

Developer Certification Preparation – Mini Project

Part I – ADE Practice

These exercises are designed to help you prepare for IBM Cúram SPM V7.X Application Developer certification. The exercises require you to implement many of the features that you will be asked about in the certification test. However, it does not implement all the features that you need to know for the test. For example, widgets or page logic is not covered.

Make sure that you review the certification objectives when preparing for the test:

[Cúram \(SPM\) Certification Program](#)

The exercise instructions are short so you must think about how to implement them. Note that there may be several ways to implement them, so solutions are not provided.

Prerequisites

During *Merative Cúram SPM for Developers (ADE) - 9D73G* you implemented features for the Person Demo application. Exercise 20 for 9D73G extends the Person Demo application with a simple Case feature. I'd recommend completing exercise 20 because it makes you review core concepts. You must load the Person Demo application (solution for exercise 19) before starting exercise 20. The Person Demo and case study (exercise 20) must be working before you can tackle the additional features below. All features should be implemented in the CourseWorkshop. For simplicity, it is best to use the development environment that you used in the developer training. If the environment that you used for 9D73G and 9D74G training has expired, you can register for 9D77G - *Merative Cúram Development Environment 7.X: Sandbox*.

Original Case Study (Exercise 20)

Once a Person makes a claim for a program, a case is created for them on the system to store the claim details. Allow for the following types of claims:

- Unemployment Assistance
- Cash Assistance

- Food Stamps
- Disability
- Training

Add features to your Person Demo application to allow multiple cases to be stored for each Person and displayed in a list page. The user should be able to create and list the cases.

Design the necessary entity and process classes to provide the functionality. When considering where this information will be presented, you should replace the Dummy page referenced by the Dummy tab that you set up in the Configure Application Navigation exercise.

Additional Features

Introduce the features one at a time so that you have a working minimal viable product (MVP) after each step. Pay attention to the design of the feature so that you know how it will be used by the end-user and which artifacts you need to modify to implement the feature. Note that the features are simplified for educational reasons. Therefore, the suggestions of where to locate them does not always represent the best way to implement features in SPM Cúram.

1. Generate a unique id for the caseID. The caseID should be a human readable key from its own keyset. You can generate the caseID by using an exit point.
2. Create a separate RSAD fragment for the Person and Case parts of the model.
3. The generated and handcrafted code for the case feature should be in its own code package called **cases**.
4. Add a foreign key relationship between the Person and Case tables.
5. Model an index to speed up accessing case records by person.
6. Add the following attributes for the case: dateRegistered and comments. Allow the user to enter values for these attributes when they create the case. Display the attribute values when cases are listed.
7. When adding a case, force the user to select a caseType from the list of cases rather than defaulting to the first case in the list. That is, the first caseType they see should be blank.
8. Make the caseType and dateRegistered mandatory input fields.

9. Add an attribute to the Case table called caseStatus. caseStatus can have a value of Pending, Approved, and Closed. Set the status to Pending when a case is added. Ensure that the case status is displayed cases are listed.
10. Create test data so that you do not have to add names and cases manually after you build the database. Create test data in the PERSON and CASE tables and extract them to DMX files. Ensure that the primary keys of the test data do not conflict with generated keys.
11. Add validation to prevent users from adding two cases of the same caseType unless the case is Closed. The message should say: (<firstName>) already has an open case of this type.
12. From the list of cases, allow a user to select a caseID to open the case in separate tab called Case. The case home page should show all the case details.
13. Provide a context panel for the Case tab. The context panel should show the following: case type, concern role ID, and the person's full name. The concern role id should display as a hyperlink to allow the user to link to the person page. The stick out tab for the case should display "Case – 999", where 999 is the case number.
14. Allow the user to approve a case from the Case Tab Actions menu. For simplicity, you can ignore the case status. A modal dialog should appear to ask the user if they are sure they want to approve the case. If the user answers Yes, the status is set to Approved.
15. If the client has a closed case of the same type as is been added, display the information message "(<name>) had a case of this type before". However, allow the case to be added.
16. Allow the user to list all cases that have been Pending from a specified date. The user enters a cut-off date. The report lists cases with a status of Pending and dateRegistered on or before the cut-off date. The report should show the concernRoleID, person name, case id, case type, status, and date registered. List in order of person and dateRegistered. Provide hyperlinks for concern role Id and case ID. Access the report from the shortcuts panel.
17. Add a new Section called Reports with its own shortcuts panel. Display the Aged Cases Report as the default page in this section. Move the report from the default shortcuts panel to the shortcuts panel in the Reports section.

If you want more features...

There are an endless number of features you could add to practice using the Cúram ADE, for example:

- Allow users to modify case information, such as comments and dateRegistered.
- Add a JUnit test for inserting and reading a case.
- Add a feature to close all cases for a person. Select an appropriate entity operation.
- Add help for a page.
- Make the list of cases expandable. Display dateRegistered and comment in the inline page.