**Security Types:**

**Access Group:**

**Access Roles/Access Role Name: Roles Set for Every Operator/Class Names**

**Access of Role to Object: Environment Permissions**

**Privilege:  Rule Level Permissions**

**Access When & Access Deny:  To Give Access or Deny Access in Random Environments**

**Environments:**

**Case Study/Experimental: 1**

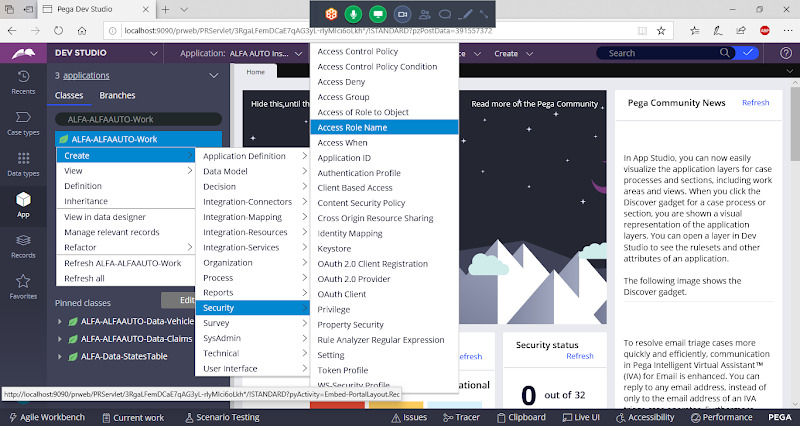
**Developer/Development: 2**

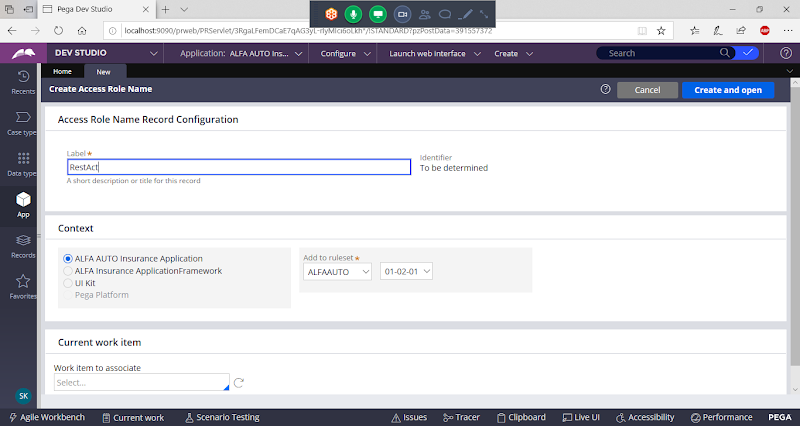
**Testing/QA: 3**

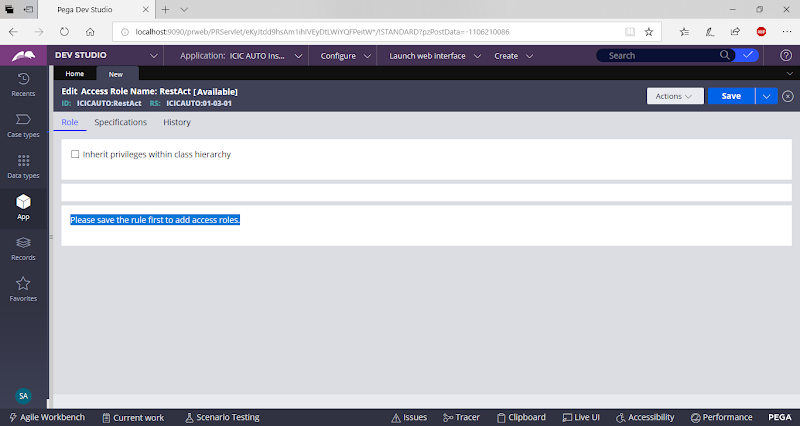
**Pre-Production/UAT : 4**

**Live/Production: 5**

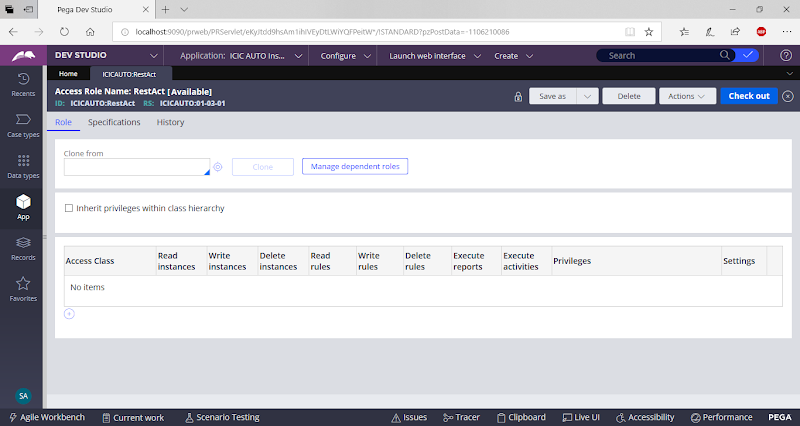
**Create Access Role Name:**



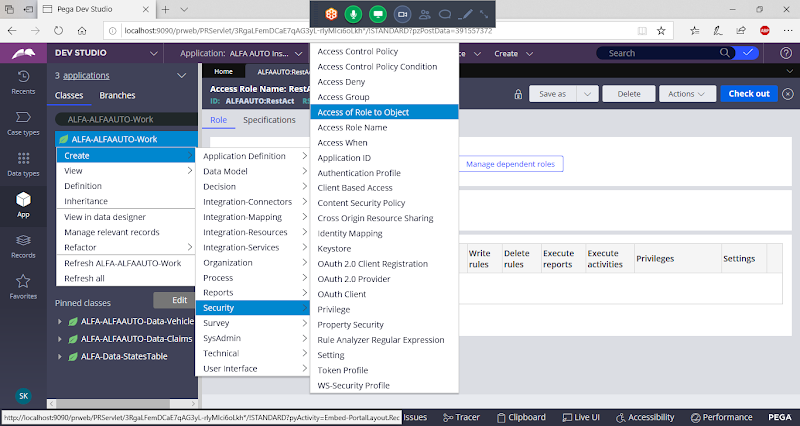


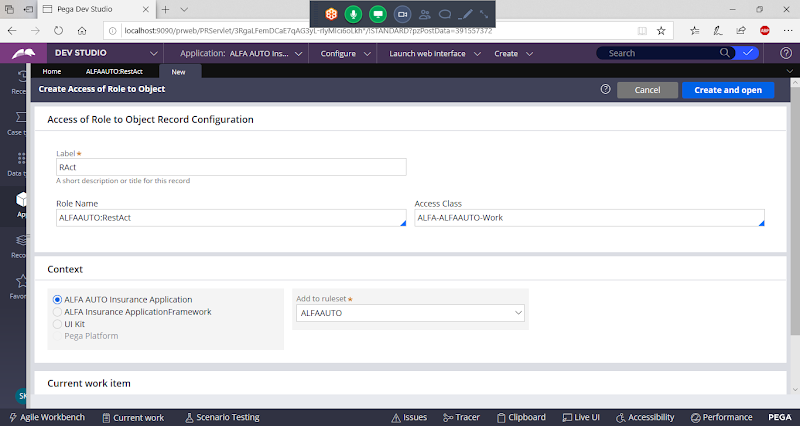


**Save the Rule**



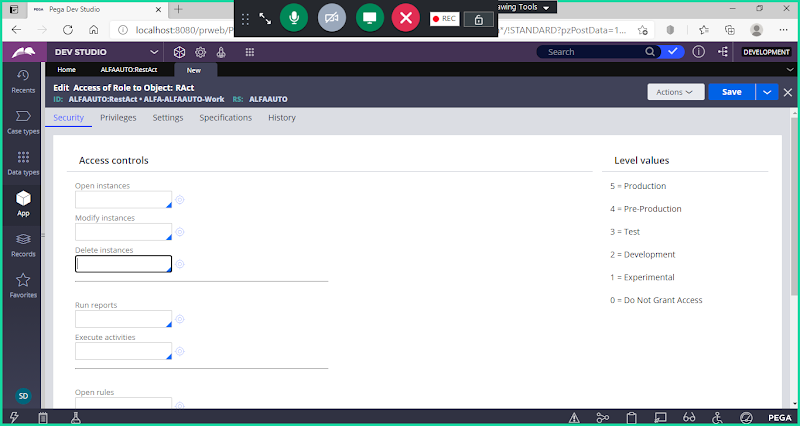
**Now Create Access of Role to Object:**

