Test specification E2E OAT

TC-ID	Testcase	Description	Manual test steps			
	INT_WalletApp_Citizen_scans_QR-Code	scann the QR-code with the wallet app		Step	Input/Data	Expected Results
			1	open the (internal) QR-codescanner via "scan code"		QR-codescanner starts
TXR-2028			2	position the QR-Code under the camera		QR-code is displayed sharply
	INT_WalletApp_shows_the_certificate_on_mobile_device	show the saved certificate on mobile device within the details of the data		Step	Input/Data	Expected Results
			1	open the internal storage		all scanned QR-codes will be listed
TXR-2029			2	Choose one QR-code		QR-code will be displayed on screen
	INT_WalletApp_biometric_security	on Start the WalletApp a biometric request has to start. To be sure that a verified person get access to WalletApp Data		Step	Input/Data	Expected Results
			1	open WalletApp on Mobile Device		biometric data are requested
TXR-2032			2	scan your biometric data		WalletApp starts
	INT_WalletApp_negative_biometric_security	on Start the WalletApp a biometric request has to start. To be sure that a verified person get access to WalletApp Data		Step	Input/Data	Expected Results
			1	open WalletApp on Mobile Device		biometric data are requested
TXR-2033			2	scan wrong biometric data		Error: Access denied
	INT_VERIAPP_verify_qr_code_for_a_valid_dgc	A Digital Green Certificate with: 1) a valid QR Code; 2) valid Payload; 3) valid Attributes. is presented for offline verification. The Verifier App confirms that the DGC is valid. It also tests that the same DGC can be verified twice by the same VeriApp instance.		Step	Input/Data	Expected Results
			1	VeriApp scans QR-Code.		QR-Code is approved as verified.
			2	VeriApp scans the same QR-Code for a second time.		QR-Code is again approved as verified.
TXR-2075						
	INT_VERIAPP_neg_verify_qr_code_with_invalid_signature	A Digital Green Certificate (DGC) with invalidly signed QR-Code is presented for offline verification. The Verifier App evaluates the DGC as invalid.		Step	Input/Data	Expected Results
TXR-2077			1	VeriApp scans QR-Code.		The VeriApp evaluates the DGC as invalid.
	INT_VERIAPP_neg_verify_qr_code_with_invalid_payload_syntax	A Digital Green Certificate with correct signature but syntactically invalid payload (e.g. missing name etc.) is presented for verification. The signature is validated but the DGC is evaluated as invalid due to invalid Payload. An Error Code "Invalid Payload" is returned.		Step	Input/Data	Expected Results
			1	VeriApp scans QR-Code.		QR-Code signature is approved as valid.
TXR-2079			2	VeriApp reads payload.		The DGC is evaluated as invalid. An Error Code "Invalid Payload" is shown.
	INT_VERIAPP_render_dgc_for_type_PCRtest	A validly signed Digital Green Certificate of type (PCR) TEST is presented for verification. The testcase tests presentation of the DGC Data for the DGC of type test, independently of test result (postive or negative).		Step	Input/Data	Expected Results
			1	VeriApp scans QR-Code.		The DGC is approved as valid and the Contents Data is presented for type TEST. The content is preseneted as a positive or negative quick test.
TXR-2084						
	INT_VERIAPP_render_dgc_for_type_vac	A valid Digital Green Certificate of type VAC (owner has been vaccinated) is presented for verification. The testcase tests presentation of the DGC Data.		Step	Input/Data	Expected Results
			1	VeriApp scans QR-Code.		The DGC is approved as valid and the Contents Data is presented for type VAC.
TXR-2085						
	INT_VERIAPP_render_dgc_for_type_rec	A valid Digital Green Certificate of type REC (owner has recovered) is presented for verification. The testcase tests presentation of the DGC Data.		Step	Input/Data	Expected Results
			1	VeriApp scans QR-Code.		The DGC is approved as valid and the Contents Data is presented for type REC.
TXR-2086						
	INT_VERIAPP_fetch_an_use_manually_trigered	The Verifier App has to support the manual triggering of the synchronisation process.		Step	Input/Data	Expected Results
			1	The VerifierApp has been installed. Internet connection is available. It has been less than 24 hours since the last synchronisation. The user triggers the synchronisation manually.		A Synchronisation process has been triggered and the keys have been updated.
