TO DO

Ticket Number	Name	Pre Condition	Parent Folder Name (Testiny)	Steps	Expected Results	Note
EP-6669	IdP Integration for device users	Parent ticket of the following: EP-6754 | EP-6812 | EP-6816 EP-6817 | EP-6875 | EP-6887				
EP-6587	Edit Label to another existing Label	Label should be **listed** in **Manage Labels**	Customer Console **>** Regression Script **>** 004. Fleet Management **>** a. Device List	**1**. Go to Fleet Management > select Device List **2**. Click the **Manage Labels** dropdown > select **Edit / Delete** Screenshot for Step 2 **3**. Select a **Label** > click **Edit** Screenshot for Step 3 **4**. Change Label to an Existing Label (e.g. **QA Xiaomi Test Label** >> **QA Xiaomi**) Screenshot for Step 4	The **proper** error message should appear (e.g. "A label with the same name already exists!")	Upon reading this ticket, the real problem was not having a proper **error message** for duplicate labels when **EDITING** them. ** **CREATING** a **duplicate** label has this error message currently: Error Message Screenshot
EP-6820	Check if remote control should still be displayed + support URL to use (Contact Support button)	Remote Control feature is **not** activated **Instance** type must be **Jamf**.	Customer Console **>** Regression Script **>** 004. Fleet Management **>** d. Device Information Page **>** 009 Remote Control	**1**. Go to **Fleet Management** > select **Device List** **2**. Select a device **3**. Go to **Remote Control** Tab **4**. Click "**Contact Support**" Screenshot for Step 4	On **Jamf** instances, it should **point** to "https://account.jamf.com". Button label should be "**Contact Jamf Support**".	
EP-6745	Filtered Field Names	No Precondition	Customer Console **>** Regression Script **>** 007. Configuration **>** b. Wi-Fi Networks	**1**. Go to **Configuration** > select **Wi-Fi Networks** **2**. Check the **Description** of the **filter box** Screenshot for Step 2	Description of the filter box reads "Filter by name" **Currently**. Description of the filter box should keep **upper-case** names **as-is**, and remove the **double space** at the **beginning** of the field **names**, e.g.: "Filter by name, SSID, security protocol".	Not sure if this test case is correct, but this is what I understood based on the ticket
EP-6694	Prevent folder from being assigned to itself	Preexisting folders and devices must be present	Customer Console **>** Regression Script **>** 004. Fleet Management **>** a. Device List **>** Device Folders	**1**. Go to Fleet Management > Android Device List **2**. **Select** a Device with an assigned **folder** **3**. Click Manage Folders > assign Screenshot for Steps 2-3 **4**. Choose the **same folder** of **selected device** and click **OK** Screenshot for Step 4	Assigning a device folder to itself should not be allowed. (Can display an error message)	This is based on what was in the ticket
EP-6700	Only allow single-folder selection	Preexisting folders and devices must be present	Customer Console **>** Regression Script **>** 004. Fleet Management **>** a. Device List **>** Device Folders	**1**. Go to Fleet Management > Android Device List **2**. Click Manage Folders > Show Folder Section Screenshot for Step 2 **3**. **Click** a specific folder Screenshots for Step 3	Should only be able to select one folder at a time	
EP-6545	Zebra VisibilityIQ support	Zebra test devices are **enrolled** and **active**. VisibilityIQ integration feature flag is **enabled**. Parent ticket of the following: EP-6547 | EP-6679	Customer Console > Regression Script > 004. Fleet Management > VIQ Integration	**1**. **Confirm** test environment access (Partner Console and test customer accounts) **2**. **Review** device data flow between WizyEMM and Zebra VIQ. **3**. **Validate** API communication logs between microservice and VIQ endpoints. **4**. **Execute** all linked child work items.	Zebra VisibilityIQ integration works **properly**. Data is **transmitted** to VIQ without errors. Logs show **successful** API calls.	This is the **main** epic encompassing setup, testing, and API validation.