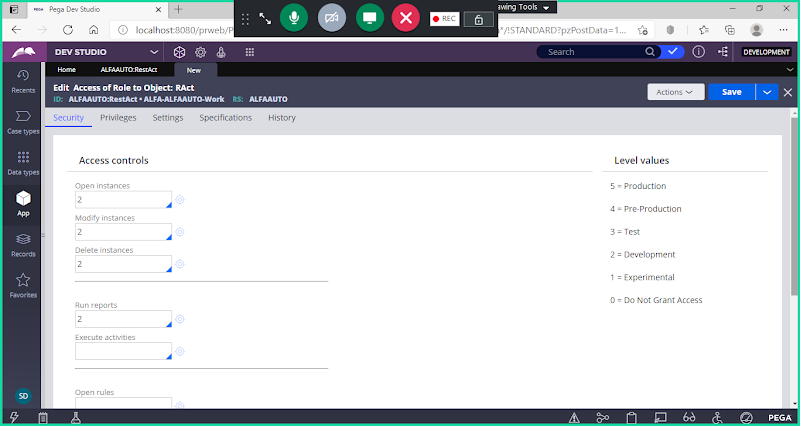


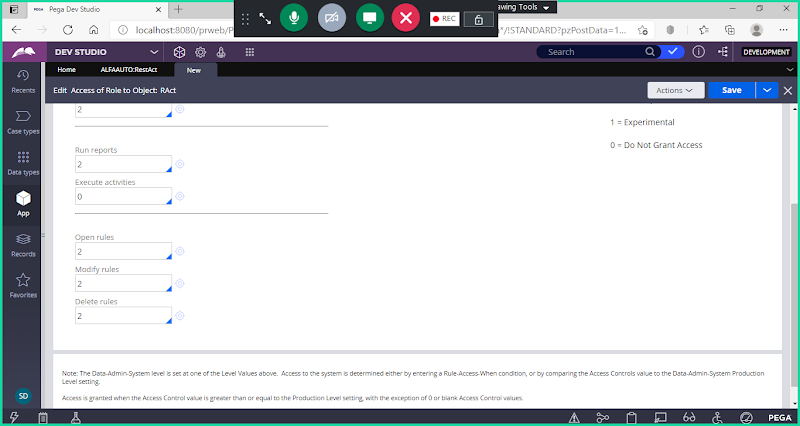


**Add Access Class**

**Open**



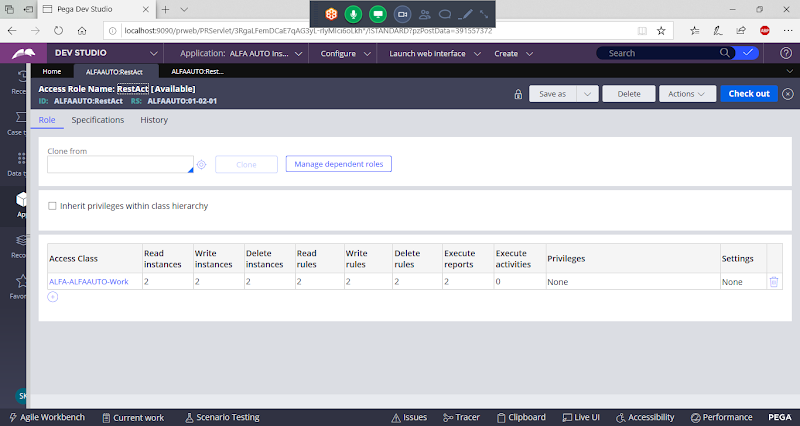




**Save**

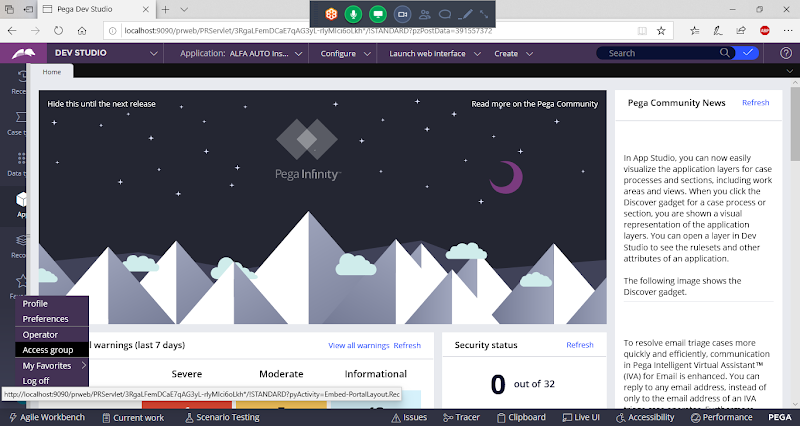
**Now open Access Role Name & Refresh**

**Refersh**

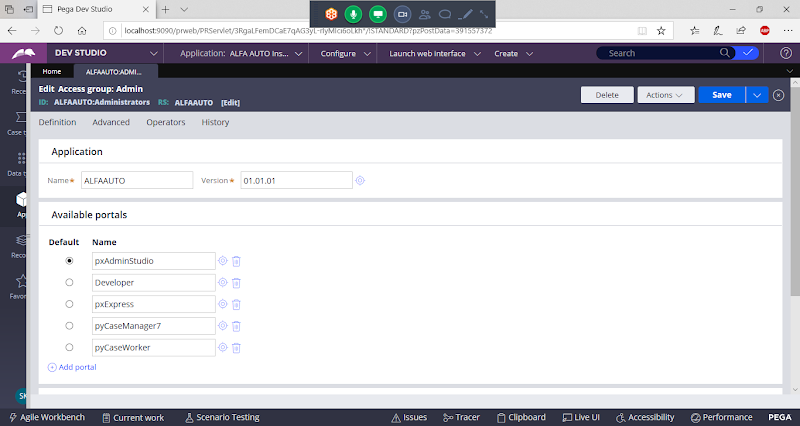


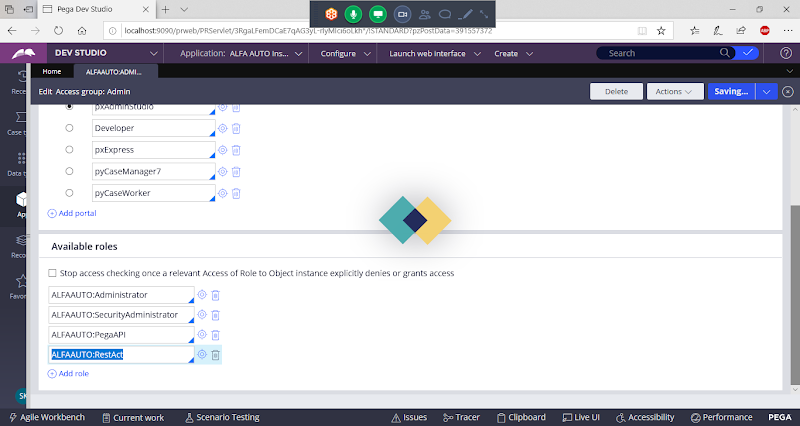
**===**

**Now Open Access Group Rule Form:**



**Under Available Roles Add New Access Role Name:**



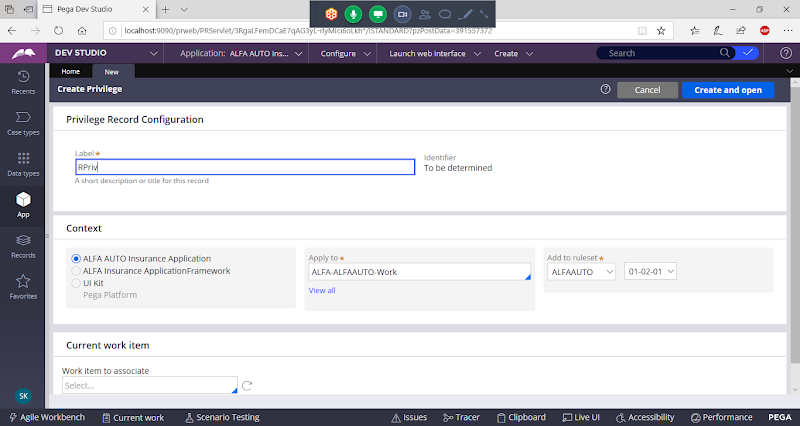


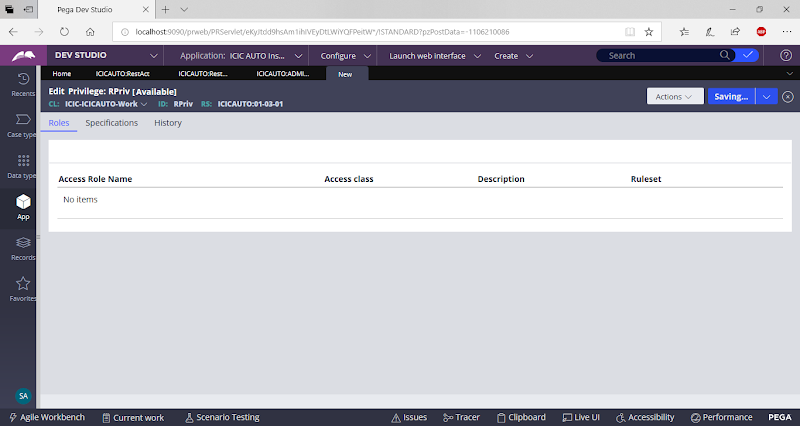
**Save Access Group:**

**===============**

**Add Privilege Rule:**

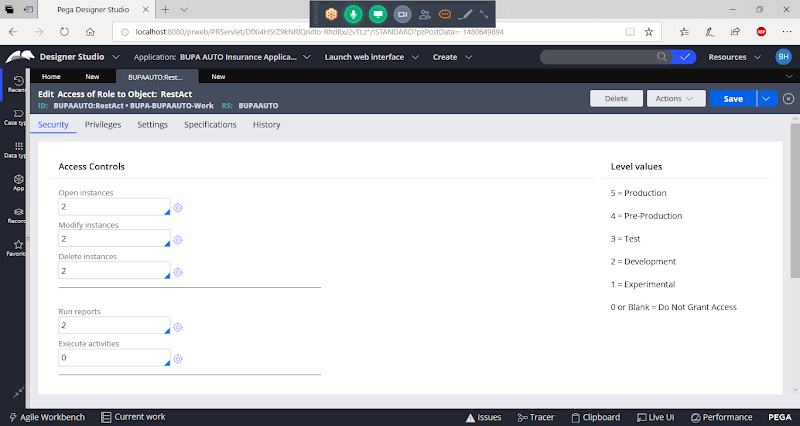




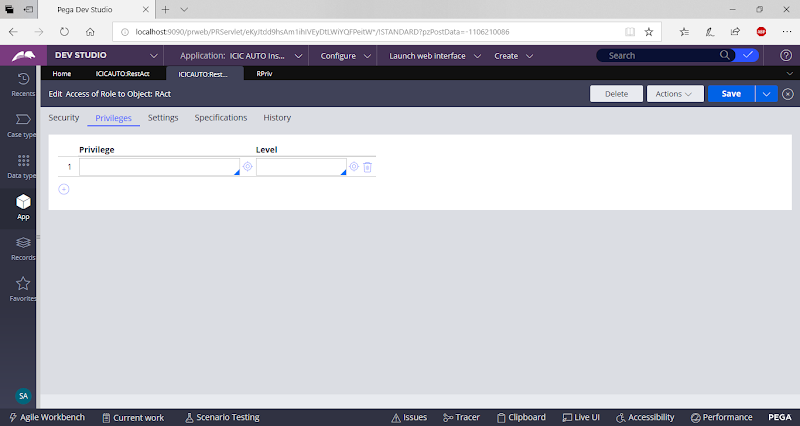


**Save Privilege**

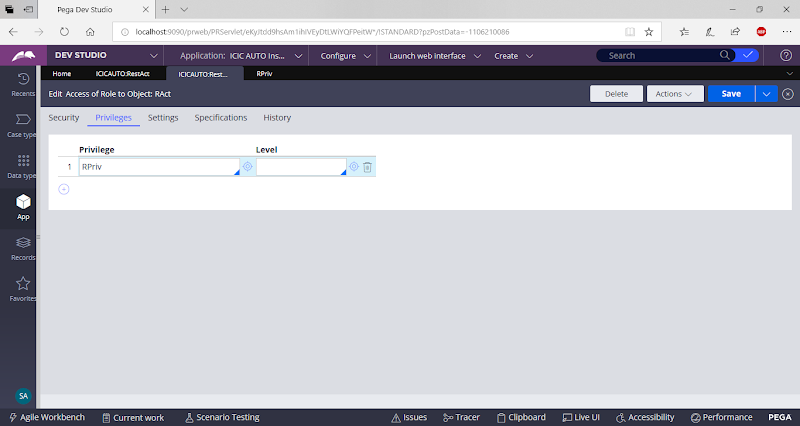
**Now Open Access of Role to Object:**



**Go to Privileges**

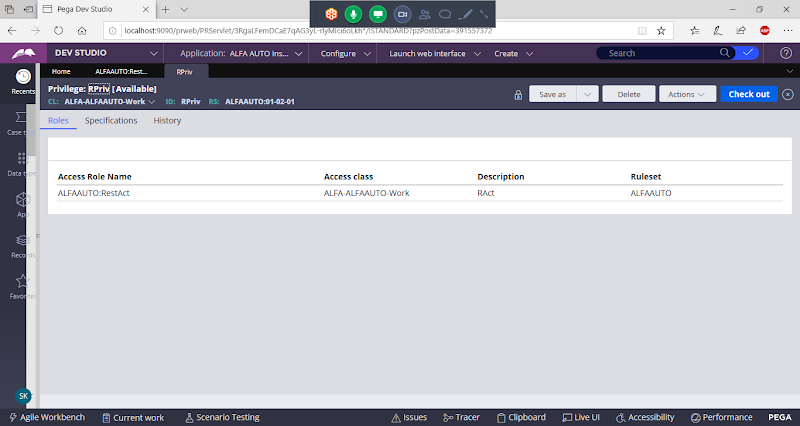


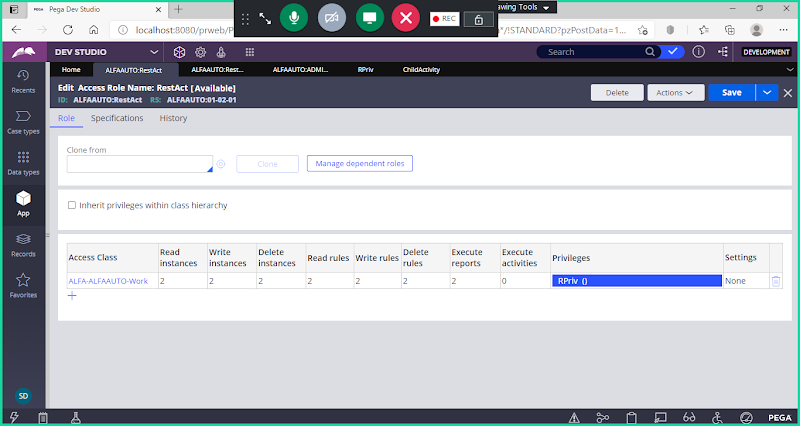
**Call Privilege Rule Name:**



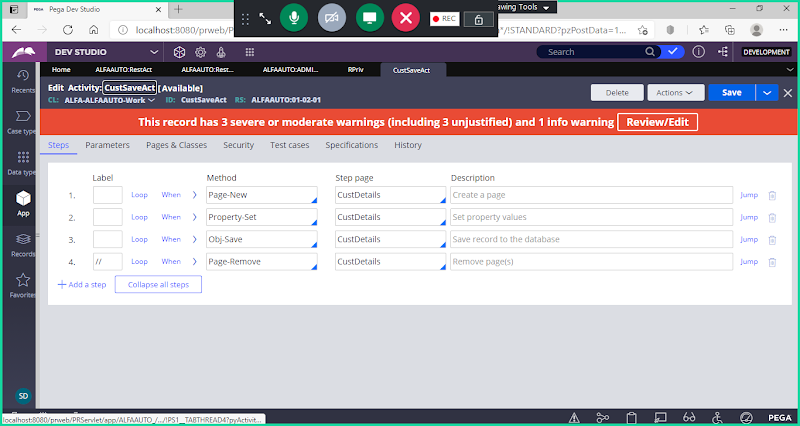