TXR-2087						
	INT_VERIAPP_fetch_and_use_resynchronise_after_offline_state	This testcase examines the case where no synchronisation has taken place in the last 24 hours due to missing internet connection. As soon as the internet connection is available again, the verifier app should initiate synchronisation.		Step	Input/Data	Expected Results
			1	The VerifierApp has been installed and at it is has been 24 hours since the installation.		A Synchronisation process has been triggered and the keys have been updated within the last 24 hours.
			2	After the synchronisation has been done, the internet is switched off for at least 24 hours.		No synchronisation of the keys database could take place.
			3	The internet connection is available again.		The verifier app initiates synchronisation (fetch and use) within the next 24 hours.
TXR-2088						

TXR-2089	INT_VERIAPP_fetch_an_use_daily_synchronisation	The Verifier App has to synchronise its public key database daily with the backend. Internet Connection is available.		Step	Input/Data	Expected Results
			1	The VerifierApp has been installed and at it is has been 24 hours since the installation.		A Synchronisation process has been triggered and the keys have been updated within the last 24 hours.
TXR-2094	INT_VERIAPP_render_dgc_for_test_result_positive	A validly signed Digital Green Certificate of type POSITIVE TEST (owner has tested positive) is presented for verification. The testcase tests presentation of the DGC Data.		Step	Input/Data	Expected Results
			1	VeriApp scans QR-Code.		The DGC is read and a positive test result is displayed.
TXR-2103	INT_WalletApp_register_QR-Code_with_TAN	The QR-code is only allowed to save on one device. Therefor the citizen gets a TAN wich can be used only one time. After the registration, the TAN can't be used twice.		Step	Input/Data	Expected Results
			1	scann QR-code with integrated barcode-scanner		Barcode will be shown on screen
			2	push save button		TAN will be requested
			3	insert valid TAN		scanned QR-code will be saved
TXR-2105	INT_WalletApp_start_WalletApp_with_PIN	If the citizen has no biometric data on his mobile device it shoul be possible to start the device by PIN		Step	Input/Data	Expected Results
			1	start the WalletApp on mobile device		biometric data are requested
			2	user push cancel		a user PIN is requested
			3	insert the correct PIN		WalletApp starts
TXR-2106	INT_WalletApp_negative_register_QR-Code_with_TAN_-_TAN_expired	The QR-code is only allowed to save on one device. Therefor the citizen gets a TAN wich can be used only once for a defined time after creation. (Expirationtime has to be defined) After this time, the TAN can't be used anymore.		Step	Input/Data	Expected Results
			1	scann QR-code with integrated barcode-scanner		Barcode will be shown on screen
			2	push save button		TAN will be requested
			3	insert expired TAN		An error occurred: TAN expired QR-code will not be saved
TXR-2107	INT_WalletApp_negative_register_QR-Code_with_TAN _wice	The QR-code is only allowed to save on one device. Therefor the citizen gets a TAN wich can be used only one time. After the registration, the TAN can't be used twice.		Step	Input/Data	Expected Results
			1	scann QR-code with integrated barcode-scanner		Barcode will be shown on screen
			2	push save button		TAN will be requested
			3	insert valid TAN a seconed time		an error occurred: TAN can't be used twice
TXR-2182	INT_VERIAPP_render_dgc_for_test_result_negative	A validly signed Digital Green Certificate of type Negative TEST (owner has tested negative) is presented for verification. The testcase tests presentation of the DGC Data.		Step	Input/Data	Expected Results