EP-6813	Check for user email duplicates on production environments	Must be on Production Server	Customer Console **>** Regression Script **>** 007. Configuration **>** a. Users	**1**. Go to **Configuration** > select **Users** **2**. Check for any **Duplicate** emails in the table Screenshot for Step 2	There should be **no duplicate emails** on production environments.	
EP-6777	Check if password field is used by customers	These tickets are for Developers only				
EP-6822	Use a separate translation version for dev/staging instances					
EP-6862	Grant new permissions for service deployment					
EP-6868	Update configuration file to wizy.io website					
EP-6887	Request new test devices for Manila					
EP-6875	Implement database schema for SCEP					
EP-6884	SCEP Proxy Service					
EP-6885	Android SCEP Client					
EP-6881	Implement Error Handling					
EP-6882	External CA Configuration					
EP-5992	Backend Cloud Run migration					
EP-6883	SCEP Profile Creation					
EP-6886	Configure SCEP on Device Profile					
EP-6818	Automate JSON file splitting					
EP-6819	Update translation spreadsheet from JSON files					
EP-6788	Secure Cloud Storage versioning and retention					
EP-6699	Only show folder hamburger menu for selected folder					
EP-6701	Make the folders panel resizeable					
EP-6869	CSV Export Improvements					

IN PROGRESS

Ticket Number	Name	Pre Condition	Parent Folder Name (Testiny)	Steps	Expected Results	Note
EP-6547	Test current status of VIQ integration	VIQ package has been **re-enabled** in the system. Test customer accounts exist and are **linked** to partners.	Partner Consnole > VIQ Integration	**1**. Log in to Partner Console as Partner Administrator. **2**. Navigate to Customer Management. **3**. Select a test customer with VIQ package assigned. **4**. Verify package visibility and details. **5**. Check Packages and Options History. **6**. Open Monthly Statements tab. **7**. Confirm that VIQ-related entries appear correctly.	VIQ package is **visible** and **functional** in Partner Console. Historical and billing data are displayed **correctly**. **No errors** or UI inconsistencies occur.	**Verify** against previous hidden state (**EP-5533**). **Ensure** all dependent modules load properly.
EP-6573	Enable VIQ package back	VIQ package previously hidden under **EP-5533**. **Access** to configuration and database entries **required**.		**1**. **Access** system configuration or database where packages are **listed**. **2**. **Unhide** or **re-enable** the VIQ package. **3**. **Validate** that VIQ appears in the Partner Console under available packages. **4**. **Assign** VIQ to a test customer and save changes. **5**. **Verify** no system crash or dependency errors occur.	VIQ package is **successfully** re-enabled and visible in the package list. Can be **assigned** to customers without error.	This ticket serves as a **prerequisite** for EP-6547 validation. **Confirm** visibility before running full integration tests.
EP-6679	Update microservice to support latest VisibilityIQ API changes	**Existing** microservice connected to Zebra APIs. Test credentials and API keys **available**. Parent ticket of the following: EP-6680 | EP-6691 | EP-6692		1. Deploy updated microservice with latest VIQ API compatibility. 2. Validate all submodules (Device Management, Battery Management, Application Health). 3. Verify successful data retrieval using latest API versions. 4. Monitor logs for error or mismatch with deprecated endpoints.	Updated microservice responds **correctly** to all Zebra VIQ APIs. **No error** in fetching device, battery, or application health data.	
EP-6691	Zebra VisibilityIQ Battery Management	**Updated** microservice with Zebra VIQ Battery API access **enabled**.		**1**. **Call** the Battery Management endpoint of the updated microservice. **2**. **Retrieve** battery health and cycle information for all connected devices. **3**. **Verify** data accuracy (health score, charge count, replacement recommendation). **4**. **Compare** results against Zebra console battery reports.	Battery data is **correctly** retrieved and reflects up-to-date health metrics per device. API response aligns with Zebra's **expected** data schema.	**Ensures** that battery analytics remain functional and compatible with the **new** API version.
