Story 1 - Validate, if invoice is expired
Description
For any invoice, the system shall validate if based on its maturity the invoice must be expired.

Actor
CRX System

Pre-Condition
User uploads an invoice.

Functional Requirements
On invoice upload the system shall validate invoice expiration as follows:
- Call synchronous system interface of Holidays service getTargetDate with the next input parameters:
  - startDate = Invoice.maturityDate,
  - numberOfWorkingDays = delay,
  - direction = „BACKWARD“
  - type = „EUR“
- Check if returned target date >= Today.
  - If yes – invoice is uploaded with status = UPLOADED
  - Else – invoice is uploaded with status = EXPIRED

Test Plan Name
Invoice Status Validation upon upload of an Invoice based on the Invoice Maturity Date.

- Scenario 1: Verify the Invoice Upload Status during upload process when **Invoice Maturity date is set 5 days ahead in the future with delay 3 and no holidays or weekend in the delay period in EUR so that target date should appear as same date**

  GIVEN Pre-Condition:
  User uploads an invoice on 3 Jul 2023 (or current date considering it a Monday)

  Setup and Validate:
  1. Invoice.maturity Date is set to 7 Jul 2023 (5 days ahead of current date considering it a Friday)
  2. Delay is set to 3
  3. Direction is set to ‘BACKWARD’
  4. Type is set to EUR

Expected Result: Application calls the synchronous system interface of Holidays service getTargetDate.
The returned target date is 4 Jul 2023 which is greater than Current Date. Invoice is uploaded with status = UPLOADED

- Scenario 2: Verify the Invoice Upload Status during upload process when **Invoice Maturity date is set 5 days ahead in the future with delay of 4 days and no holidays or weekend in delay period in EUR so that target date should appear as same date**

  GIVEN Pre-Condition:
  User uploads an invoice on 3 Jul 2023 (or current date considering it a Monday)

  Setup and Validate:
  1. Invoice.maturity Date is set to 7 Jul 2023 (5 days ahead of current date considering it a Friday)
  2. Delay is set to 4
  3. Direction is set to ‘BACKWARD’
  4. Type is set to EUR
  5. Trigger a call to synchronous system interface of Holidays service getTargetDate

Expected Result: Application calls the synchronous system interface of Holidays service getTargetDate.
The returned target date is same as Current Date. Invoice is uploaded with status = UPLOADED

- Scenario 3: Verify the Invoice Upload Status during upload process when **Invoice Maturity date is set 5 days ahead in the future with delay of 5 days and no holidays in delay period in EUR so that target date should appear as past date**

  GIVEN Pre-Condition:
  User uploads an invoice on 3 Jul 2023 (or current date considering it a Monday)

  Setup and Validate:
  1. Invoice.maturity Date is set to 7 Jul 2023 (5 days ahead of current date considering it a Friday)
  2. Delay is set to 5
  3. Direction is set to 'BACKWARD'
  4. Type is set to EUR
  5. Trigger a call to synchronous system interface of Holidays service getTargetDate

Expected Result: Application calls the synchronous system interface of Holidays service getTargetDate.
The returned target date is 30 Jun 2023 which is in past and less than Current Date that is 3 Jul 2023. Invoice is uploaded with status = EXPIRED

- Scenario 4: Verify the Invoice Upload Status during upload process when I**nvoice Maturity date is set 6 days ahead in the future with delay of 5 days and 2 holidays in delay period in EUR so that target date should appear as past date**

  GIVEN Pre-Condition:
  Change system date to 22 Dec 2023
  User uploads an invoice on 22 Dec 2023 (or current date considering it a Friday)
  holiday.type. "EUR" should have holiday.date as 2023-12-25 in Holiday Service
  holiday.type. "EUR" should have holiday.date as 2023-12-26 in Holiday Service
  Setup and Validate:
  1. Invoice.maturity Date is set to 29 Dec 2023 (7 days ahead of current date considering it a Friday)
  2. Delay is set to 5 (considering the Monday and Tuesday holidays in EUR)
  3. Direction is set to 'BACKWARD'
  4. Type is set to EUR
  5. Trigger a call to synchronous system interface of Holidays service getTargetDate

Expected Result: Application calls the synchronous system interface of Holidays service getTargetDate.
The returned target date is 20 Dec 2023 which would be in the past and less than Current Date that is 22 Dec 2023. Invoice is uploaded with status = EXPIRED

- Scenario 5: Verify the Invoice Upload Status during upload process when **Invoice Maturity date is set in current date with delay 0 and no holidays in delay period in EUR**

  GIVEN Pre-Condition:
  User uploads an invoice on 3 Jul 2023 (or current date considering it a Monday)

  Setup and Validate:

1. Invoice.maturity Date is set to 3 Jul 2023 (which is current date)
2. Delay is set to 0
3. Direction is set to ‘BACKWARD’
4. Type is set to EUR

Expected Result: Application calls the synchronous system interface of Holidays service getTargetDate.
The returned target date is Current Date. Invoice is uploaded with status = UPLOADED

Note:
The possibilities of the following 2 scenarios are not explained in the Acceptance Criteria of the story, however, I have considered it for my boundary value analysis.

- Scenario 6: Verify the Invoice Upload Status during upload process when **Invoice Maturity date is not set**

  GIVEN Pre-Condition:
  User uploads an invoice on 3 Jul 2023 (or current date considering it a Monday)

  Setup and Validate:
  1. Invoice.maturity Date is not set
  2. Delay is set to 0
  3. Direction is set to ‘BACKWARD’
  4. Type is set to EUR

Expected Result: Application calls the synchronous system interface of Holidays service getTargetDate.
A mandatory field check of Invoice Maturity Date should be performed and error should be displayed for missing data

- Scenario 7: Verify the Invoice Upload Status during upload process when **Invoice Maturity date is set in the past**

  GIVEN Pre-Condition:
  User uploads an invoice on 3 Jul 2023 (or current date considering it a Monday)

  Setup and Validate:
  1. Invoice.maturity Date is set to 28 June 2023
  2. Delay is set to 0
  3. Direction is set to ‘BACKWARD’
  4. Type is set to EUR

Expected Result: Application calls the synchronous system interface of Holidays service getTargetDate.
The returned target date is less than Current Date. Invoice is uploaded with status = EXPIRED

Also, considering the following as a potential scenario if, CRX marketplace operations are carried out outside European timezone

- Verify the Invoice Upload Status during upload process when **Invoice Maturity date is set for a date with delay 0 and no holidays in delay period in EUR from a timezone that is already on next date but EUR is not - Invoice status should be UPLOADED**
- Verify the Invoice Upload Status during upload process when **Invoice Maturity date is set for current date with delay 0 and no holidays in delay period in EUR from another timezone but EUR is already on next date - Invoice status should be EXPIRED**