**Save Access of Role to Object**

**Close & Open Privilege Rule from Recent Items/ Logout & Login to open Privilege Rule:**

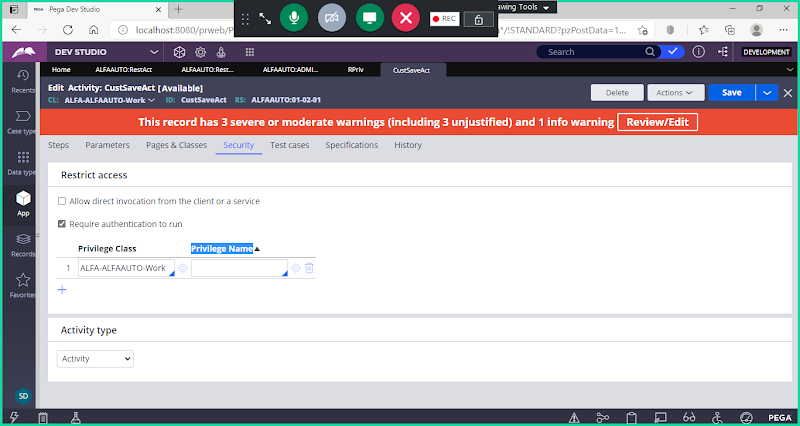


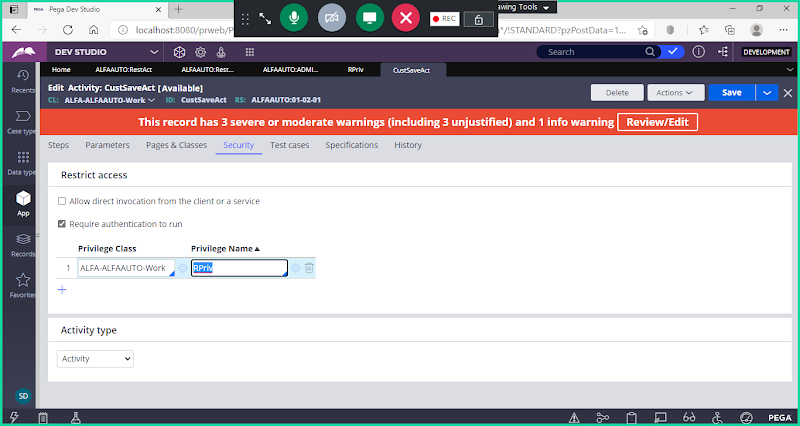


**Open Any Activity in the Application :**



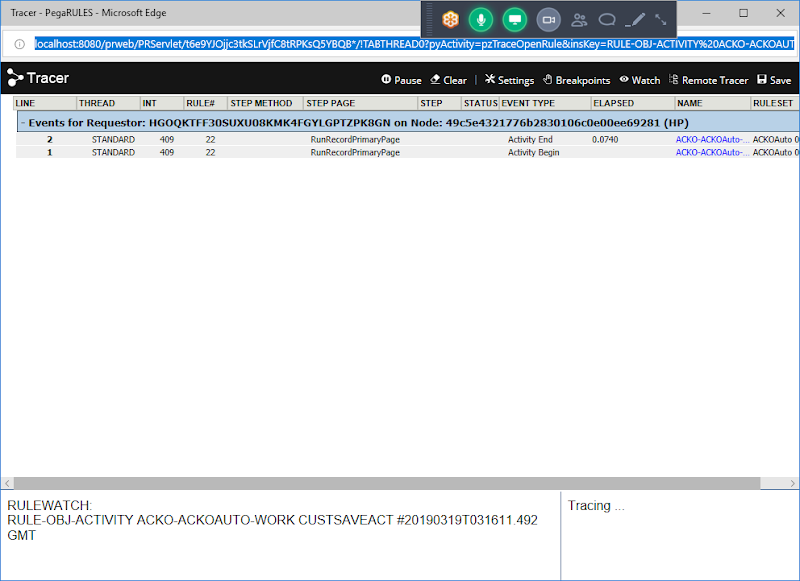
**Go to Security: Call Privilege**





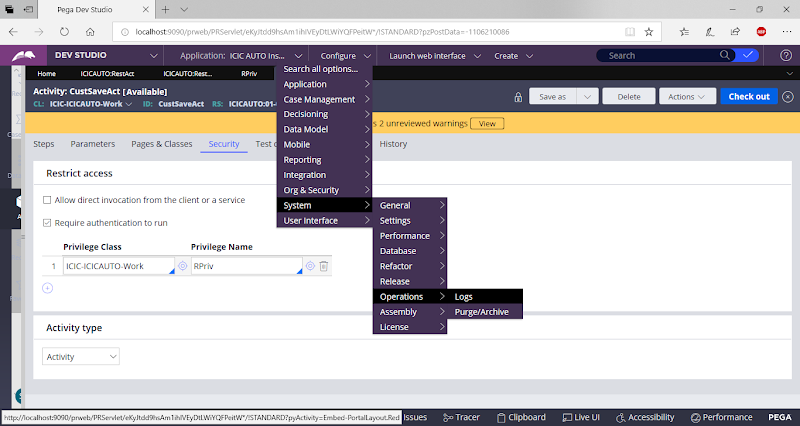
**Save & Check in**

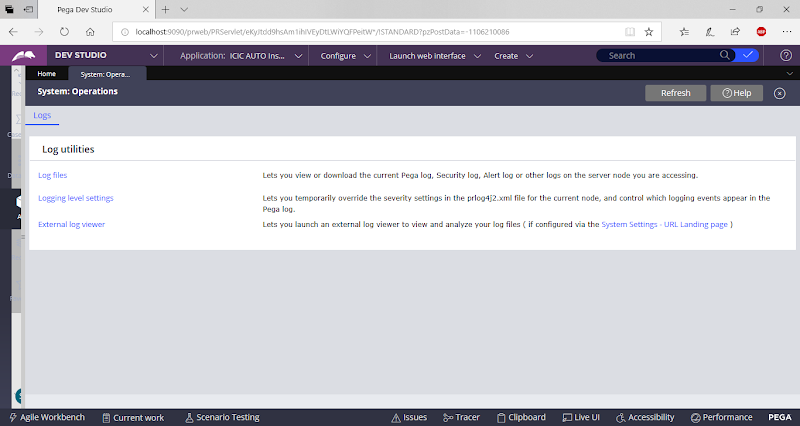
**Run Activity & Trace:**



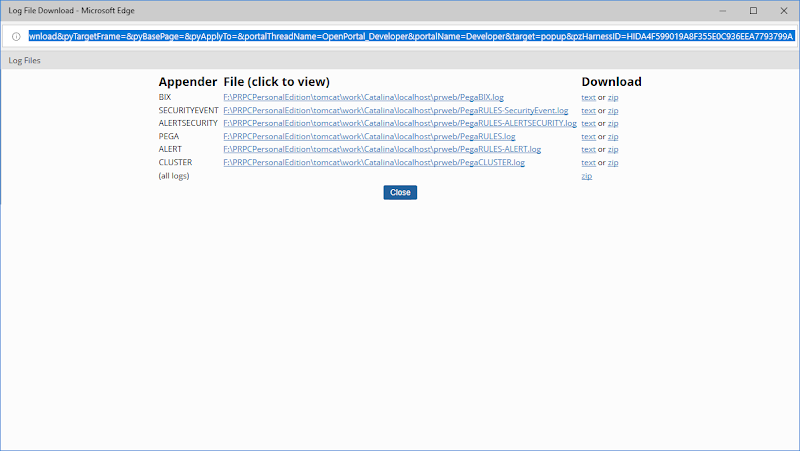
**It will not read activity:**

**Check the logs:**

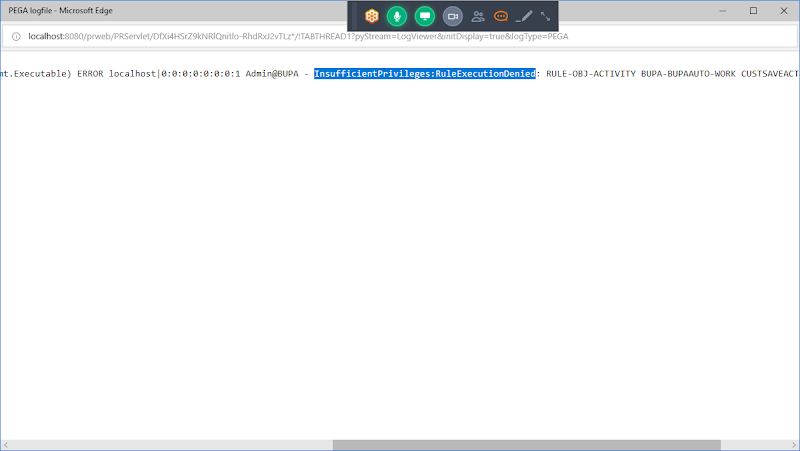




**Open Log Files**



**Click on pega**



**Access of Role to Object:  Permissions are set Access group Level**

**Privilege: Permissions are set at Independent Rule level for Activities & Report Definition**

**AccessWhen: Using this they can set permissions for Random Environments ( Ex: 2 & 4)**

**AccessDeny: Using this they can Deny permissions for Random