			1	VeriApp scans QR-Code.		The DGC is read and a negative test result is displayed.
TXR-2187	INT_WalletApp_valid_TAN_which_does_not_belong_to_this_qr-code	Issuer has created two different QR-codes. Each with valid TAN. He gave citizen A qr-code A with with valid TAN to qr-code B. He gave citizen B qr-code B with with valid TAN to qr-code A. So, we have A valid TAN which belongs to an other valid QR-code.		Step	Input/Data	Expected Results
			1	scann QR-code with integrated barcode-scanner		Barcode A will be shown on screen
			2	push save button		TAN will be requested
			3	insert valid TAN B which does not belong to this qr-code (dgci)		TAN B will be accepted by wallet app
			4	send data to national backend		national backend will proof the data and returns an error to wallet app
			5	get error code from national backend		qr-code will not be saved
TXR-2200	INT_WalletApp_use_deep_link_by_SMS_to_import_certificate	Optionally you can use a deep link instead of a 2D Code to initiate the certificate import in the wallet app. The deep link can look like: dgc://example.authority.com?token=ey… & [publickey] In this case the token is received with the link, and the public key must be replaced by the key of the new generated key pair of the certificate container in the wallet app. The deep link can be delivered by SMS, Email or by resenting another 2D Code for scan.		Step	Input/Data	Expected Results
			1	Get a deep link by sms	dgc://example.authority.com?token=ey… & [publickey]	open deep link from sms
			2	WalletApp should start		WalletApp is opening and request biometric data
			3	scan correct biometric data		WalletApp starts the import of the certificate
	INT_WalletApp_use_deep_link_by_email_to_import_certificate	Optionally you can use a deep link instead of a 2D Code to initiate the certificate import in the wallet app. The deep link can look like: dgc://example.authority.com?token=ey… & [publickey]		Step	Input/Data	Expected Results
			1	Get a deep link by email	dgc://example.authority.com?token=ey… & [publickey]	open deep link from email

TXR-2201		In this case the token is received with the link, and the public key must be replaced by the key of the new generated key pair of the certificate container in the wallet app. The deep link can be delivered by SMS, Email or by resenting another 2D Code for scan.	2	WalletApp should start		WalletApp is opening and request biometric data
			3	scan correct biometric data		WalletApp starts the import of the certificate
TXR-2202	INT_WalletApp_negativ_use_fake_deep_link_by_SMS_to_import_certificate	Optionally you can use a deep link instead of a 2D Code to initiate the certificate import in the wallet app. The deep link can look like: dgc://example.authority.com?token=ey… & [publickey] In this case the token is received with the link, and the public key must be replaced by the key of the new generated key pair of the certificate container in the wallet app. The deep link can be delivered by SMS, Email or by resenting another 2D Code for scan.		Step	Input/Data	Expected Results
			1	Get a fake deep link by sms	dgc://example.authority.com?token=ey… & [publickey]	open deep link from sms
			2	WalletApp should start		WalletApp is opening and request biometric data
			3	scan correct biometric data		WalletApp shows an error with invalid link
TXR-2203	INT_WalletApp_negativ_use_fake_deep_link_by_email_to_import_certificate	Optionally you can use a deep link instead of a 2D Code to initiate the certificate import in the wallet app. The deep link can look like: dgc://example.authority.com?token=ey… & [publickey] In this case the token is received with the link, and the public key must be replaced by the key of the new generated key pair of the certificate container in the wallet app. The deep link can be delivered by SMS, Email or by resenting another 2D Code for scan.		Step	Input/Data	Expected Results
			1	Get a fake deep link by email	dgc://example.authority.com?token=ey… & [publickey]	open deep link from email
			2	WalletApp should start		WalletApp is opening and request biometric data
			3	scan correct biometric data		WalletApp shows an error with invalid link
TXR-2205	INT_WalletApp_start_WalletApp_with_wrong_PIN	If the citizen has no biometric data on his mobile device it shoul be possible to start the device by PIN		Step	Input/Data	Expected Results
			1	start the WalletApp on mobile device		pin is requested instead of biometric data. Only works when no biometric data are saved
			2	insert the wrong PIN		WalletApp shows an error
TXR-2823	INT_VERIAPP_choose_CoD	Precondition: The default ruleset (CoA) is the initial setting. Verifier has to choose the ruleset from CoD (Country of departure). The Verifier then switches back to the default ruleset (CoA).		Step	Input/Data	Expected Results
