.	Arendaja	Testija
9.19	Nutiseadmetele tuleb luua eraldi kohaldatud kasutajaliides.	See vajadus spetsifitseeritakse arenduse tellimisel.	Arendaja	Testija
9.20	Päringu vastusena kuvatud tabeli veerge on võimalik andmete/teksti tähestikulises järjekorras sorteerida.		Arendaja	Testija
9.21	Andmeväljade kohustuslikkus peab olema infosüsteemi väljadel märgitud tärniga (*).		Arendaja	Testija
9.22	Rakenduse andmeväljade mõisted peavad olema üheselt identifitseeritavad, korrektses eesti keeles (ilma kirjavigadeta) ja vajadusel sisaldama selgitavat teksti. Abiinfo (kasutusjuhendid) peab olema kättesaadav rakenduse toimimise erinevatel etappidel.		Arendaja	Testija
9.23	Kasutajaliides peab vastama ka dokumendis "Front-end arendusreeglid" kirjeldatud reeglitele.	<a href="#">Front-end arendusreeglid</a>	Arendaja	Arhitekt Testija
9.24	Kasutajaliidesel peab olema hooldusteate võimekus.	Hooldusteate sisu, kuvamise aeg/period peab olema seadistatav ja ei tohi nõuda esitluskihi uuesti kobileerimist.  Mõistlik on hooldusteate võimekiust juhtida taustteenuse abil. Näiteks läbi esitluskihi jaoks loodud seadete REST liidese.	Arendaja	Testija Arhitekt Administraator
<b>10. Dokumentatsioon</b>				
10.1	Lõppkasutajatele ja avalikkusele suunatud rakenduse dokumentatsioon peab olema kirjutatud eesti keeles.	Erandiks võivad olla kolmanda osapoole komponentide (mis pole kirjutatud tellija jaoks) dokumentatsioon. Samuti võib erandiks olla väliste osapooltega seotud projektid. Erandid tuleb kooskõlastada tellijaga enne dokumentatsiooni koostamist	Arendaja	Projektijuht Arhitekt Administraator Testija Infoturbe spetsialist
10.2	Lahendus kirjeldatakse RIHA määruse nõuete kohaselt.	<a href="https://www.riigiteataja.ee/akt/12933746?leiaKehtiv#para6">https://www.riigiteataja.ee/akt/12933746?leiaKehtiv#para6</a>	Arendaja Projektijuht Tellija RIHA haldur	Projektijuht
10.3	Rakenduse dokumentatsioon peab vastama ka dokumendis "Nõuded infosüsteemi dokumentatsioonile" kirjeldatud nõuetele	Dokumentatsioon peab olema versioneeritud, muutmiskuupäevadega, autori nimedega, korrektses keelekasutusega, selge struktuuriga. Dokumentatsiooni detailsus peab olema piisav, et sõltumatu kolmas tehniliste IT baasteadmistega isik suudaks dokumendist vajalikke järeldusi teha (st dokument peab olema arusaadav sellele isikule, kuid näiteks paigaldusjuhise järgi toimetades ei pea ta ebaõnnestunud tarnele teostama veaanalüüsi).  <a href="#">Nõuded infosüsteemi dokumentatsioonile</a>	Arendaja	Projektijuht Arhitekt Administraator Testija Infoturbe spetsialist

10.4	Rakenduse dokumentatsioon peab sisaldama tabelite-andmete-logide mahu kasvu arvestuslikku hinnangut rakenduse sihipärase kasutamise korral ettenähtud arvu kasutajate poolt. (MB /GB kuus/aastas).	Esiolgne kirjete mahu hinnang peab tulema lähteülesandest, ning täpsustuma eel- ja detailanalüüsi käigus. Mahuhinnang peab sisaldama ka logide säilitamise, arhiveerimise tähtaegu.	Arendaja	Projektijuht Arhitekt Administraator Infoturbe spetsialist
10.5	Iga uue versiooniga peab alati välja tooma versiooni muudatuse kirjeldused (release notes).	Release notes peab kajastama kõiki muudatusi eelmise ja uue versiooni vahel.	Arendaja	Projektijuht
10.6	Arendaja loodud lahenduse dokumentatsioonis (nt detailanalüüs vms) tuleb välja tuua kasutatavad krüpto- ja räsiialgoritmid, nende võtmepikkused, kasutuskohad, sh TLS sertifikaatide kasutuskohad.		Arendaja	Arhitekt Administraator
<b>11. Versioonihaldus</b>				