Environments ( Ex: 1 & 3)**

**================**

**Types of Rulesets:**

**Application Rulesets**

**Personal Rulesets**

**Production Rulesets**

**Branch Rulestes**

**Application Rulesets: ALFAAUTO**

**Application Rulesets are Created based on Application Setup:**

**By Default all the rules are added under application rulesets.**

**Personal Rulesets**

**For Every Operator/Developer the user id he is using to Login to Application, i.e. his Personal Ruleset.**

**Based on Operator ID personal ruleset will be different for every operator**

**My User ID is:  Admin@ALFA ( This is My Personal Ruleset)**

**Developer2: ABC1 ( This is his Personal Ruleset)**

**Developer3:  ABC2 ( This is his Personal Ruleset)**

**- The Checkout Rules are available in Personal Rulesets of an Operator.**

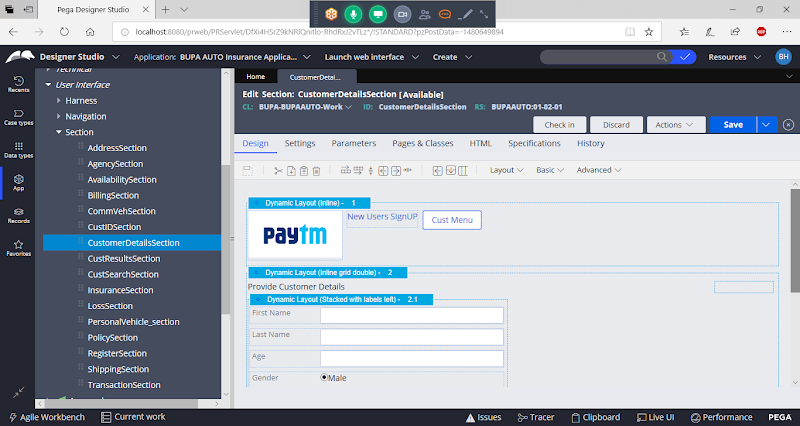
**-Personal Rulesets are not Physical Rulesets, it is a Virtual Rulesets created for every operator with the name of Operator ID**