			1	Verifier starts VerifierApp		VerifierApp starts
			2	Verifier selects "settings"		Settings will open. The default ruleset (CoA) is initially loaded.
			3	Verifier selects the ruleset of CoD		Ruleset will be set to CoD.
			4	Verifier switches back to default ruleset CoA.		Default ruleset CoA is loaded.
TXR-2824	INT_VERIAPP_choose_CoA	A user with a verifier app (“verifier”) in the country of arrival (CoA) wants to check whether a DCC holder fulfills all requirements of the CoA. The CoA is his default setting for scanning the provided		Step	Input/Data	Expected Results
			1	Verifier starts VerifierApp		VerifierApp starts
			2	Verifier selects "settings"		The default for ruleset CoA is set.
TXR-2825	INT_VERIAPP_business_validation_QR-code	Given a QR-Code which is technically valid, a business validation is to be run with respect to the QR-Code's Country of Issuance. The test checks the good case scenario where business validation is successful (i.e. positive test case).		Step	Input/Data	Expected Results
			1	Verifier scans a technically valid QR-Code. The QR-Code is created in a way that the business rules applied to the country of issuance are valid as well.		The QR-Code is scanned. Technical check is passed. Country of Issuance is read.
			2	Verifier sets the ruleset corresponding to the displayed Issuance Country, unless this is already the default value set.		The rules value set for the country of issuance is set.
			3	Verifier scans the QR-Code again, this time with the rules valueset being set.		VerifierApp shows green certificate, due to successful business validation.

TXR-2827	INT_VERIAPP_check_verificationDateTime_is_currentDateTime	The rules are checked against the Verification DateTime. If no Verification DateTime is provided, it will be filled with the current date and time. Precondition: The QR code contains information on 1 event with one entry: either a vaccination, a negative test, or a recovery statement (V, T or R). Create a valid PCR- or RAT test with the "time rule" (tbd) where verification DateTime is missing. Outcome: Verifier App will use the current date and time as verification DateTime.		Step	Input/Data	Expected Results
			1	Verifier App scans a valid PCR- or RAT test with the "time rule" (tbd) where verification DateTime is missing.		VerifierApp shows a green validation result
TXR-2828	INT_VERIAPP_negative_check_verificationDateTime_is_currentDateTime	The rules are checked against the Verification DateTime. If no Verification DateTime is provided, it will be filled with the current date and time. Precondition: The QR code contains information on 1 event with one entry: either a vaccination, a negative test, or a recovery statement (V, T or R). Create a valid PCR- or RAT test with the "time rule" (tbd) where verification DateTime is missing. Outcome: Verifier App will use the current date and time as verification DateTime.		Step	Input/Data	Expected Results
			1	Verifier App scans an expired PCR- or RAT test with the "time rule" (tbd) where verification DateTime is missing.		VerifierApp shows a red validation result
TXR-2829	INT_VERIAPP_check_signingExpiration_supersedes_certificateExpiration	In this case, we want to check, that the signing certificate expiration datetime supersedes the expiration datetime in the Green certificate. Precondition: A qr-codes which is signed with a DSC-certificate which becomes invalid during testing but the green certificate will be valid longer. For example: green certificat validity: 31.12.2022 DSC validity: 01.08.2021		Step	Input/Data	Expected Results
			1	Scan green certificate wich is actually valid		verifier App shows green validation result
			2	wait a few hours	time has to be defined	Verifier App shows invalid certificate
TXR-2830	INT_VERIAPP_check_issuer_timeZone_is_used	In this case, we check that the time zone specified in the QR code is taken into account when checking the period of validity. Precondition: A QR-Code created with respect to a different time zone than the time zone of the verifier app device.		Step	Input/Data	Expected Results
			1	Set device time to Finnish time zone		the timezone of the device is set to Finnish timezone
			2	Scan qr-code generated within German device time		VerifierApp shows the correct UTC-time
TXR-2831	INT_VERIAPP_ruleset_is_used_for_validation	In this Testcase, we check that during the validation, all valid rules in the ruleset for selected country are actually checked against the qr-code. For this perpose we need a QR-Code which conforms to all but one the selected country's rules.		Step	Input/Data	Expected Results
			1	Given a QR-Code which conforms to all but one the selected country's rules, scan it with the verifier app.		A validation error occurs and the verifier app feedbacks the validation result to the user in a table format.
	INT_VERIAPP_negative_check_ruleengine_version	Check that in the event of an incompatible rule engine, e.g. "Rule engine version> current version" , The human readable fallback case is used (Section 6.4.1 Incompatible Rule Engine Version).		Step	Input/Data	Expected Results

			1	Given incompatible rule engines, the verifier app scans a technically valid QR-Code.		The human readble fallback is presented as feedback in a table format.