11.1	Kogu rakenduse testimiseks, koolituseks või implementeerimiseks üle antav lähtekood ja tarkvarapaketi peavad olema versioneeritud. Kasutama peab Tellija versioonihalduse ja tehiste (artifaktide) repositooriumi.	Arendajale antakse selleks õigused Tellija versioonihalduse repositooriumi, kus ta peab hoidma oma erinevaid versioone. Versioonihalduse repositooriumi juurdepääsutaotlus esitatakse Tellija kasutajatoele läbi projektijuhi.	Arendaja	Arhitekt Administraator
11.2	Arendaja peab veenduma, et teeb muudatusi aktuaalsesse koodi.	Hea tava on, et paralleelse arendamise puhul võetakse igal hommikul versioonihalduse repositooriumist viimane seis koodist.	Arendaja	Arhitekt Administraator
11.3	Nii arendamisel kui ka hoolduslepingute korral kasutatakse Tellija tööde ja veahalduse keskkonda.	Arendajale antakse selleks õigused Tellija tööde ja veahalduse keskkonda. Veahalduse keskkonda juurdepääsutaotlus esitatakse Tellija kasutajatoele läbi projektijuhi.	Arendaja	Projektijuht
<b>12. Paigalduspaketi kooste</b>				
12.1	Juhul kui versioonihalduse keskkond ei paku paigalduspaketile kontrollsumma (checksum) automaatselt koostamist, siis koostatakse kontrollsumma arendaja poolt ja pannakse eraldi .sum failina tarnele kaasa.	Räsiialgoritms tuleb kasutada SHA256. Linuxi käsurealt kontrollkoodi koostamiseks: \$ sha256sum filename [filename2] ... > kontrollkood.sum.	Arendaja	Administraator
12.2	Tarnitava lahenduse koosseisus üle antava lähtekoodiga peavad kaasas olema kirjeldused sellest paigalduspaketi koosteks.	Näiteks võib lahenduse paigalduspaketi koosteprotsess ette näha, et käivitada tuleb rida shell-käsku või võivad lahenduse koosseisus olla valmis (ant, ..) koosteskriptid või mis iganes muu moodus paigalduspaketi tekitamiseks.  Eelistatud on kasutada Dockerfile ja Gitlab töövooge.	Arendaja	Projektijuht Administraator
12.3	Kooste kirjelduste alusel valmiv paigalduspakett tohib sisaldada ainult minimaalse rakenduse käitamiseks vajamineva failikomplekti.	Näiteks: kompilleeritavate keelte puhul ei tohi sisaldada lähtekoodi, kui see pole vajalik rakenduse käitamiseks.	Arendaja	Arhitekt Administraator
12.4	Kooste kirjelduste alusel valmivat paigalduspaketti peab olema võimalik liigutada erinevate masinate vahel.	Näiteks ei tohi tekitada olukorda, kus rakenduse jooksutamiseks uues serveris tuleb see tingimata just sealsamas kokku kompilleerida.	Arendaja	Administraator
12.5	Rakenduse kõik sõltuvused peavad olema kompilleerimisel saadavad Tellija tehiste repositooriumist.	<a href="https://repo.tehik.ee">https://repo.tehik.ee</a>	Arendaja	Arhitekt
12.6	Andmebaasi paigalduse skriptid ei tohi olla kompilleeritud.	Administraator tahab veenduda skripti sisus.	Arendaja	Administraator
12.7	Rakenduse lähtekoodi juures peab leiduma skriptid rakenduse keskkonnast sõltumatult (konteinerlahenduses) kokku kompilleerimiseks.	Tellijal peab olema võimalik suuri pingutusi tegemata ja keskkonna erinevusi vältides teha rakendusest paigaldatav pakk.	Arendaja	Arhitekt
12.8	Rakenduse lähtekoodi juures peab leiduma skriptid rakenduse lokaalselt mõnes konteinerlahenduses (Docker) käivitamiseks.	Vajadusel peab konteinerlahendus käivitama ka rakenduse muud sõltuvused (näiteks andmebaas).  See on Täitjale uue meeskonnaliikme liitumise lihtsustamiseks ja Tellijale võimalus suuri pingutusi tegemata süsteemi testimiseks.	Arendaja	Arhitekt
12.9	Paigalduspakett koostatakse Tellija pideva integratsiooni (continuous integration - CI) ja paigaldus (continuous deploy - CD) arendus keskkonnas.		Arendaja	Arhitekt
12.10	Kubernetesel (K8s) orkestreeritavate lahenduste paigalduste jaoks tuleb luua <i>Helm chart</i> .	<a href="https://helm.sh/">https://helm.sh/</a>  Helmi jaoks kasutatava malli annab Tellija poolne arhitekt.	Arendaja Arhitekt	Arhitekt Administraator