**Admin@ALFA -  Checkout**

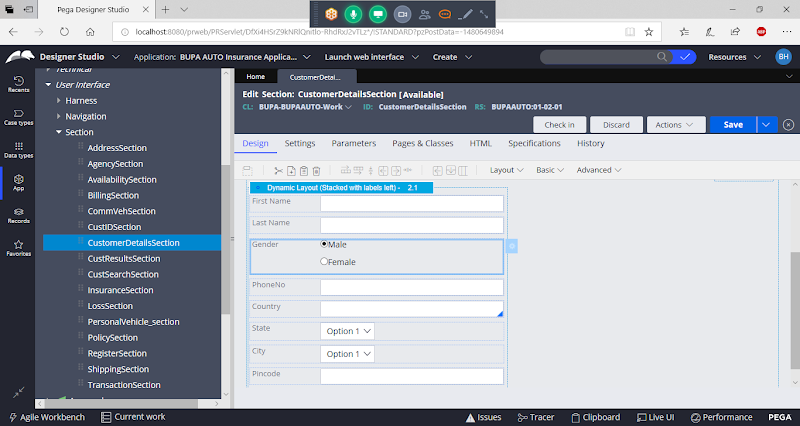
**Understanding Personal Ruleset:**

**Open Customer Section: ( 01-02-01)**

**Delete Some Properties: To Delete any Fileds Click on Checkout**

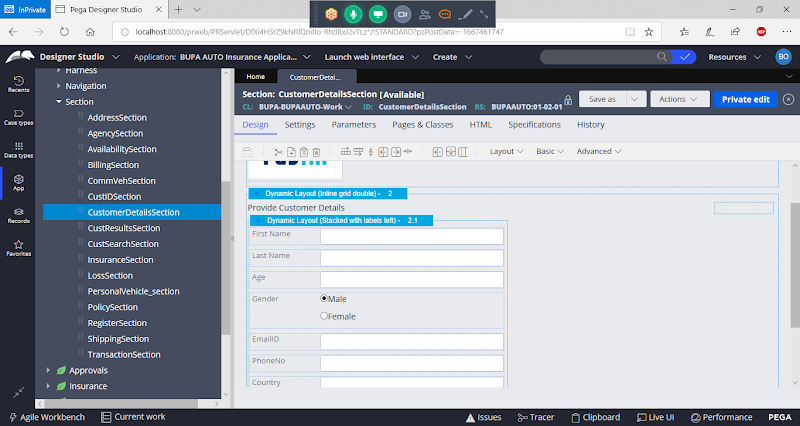


**Delete Age & Email ID:**



**Save the Rule ( But don't Check in)**

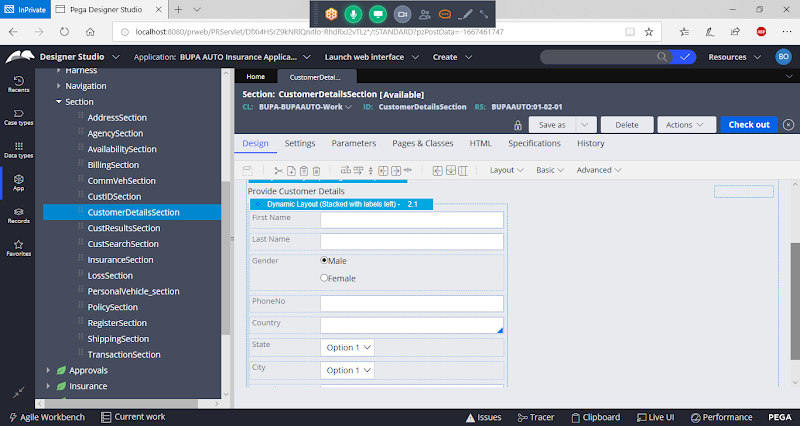
**Now Open Another Operator in another session: ALFA1**



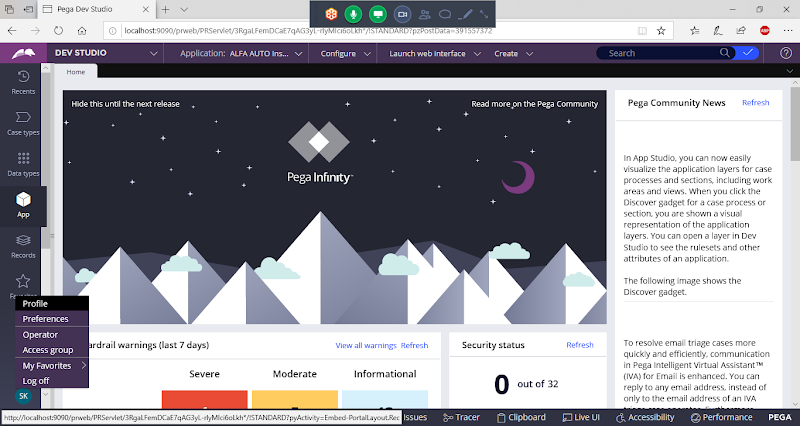
**In ALFA1  it will show Age & email ID.**

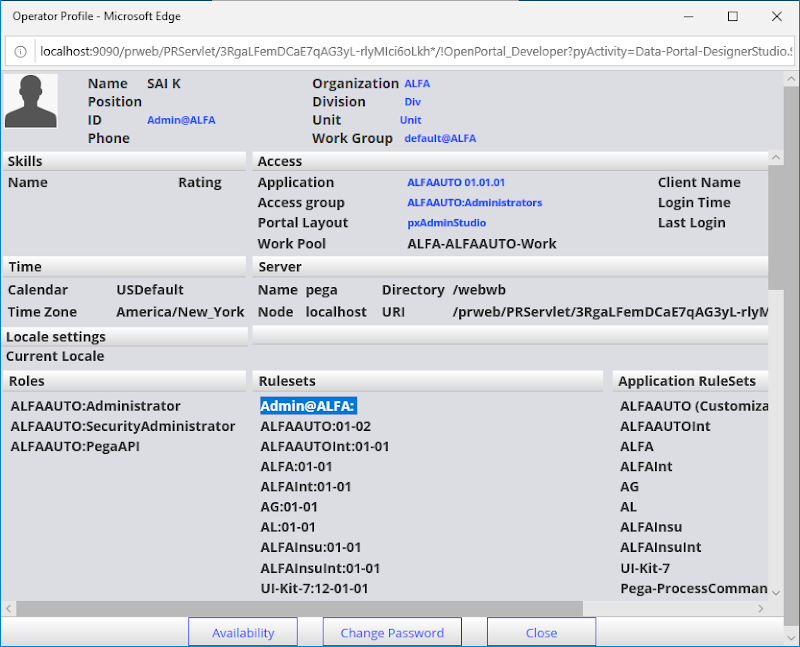
**because the changes has been saved only in Personal Ruleset, after doing changes we have to click on Check in to reflect these changes in other opeartor Id's.**

**After Check in , Changes will show in other Operator ID:**



**- Ruleset Hierarchy we can see under Operator Profile:**





**=====**

**Production Ruleset:**

**Customization of rules allowed in live environment then we can create such rules in Production Ruleset.**

**Production Rulesets must be left unlocked, so the rules available under production ruleset can be customized.**

**Delegate the Rule/Rule Delegation:**

**Ex: Update Decision table every day which deals with price updating in day to day business.**

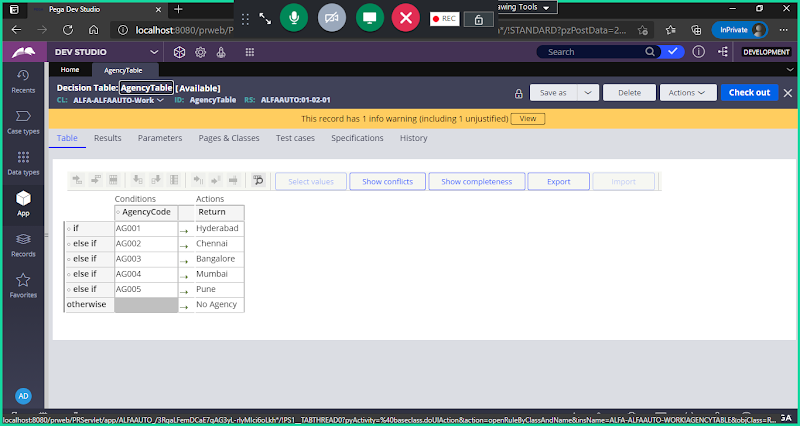
**- Any Rules created under Production Rulesets, such type of rules can be customized in live env. we can left the Ruleset Unlock.**

**- If we lock production rulesets, then customization is not possible for the Rules which are created under Production Rulesets.**

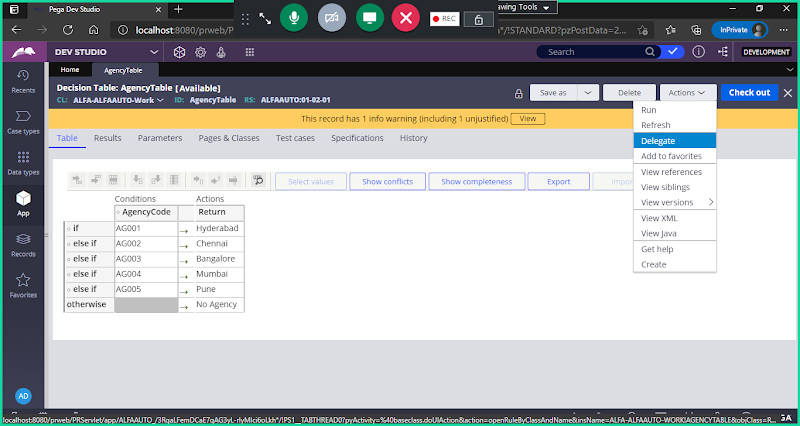
**Rule Delegation: Using this we can make the rule available for modification for the business.  The concept is called Rule-Delegation.**

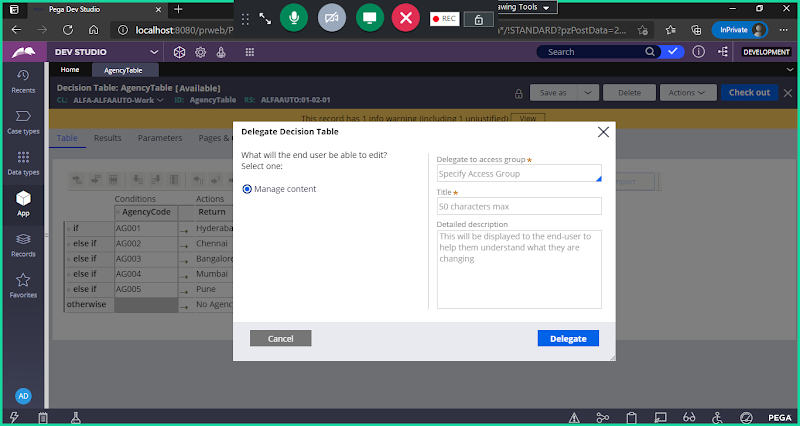
**They can open the Rule & Modify the Rule.**