TXR-2832		Precondition: a set up with incompatible rule engines.				
	INT_WalletAPP_ruleset_is_used_for_validation	In this Testcase, we check that during the validation, all valid rules in the ruleset for selected country are actually checked against the qr-code.		Step	Input/Data	Expected Results
			1	Choose a certificate		Details of the certificate is shown
		Precondition:	2	choose a ruleset of a eu-country, where not the certificate was created		The ruleset is choosen to check and each rule is checked. The screen shows the validation result. (valid = green arrow, invalid = red cross)
TXR-2848		Citizen has a valid certificate stored in the walletApp				
	INT_WalletAPP_negative_check_ruleengine_version	Check that in the event of an incompatible rule engine "Rule engine version> current version" (?), The human readable fallback case is used (Section 6.4.1 Incompatible Rule Engine Version)		Step	Input/Data	Expected Results
			1	Given incompatible rule engines, the wallet app scans a technically valid QR-Code.		The human readble fallback is presented as feedback in a table format.
TXR-2849		(In case of incompatible rule-engine, the verifierApp checks the qr-code with the old ruleset.(?))				
	INT_WalletAPP_check_issuer_timeZone_is_used	In this case, we check that the time zone specified in the QR code is taken into account when checking the period of validity.		Step	Input/Data	Expected Results
			1	Set the Mobile to Finnish time zone		the timezone of the mobile is set to Finnish timezone
		Precondition:	2	scan qr-code generated with german time		walletApp request TAN
TXR-2850		a QR-code with another timezone than the tomezone of the mobile with the verifier app-- in this case the timezone for Finland - is created.	3	Insert valid TAN		WalletApp shows certificate and the correct UTC-time
	INT_WalletAPP_check_signingExpiration_supersedes_certificateExpiration	In this case, we want to check, that the signing certificate expiration datetime supersedes the expiration datetime in the Green certificate.		Step	Input/Data	Expected Results
			1	Scan green certificate wich is actual valid		WalletApp shows correct Data
		Precondition:	2	wait a few hours	time has to be defined	Wallet App shows invalid certificate
		A qr-codes which is signed with a DSC-certificate which becomes invalid during testing but the green certificate will be longer valid.				
		For example:				
TXR-2851		green certificat validity: 31.12.2022 DSC validity: 01.08.2021				
	INT_WalletApp_compareCertificate_with_country_ruleset	The holder must be able to select each onboarded country from the complete EU country list (In order to be able to check its selected ruleset against a specific selected QR code).		Step	Input/Data	Expected Results
		Precondition:	1	Choose a certificate		Details of the certificate is shown
		List of all onboared countries is known	2	choose a ruleset of a eu-country, where not the certificate was created		The ruleset is choosen to check and each rule is checked. The screen shows the validation result. (valid = green arrow, invalid = red cross)
TXR-2852		a valid certificate is already scanned and successfully claimed				
	INT_VERIAPP_process_result_invalid	The scanned techniclally valid QR Code contains a rule, which evaluates to false.		Step	Input/Data	Expected Results
			1	Verifier starts verifierApp		verifierApp starts successful
			2	Verifier choose the needed country ruleset		the needed ruleset will be loaded
TXR-2914			3	Verifier scans a technically valid but logically invalid qr-code.		VerifierApp shows red certificate and the details of result
	INT_VERIAPP_unsupported_Schema	6.4.2 Incompatible Schema		Step	Input/Data	Expected Results
		In the case of a schema version provided by the DCC which is unknown or not supported by the verification application, the verifier application can only present the complete content and let the verifier decide.	1	Verifier starts verifierApp		verifierApp starts successful
			2	Verifier choose the needed country ruleset		the needed ruleset will be loaded
		Prerequisite:	3	Verifier scans the QR-Code.		The complete content is shown in a human readable format, so that the verifier is able to make a decision himself.

TXR-2933		A QR-Code for a DCC with an unknown or unsupported schema.	