EP-6692	Zebra VisibilityIQ Application Health	Microservice **updated** and API keys authorized for **Application Health** endpoint.		**1**. **Query** Application Health endpoint using the updated API. **2**. **Retrieve** metrics related to app uptime, version, and crash statistics. **3**. **Verify** response consistency with Zebra VisibilityIQ dashboard data. **4**. **Confirm** that old endpoints no longer return results (**deprecated**).	Application Health data is **properly** synchronized and displayed with no API errors. Metrics are **consistent** with Zebra's dashboard values.	**Confirms** successful migration of Application Health monitoring to the latest VIQ API version.
EP-6743	Fix Remote Control "Contact Support" button	- User is logged in as an **Admin** or **Manager**. - A device is **enrolled** and **accessible**. - The account is a **Jamf instance** (**not WizyEMM**).	Customer Console > Regression Script > 004. Fleet Management > c. Remote Control	**1**. Log in to the Customer Console (**Jamf environment**). **2**. Go to Fleet Management → Android Device List. **3**. **Select** an enrolled device and open the Remote Control tab. **4**. **Observe** the "Contact Support" button displayed when Remote Control is **not activated**. **5**. **Click** the "Contact Support" button.	- On **Jamf instances**, the button label should be "**Contact Jamf Support**." - The **button** should redirect to https://account.jamf.com. - The **previous** link to support@wizyemm.com should no longer be used. - On **non-Jamf instances**, the existing "Contact Support" behavior should remain unchanged.	- **Verify** across staging and production **Jamf** environments. - **Confirm** no regression in WizyEMM or **Chronopost** accounts. - **Check** responsive layout and text alignment for the **updated** button label. - **Validate** that the link opens in a new browser tab.

EP-6754	Enable Basic Managed Google Account Authentication (Fully ManagedDevices)	- Device is factory **reset** and ready for fully managed enrollment. - QR code and enrollment token are **configured** for AMAPI enrollment. - Test Google Workspace user **exists** in the sqlenduser table. - Internet connectivity is **available**. - Enrollment performed on a **fully managed** device (**not work profile**). Parent ticket of the following: EP-6756 | EP-6757 | EP-6777 | EP-6805 | EP-6806 | EP-6813	Customer Console >Regression Script >003. Profile Management >a. Enrollment Settings	**1**. **Start** device setup and scan the **fully managed** enrollment QR code. **2**. **Continue** through setup until the Google Workspace sign-in screen appears. **3**. **Sign in** using a valid **Google Workspace** managed account. **4**. **Observe** that the enrollment process continues **successfully** after authentication. **5**. **Repeat** the process using a **personal** Gmail account (**@gmail.com**). **6**. **Observe** the expected rejection or error message. **7**. **Check** backend logs or API responses for the authenticatedUserEmail value in the provisioningInfo parameter. **8**. **Verify** that enrollment succeeds only if the authenticated user **exists** in the sqlenduser table. **9**. **Confirm** that the device is **associated** with the corresponding user in the system.	- Google Workspace sign-in screen is **displayed** automatically after QR code setup. - **Managed Google** account authentication **succeeds** and **continues** to app installation. - **Personal Gmail** accounts are **rejected** with a clear error message. - Backend **verifies** authenticatedUserEmail from provisioningInfo against sqlenduser. - Enrollment **fails** if the user does **not exist** in the system. - Device is **properly linked** to the corresponding end user record. - Enrollment applies only to **fully managed** devices.	- **No auto-provisioning** in this version (v1). - **Feature** leverages Android Management API's Better Together support. - **Verify** proper error messaging for **missing** or **invalid** user accounts. - **Test** across multiple Google Workspace domains to **ensure** compatibility. - **Research** impact if managed account access is later **disabled** after enrollment.