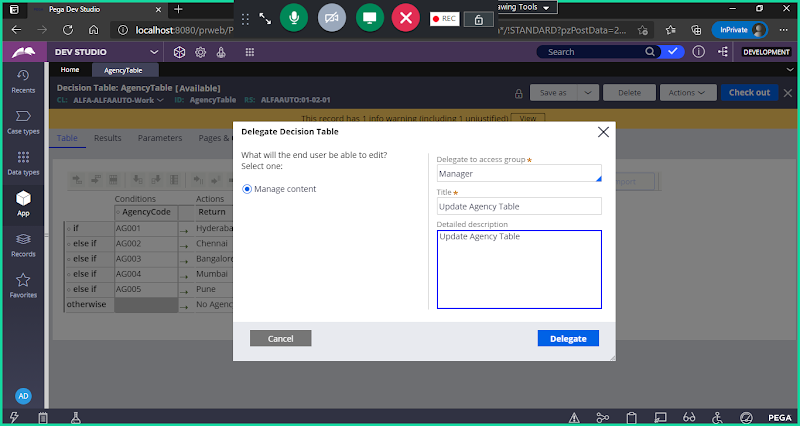
**Rule Delegation can be done to an Access Group.**



**Go to Actions Dropdown & Select Delegate**



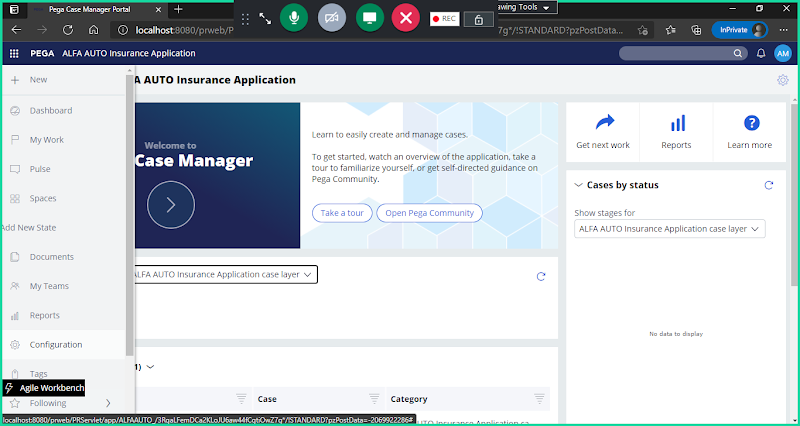




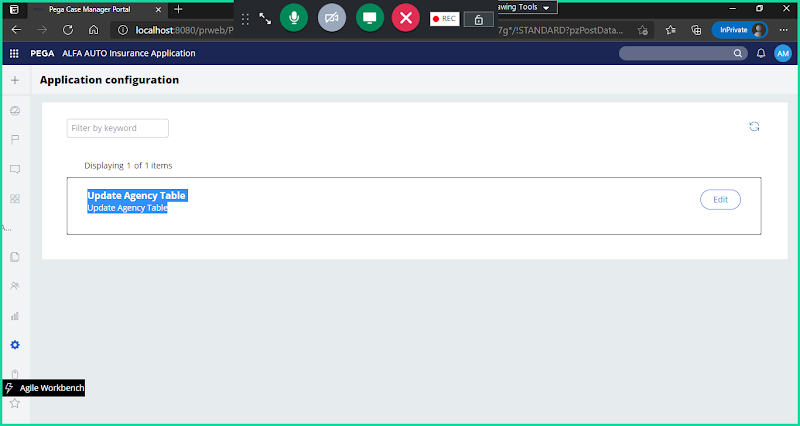
**Click on Delegate**

**Logoff & Login to  Manager Portal**

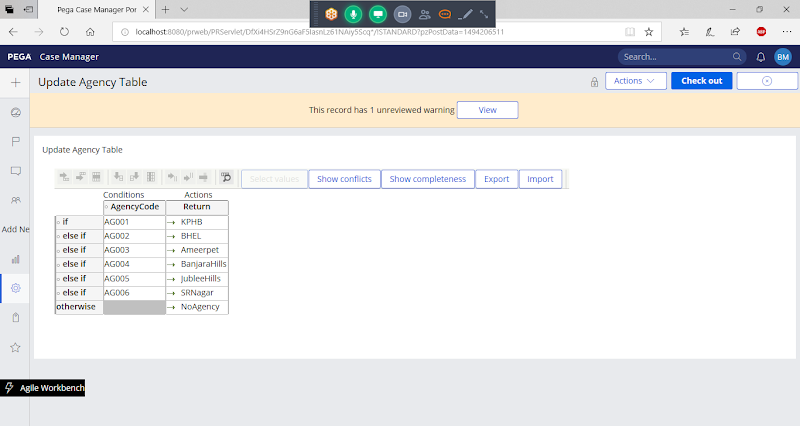
**Login Using: ALFAM1 , Password : rules**



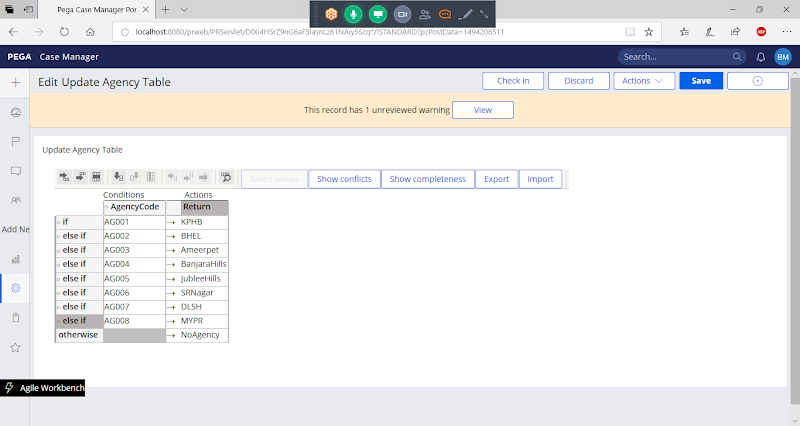
**Click on Configuration:**



**Click on edit**



**Click on Check Out**

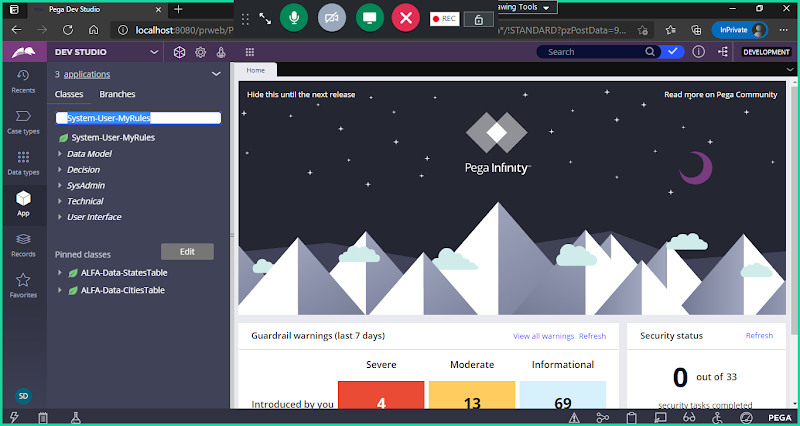


**Save & Check in**

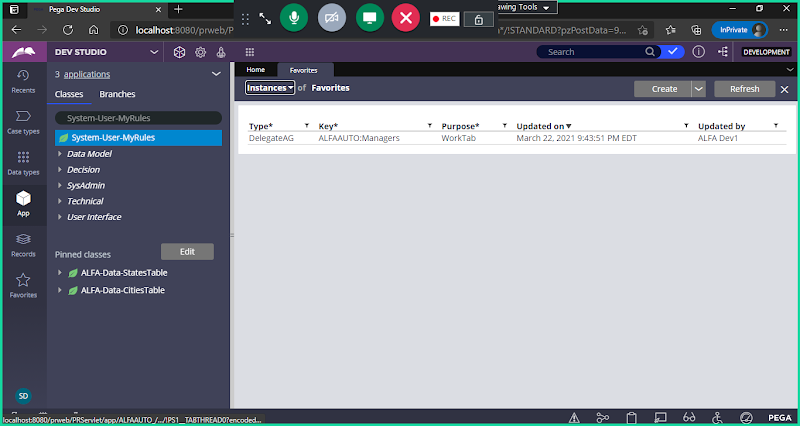
**Now Go to Developer Portal**

**System-User-MyRules:**

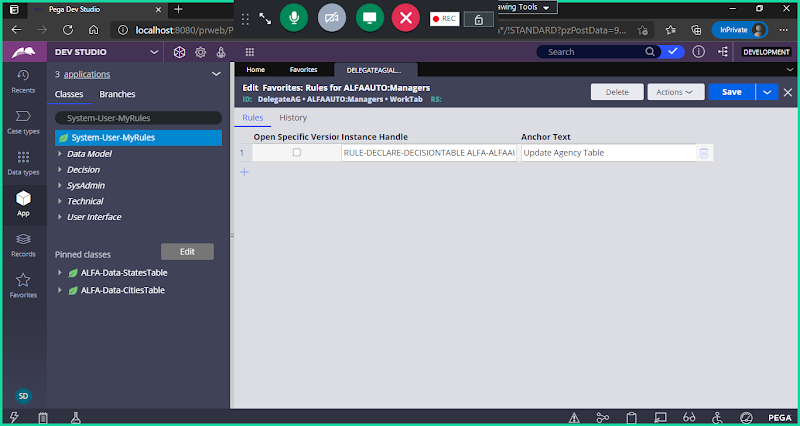
**All the Delegated rules are available under System-User-MyRules Class:**