Test specification E2E OAT

TC-ID	Testcase	Description	Manual test steps			
		Open the WebApp in Browser		Step	Input/Data	Expected Results
	INT_IssApp_Start_WebApp		1	Open Browser	https://dgca-issuance-web.cfapps.eu10.hana.ondemand.com	WebApp is starting
TXR-2013						
		Insert relevant Data in Issuer App. Send inserted Data to national backed.		Step	Input/Data	Expected Results
			1	open the data entry mask		Data Entry Mask is shown
			2	insert Family name in textfield "Family name"		"Family name" is shown in textfield
			3	insert given name in textfield "Given name"		"Given name" is shown in textfield
			4	Choose Date of Birth Format		The textfield of DOB changes its format according to the choosen format
			5	insert date of birth in textfield with picker 'Date of Birth'		date is shown in textfield
			6	insert "Disease/Agent*" in textfield Disease/Agent*		"Disease/Agent*" is shown in textfield
			7	choose vaccination type in combo box 'Vaccine/Prophylaxis*'		vaccination type is shown textfield
	INT_IssApp_Create_QR-Code		8	choose medical product in combo box 'Medicinal Product*'		medical product is shown in textfield
			9	choose Organisations Management System* in combo box 'Organisations Management System*'		Organisations Management System is shown in textfield
			10	insert dose number in Textfield "Dose Number*"		dose number is shown in textfield
			11	insert total series of doses in Textfield "Total Series of Doses*"		total series of doses is shown in textfield
			12	insert vaccination date in textfield with picker'vaccination date'		vaccination date is schown textfield
			13	choose Issuer country in combo box 'Issuer Country*'		issuer country is shown in textfield
			14	insert certificate issuer in textfield "Certificate Issuer*"		Certificate Issuer is shown in textfield
TXR-2017			15	push "next" button		QR-code will be generated with inserted data
		send unsigned QR-code to national Backend, which signs it and send it back to the Issuer App. The signed QR-code will be displayed on screen		Step	Input/Data	Expected Results
	INT_IssApp_Request_signed_QR-Code		1	Send created QR-Code to national backend via "finish process" button		QR-Code will be send - national backend returns signed QR-Code
TXR-2019						
		print the QR-code/vaccination certificate with included print service(?)		Step	Input/Data	Expected Results
	INT_IssApp_Print_signed_QR-Code		1	Create signed QR-code		signed QR-code created
			2	Push the "Create PDF" Button		A PDF document is created and downloaded with all necessary dates
TXR-2020						
		Insert relevant Data in Issuer App with wrong birthdate. Start creation of QR-code. Get QR-code with wrong birthday. proof data in QR-code and find the misstake. correct birthday in Issuer App and create new QR-code.		Step	Input/Data	Expected Results
			1	open the data entry mask		Data Entry Mask is shown
			2	insert Family name in textfield "Family name"		"Family name" is shown in textfield
			3	insert given name in textfield "Given name"		"Given name" is shown in textfield
			4	insert wrong date of birth in textfield with picker 'Date of Birth'		date is shown in textfield
			5	insert "Disease/Agent*" in textfield Disease/Agent*		"Disease/Agent*" is shown in textfield
			6	choose vaccination type in combo box 'Vaccine/Prophylaxis*'		vaccination type is shown textfield
			7	choose medical product in combo box 'Medicinal Product*'		medical product is shown in textfield

TXR-2113	INT_IssApp_Create_corrected_QR-Code		8	choose Organisations Management System* in combo box 'Organisations Management System*'		Organisations Management System is shown in textfield
			9	insert dose number in Textfield "Dose Number*"		dose number is shown in textfield
			10	insert total series of doses in Textfield "Total Series of Doses*"		total series of doses is shown in textfield
			11	insert vaccination date in textfield with picker'vaccination date'		vaccination date is schown textfield
			12	choose Issuer country in combo box 'Issuer Country*'		issuer country is shown in textfield
			13	insert certificate issuer in textfield "Certificate Issuer*"		Certificate Issuer is shown in textfield
			14	push "next" button		QR-code will be generated with inserted data
			15	push "next" button		QR-code will be generated with inserted data
			16	check the inserted data		wrong birthday is shown
			17	push "correct patient data" button		inserted data will be shown in data entry mask
			18	edit birthday field and insert correct birtday		corrected birtday is shown
			19	push "next" button		QR-code will be generated with corrected data