EP-6812	Detect migration to Managed Google Domain type	- The AMAPI Service and backend are **deployed** with Liquibase migration applied. - The Pub/Sub topic for **ENTERPRISE_UPGRADE** is **enabled** in the environment (**dev** or **staging**). - The sqlcustomer table includes the following columns: - **enterprisetype** - **managedgoogleplayaccountsenterprisetype (nullable)** - **managedgoogledomaintype (nullable)** - A valid customer enterprise **exists** in the database with a known enterpriseid. - **Integration** between AMAPI Service and Pub/Sub is configured and operational. Parent ticket of the following: EP-6897 | EP-6898	Customer Console > Regression Script > 004. Fleet Management > c. Remote Control	**1**. **Trigger** an ENTERPRISE_UPGRADE event from AMAPI Pub/Sub containing upgradeState = "**UPGRADE_STATE_SUCCEEDED**". **2**. **Include** fields enterpriseId, managedGooglePlayAccountsEnterpriseType, and managedGoogleDomainType in the JSON payload. **3**. **Observe** the backend logs to **confirm** routing to ProcessEnterpriseUpgradeMessage(). **4**. **Verify** that the enterprise ID is **correctly** matched to an **existing** record in sqlcustomer. **5**. **Check** that all three database fields (**enterprisetype, managedgoogleplayaccountsenterprisetype, and managedgoogledomaintype**) are **updated** in one transaction. **6**. Confirm that the correct values are stored in the database: - **enterprisetype = "MANAGED_GOOGLE_DOMAIN"** - **managedgoogleplayaccountsenterprisetype = "CUSTOMER_OWNED"** - **managedgoogledomaintype = "DNS_VERIFIED"** **7**. **Validate** that the logs include **success** confirmation for the enterprise ID and customer name. **8**. **Confirm** that a Sentry breadcrumb is created for traceability. **9**. **Repeat** the test with a **missing** managedGooglePlayAccountsEnterpriseType field in the event. **10**. **Check** that the database sets managedgoogleplayaccountsenterprisetype **= NULL** and still updates other fields. **11**. **Repeat** the test using a **non-existent** enterpriseId and **verify** that an error is logged and sent to Sentry. **12**. **Test** with upgradeState set to **UPGRADE_STATE_IN_PROGRESS** and **confirm** that no database update occurs. **13**. **Trigger** the same successful event again to validate idempotency and **confirm** that updates reapply without errors.	- AMAPI Service routes the **ENTERPRISE_UPGRADE** notification **correctly**. - Database **updates** occur only for successful upgrade states. - sqlcustomer fields are **updated** exactly as per the event data. - **Partial** or **missing** subtype fields are handled **gracefully** (**NULL** assignment). -Unknown enterprise IDs **trigger** error logging and Sentry capture. - Non-success states **do not trigger** updates. - Idempotent updates **execute** without error on repeated events. - Log entries **confirm** successful updates with enterprise ID and customer name. - No database **integrity** or transaction errors occur.	-Liquibase migration must be **completed** before AMAPI Service deployment. - Manual Pub/Sub configuration (**ENTERPRISE_UPGRADE** type) must be **enabled** in all environments. - **Verify** successful processing through monitoring and logs. - **Ensure** sql.NullString handling in Go prevents crashes for absent subtypes. - Recommended to **verify** this behavior first in dev before staging rollout. - Feature **supports** migration detection only, **not initiation**.
EP-6816	Add "Mandatory" user signin option	- Customer has enterpriseType = **GOOGLE_MANAGED_DOMAIN**. - Profile **exists** with configurable usersignin field in sqlprofile table. - User logged in as **Super Administrator** or **Administrator**.	Customer Console > Regression Script > 003. Profile Management > a. Enrollment Settings	**1**. Log in to Customer Console as **Super Admin** or **Admin**. **2**. Go to Profile Management → Profile Details → Enrollment Tab. **3**. Under User Signin, view the "**Google Workspace Sign-in**" setting. **4**. Verify three available options: **Disabled, Optional, and Mandatory**. **5**. Select each option and save. **6**. **Enroll** a device using that profile and **observe** system behavior for each case (presence or absence of authenticatedUserEmail).	- **Mandatory**: Enrollment **blocked** if no authenticatedUserEmail. - **Optiona**l: Enrollment proceeds even if sign-in is **skipped**. - **Disabled**: Enrollment **bypasses** user authentication. - Only **Google** Managed Domain customers see all three options. - Non-Google customers see only "**Disabled**."	- No schema change required; only code logic update to handle new enum value **MANDATORY**. - **Enforced** server-side during enrollment validation. - Must **verify** behavior with **auto-insert** feature **enabled** and **disabled**. - **Error messages** must clearly indicate missing or invalid user sign-in. - **Deploy** backend and frontend updates together. - **Follow** WCAG 2.1 accessibility and cross-browser compliance.