**Click on Class:**



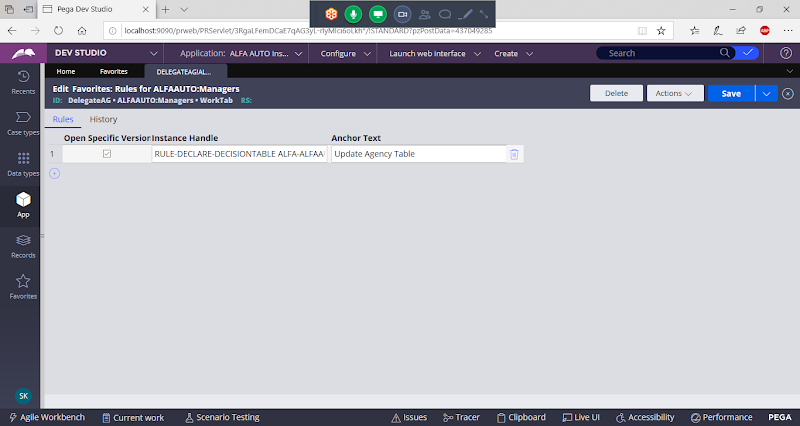
**Open the instance**



**Click on Check Box:**

**If we Select the Check box: It will pick the highest Version**

**If we don't select the Check box: Irrespective of the version, It will pick only the Delegated Rule.**



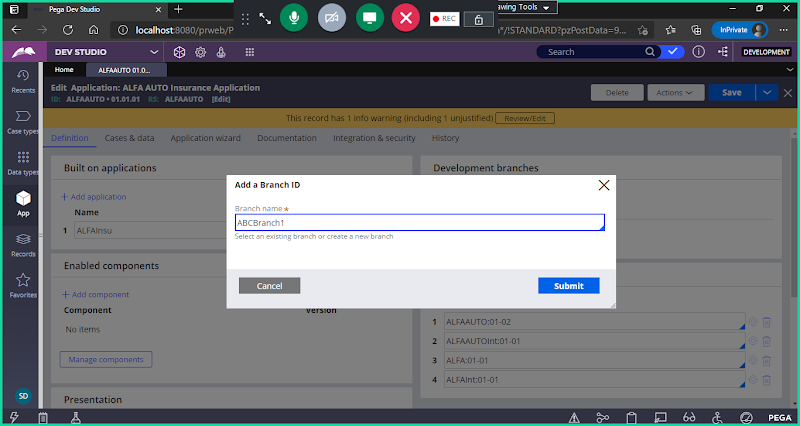
**============**

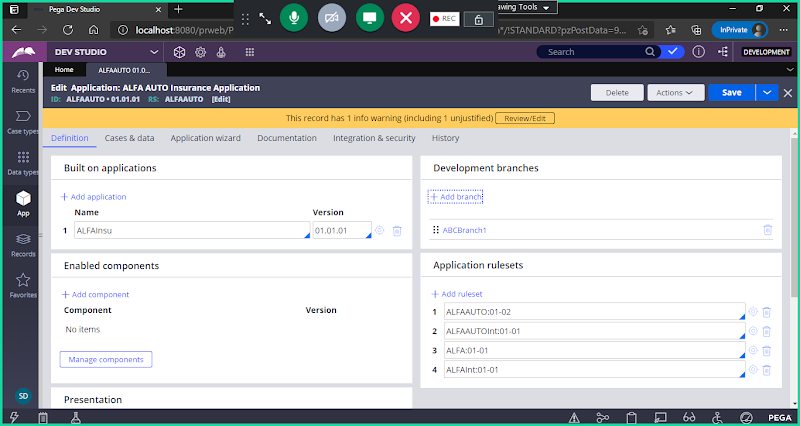
**Branch Rulesets:**



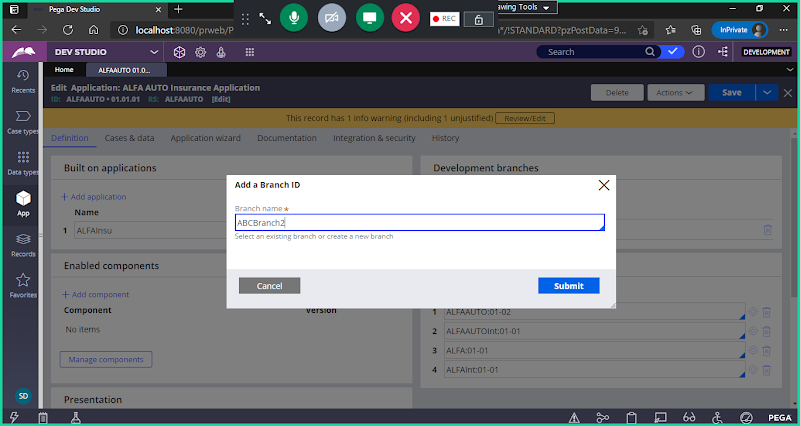
**if 2 developers want to work on the same rule, then they can use Branch Rulesets**

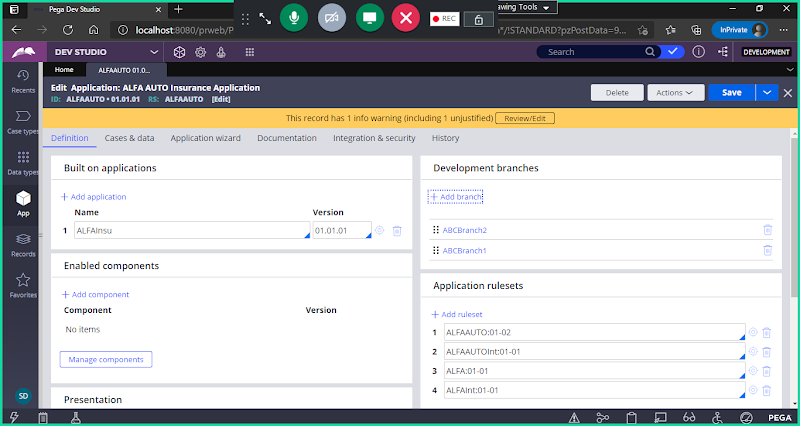
**Add Branches:**





**Add Branch2**

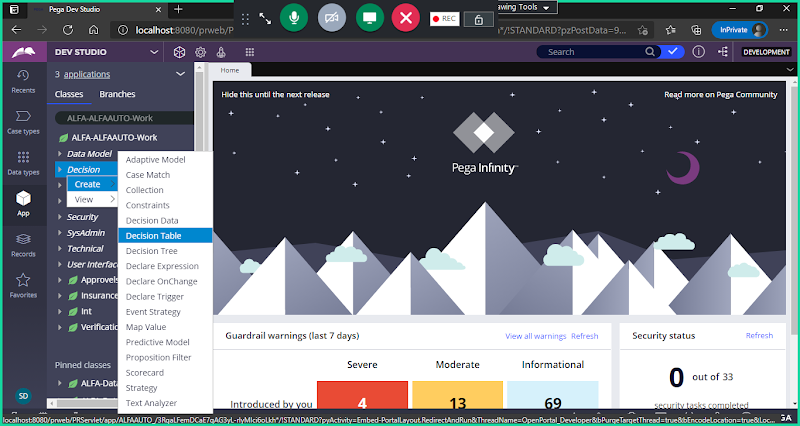


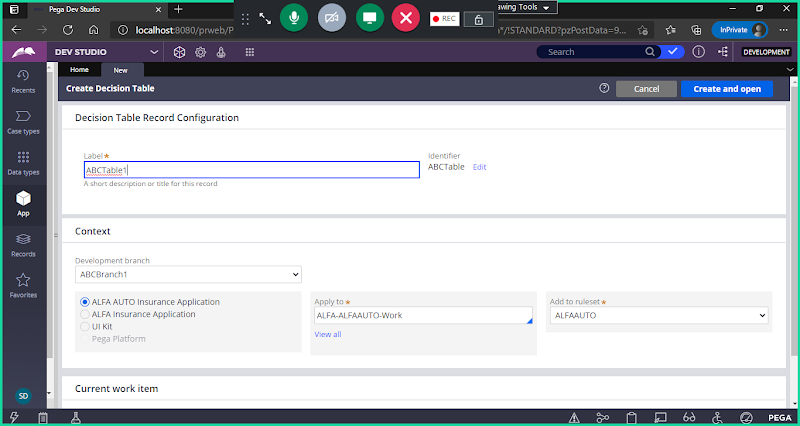


**If we have to work on same rules, we cannot wait for the other developer to complete the development, to complete work on time some times we will be working under branch rulesets.**

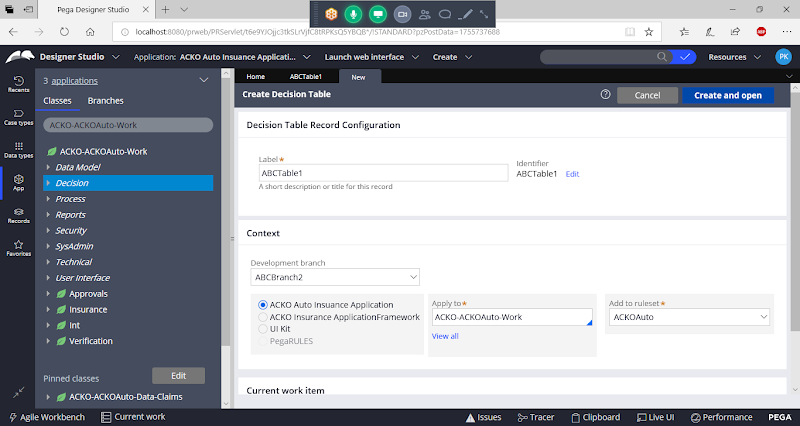
**- It is the Process of Parallel Development.**

**Craete a Rule:**



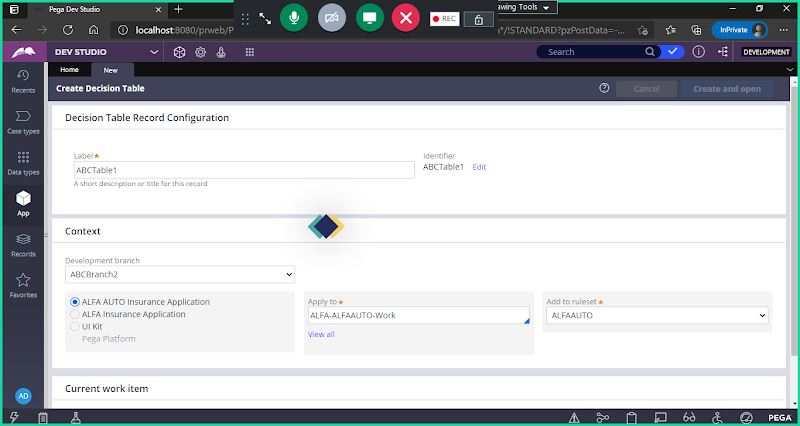


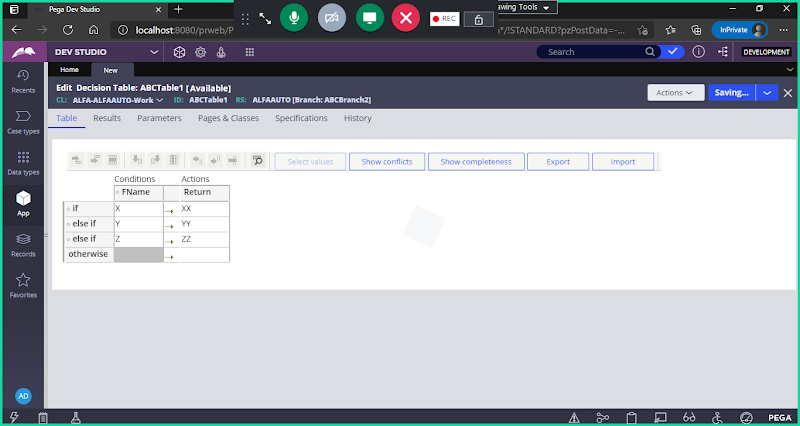
**Now Give Same Name to Decision table & Craete Under Branch2:**





**Now Create another Table with Same Name under Branch2 ( Create From ALFA1 User ID)**

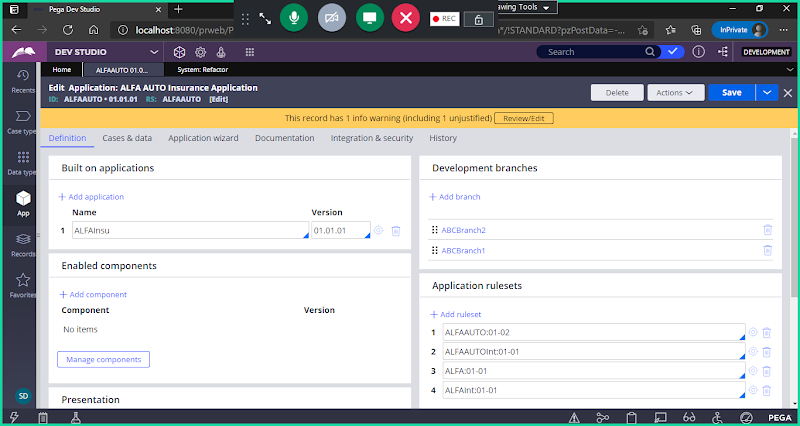




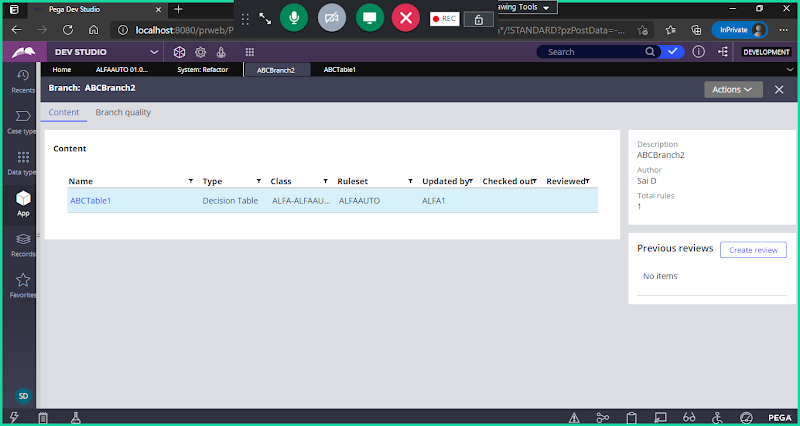
**Now Same Rule Created under 2 Branches by diff operators.**

**Now They have to Merge the Rule.**

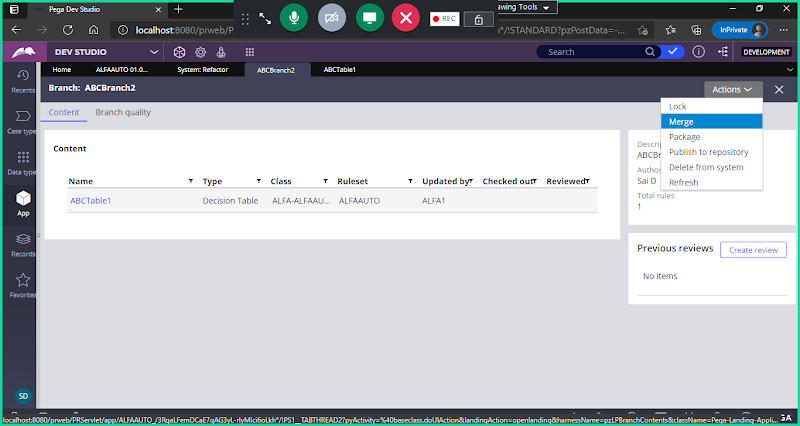
**Open Application definition:**

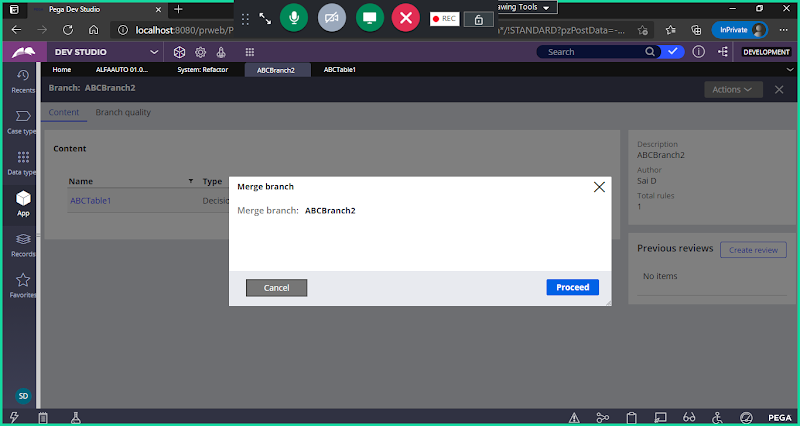


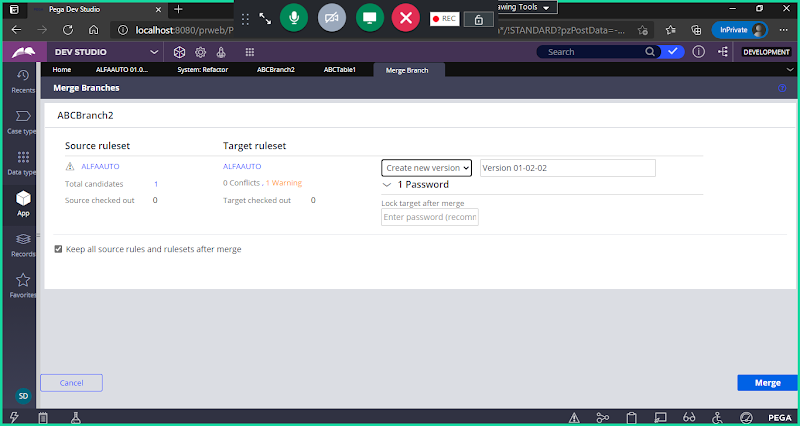
**Open Branch2:**

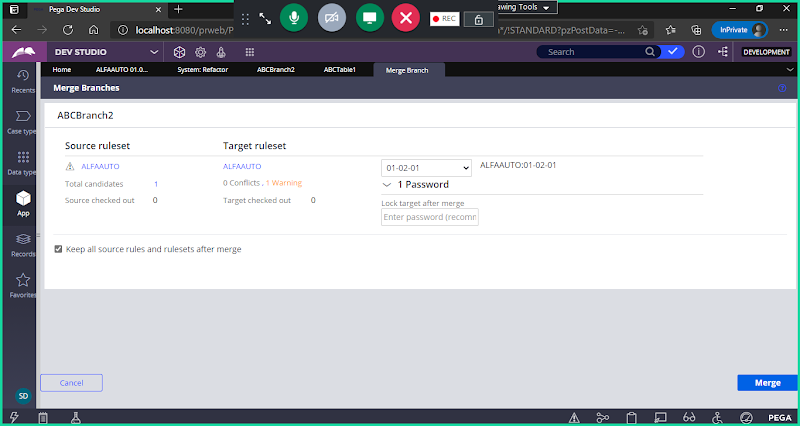


**Go to Actions -> Merge**









**Click on Merge:**