EP-6897	[Backend] Detect migration to Managed Google Domain type					
EP-6898	[AMAPI Service] Detect migration to Managed Google Domain type			These tickets are for Developers only		
EP-6879	SCEPman Setup					

IN REVIEW

Ticket Number	Name	Pre Condition	Parent Folder Name (Testiny)	Steps	Expected Results	Note
EP-4888	Add Send Intent capability	- **Ensure** the target app/device is **ready** to receive intents - **Trigger** the SEND_INTENT command from the admin panel - **Confirm** that the target app launches or executes the specified action from the intent Parent ticket of the following: EP-6849 | EP-6707	Customer Console > Regression Script > 004. Fleet Management > b. Send Intent	**Add** Send Intent capability (details may be refined based on **UI/backend** functionality)	Admin can **successfully** configure and **enable** the Send Intent capability	**Further testing** will cover actual intent sending and validation
EP-6559	Chronopost custom behavior	Parent ticket of the following: EP-6564				
EP-6564	Prevent local admins from changing device profiles	Device is **registered** and **accessible**. **User** is logged in as a **local admin** Parent ticket of the following: EP-6619 | EP-6620	Customer Console > Regression Script > 004. Fleet Management > a. Device List > Device Folders	**1**. Log in to WizyEMM as a **local admin** **2**. **Navigate** to Fleet Management > Zebra **3**. **Identify** a device under your control **4**. **Attempt** to change the **Profile** field or **update** the device **5**. **Observe** if the system allows profile modification	**Local admin** is prevented from changing the device profile; Profile field is **locked** or Update action is **disabled**	Test on **multiple devices**; **Confirm** that this behavior applies only in **Chronopost** environments
EP-6817	Allow authenticated users to be automatically added to end users					
EP-6849	Receive and Launch Send Intent Command	Device/app is **configured** to receive intents; admin has enabled Send Intent	Customer Console > Regression Script > 004. Fleet Management > b. Send Intent	**1**. **Ensure** the target app/device is **ready** to receive intents **2**. **Trigger** the SEND_INTENT command from the admin panel **3**. **Verify** that the target app receives the intent **4**. **Confirm** that the target app launches or executes the specified action from the intent	Target app **successfully receives** the intent and performs the specified action	**Validate** that logging of the intent receipt is **correct**; test edge cases with invalid or malformed intents
EP-6856	ContentProvider crash on first call after reboot					
EP-6872	Generate stored procedure to export device details for a given regional admin			These tickets are for Developers only		

IN TEST

Ticket Number	Name	Pre Condition	Parent Folder Name (Testiny)	Steps	Expected Results	Note
EP-6494	Device Folders	Parent ticket of the following: EP-6694 | EP-6699 | EP-6700 EP-6701				
EP-6707	Send ACTION_VIEW intent	Admin is logged in and "**Send Intent**" capability is **available** Parent ticket of the following: EP-4888	Customer Console > Regression Script > 004. Fleet Management > b. Send Intent	**1**. **Navigate** to the admin panel **2**. **Open** the "**Send Intent**" feature **3**. **Verify** Intent Type is locked to **ACTION_VIEW** **4**. **Verify** Intent Format is locked to URI **5**. **Enter** a valid URI in Intent Body (single line) **6**. **Click** "Send Intent"	The **SEND_INTENT** backend command is **triggered** and **executed successfully**, target app opens the specified URI	**Ensure** the **SEND_INTENT_COMMAND** event type is emitted and logged correctly. **Test** invalid or empty URI for error handling
EP-6802	Google Workspace Sign-in	customer enterpriseType must be: **MANAGED_GOOGLE_DOMAIN**	Customer Console **>** Regression Script **>** 003. Profile Management **>** Inside a Profile **>** j. Enrollment Tab	**1**. Select a specific profile **2**. Go to **Enrollment Tab** **3**. Navigate to **User Signin** **4**. On **Google Workspace Sign-in**, choose any of the following options: **Disabled | Optional | Mandatory** Screenshot for Step 4 **5**. Click **Save** Screenshot for Step 5	User Signin Card must be **visible** The following behavior **must** happen during device enrollment: If **Disabled**: Google Workspace Sign-in will be **skipped** upon enrollment If **Optional**: Google Workspace Sign-in will be **available** but **not** required for enrollment If **Mandatory**: Google Workspace Sign-in will be **required** upon enrollment	
EP-6830	Add a side-loaded application	No Precondition	Customer Console **>** Regression Script **>** 006. Application Management **>** a. Managed Applications	**1**. Go to **Application Management** > select **Managed Applications** **2**. Click **+ Add** **3**. **Upload** an **APK** File **4**. Click "**OK**" Screenshot for Steps 1-4	Application should be seen in the table with the correct details.	

EP-6620	Change profile for chr local admins (on Device List)	Must be a **chr local admin**	Customer Console **>** Regression Script **>** 004. Fleet Management **>** a. Device List **>** Device Actions on Device List	**1**. Go to **Fleet Management** > select **Device List** **2**. The "**Change Profile**" option must be **hidden** or **disabled** Screenshot for Steps 1-2	Change profile option is **hidden** for **chr local admins**	
EP-6680	Zebra VisibilityIQ Device Management	**Updated** microservice with latest Zebra VIQ API **deployed** and **reachable**.	Customer Console > Regression Script > 004. Fleet Management > VIQ Integration	**1**. **Access** the Device Management endpoint via the **updated** microservice. **2**. **Retrieve** device details from Zebra VIQ. **3**. **Verify** all expected fields (device ID, model, serial number, last sync date) are returned correctly. **4**. **Confirm** data freshness and accuracy match Zebra console records.	Device data is **successfully** retrieved from Zebra VIQ using the **updated** API. **No missing** or mismatched fields compared to Zebra portal.	**Focuses** on verifying endpoint **/device-management** functionality and schema alignment after API version upgrade.
EP-6620	Change profile for chr local admins (on Device View)	Must be a **chr local admin**	Customer Console **>** Regression Script **>** 004. Fleet Management **>** d. Device Information Page	**1**. Go to **Fleet Management** > select **Device List** **2**. Select a device **3**. On Device Information page, "**Change Profile**" option must be **hidden** or **disabled** Screenshot for Step 3	Change profile option is **hidden** for **chr local admins**	
EP-6619	Block profile changes for chr local admins on the backend	These tickets are for Developers only				
EP-6651	Check if NYT requirements can be implemented					
EP-6730	Deleted customers are still generating BigQuery sync tasks					
EP-6785	Create SQL database user account for BigData service					
EP-6756	Backend: Create user lookup logic against sqlenduser table using authenticated email					
EP-6757	Backend: Implement device-to-user association in database					
EP-6758	Backend: Update QR code generation to include managed account requirements					
EP-6805	Backend: Add a new column enterpriseType to the customers table					
EP-6806	Backend: Support enterpriseType in customer attributes in the API					
EP-6807	Backend: Add a new column usersignin to the profiles table					
EP-6808	Backend: Support usersignin in profiles attributes in the API					
EP-6798	Backend fails to execute v1 endpoint requests					
EP-6799	Update BigData service to Go 1.23					
EP-6800	Update VisibilityIQ service to Go 1.23					
EP-6811	Hibernate error: Could not create proxy factory for get_blockedSecurityRisks getter marked final					
EP-6831	[Backend] Create a microservice to retrieve metadata automatically					
EP-6832	[Backend] Add an endpoint in the to expose the microservice to the frontend					
EP-6833	Backend: Add MANDATORY UserSignin Option					
EP-6846	Backend: Allow authenticated users to be automatically added to end users					
EP-6845	Create Custom BigQuery role to insert data					
EP-6877	Review Feedback Package Installer v2					
EP-6888	Support adding system apps as managed apps (Add a side-loaded application)					
EP-6895	Create SCEPman (and RADIUSaaS) account					