Card Validator

Project Developed by **P. SARAVANAN**

Content:

- Requirement Documentation.
- Software Design Documentation.
- Technical Documentation (Backend coding).
- User Guide Documentation.

Project Overview:

Main Language: Java

Web Framework: Spring Boot

Tools: SpringToolSuite4

Project Purpose:

The project aims to validate Credit or Debit Card Numbers and PAN card Numbers.

Requirement Documentation

Credit Card Validator Requirements:

Luhn Algorithm:

A mathematical algorithm utilized internally within the application to validate credit and debit card numbers.

Credit/Debit Card Number:

This input field allows users to enter their credit or debit card number.

PAN Card Validator Requirements:

Indian PAN Card Algorithm:

An algorithm specific to Indian PAN cards, employed internally within the application for validation purposes.

PAN Card Number:

Users can input their 10-character PAN card number into this field.

PAN Holder's Last Name or Surname:

Users can input their last name or surname as it appears on the PAN card into this field.

Software Design Documentation

This project comprises totally 7 web pages.

Home Page(Page 1):

- The home page serves as the main entry point for the Card Validator application.

- It presents users with two validation options: Credit/Debit Card Validator and PAN Card Validator.

Credit / Debit Card Validator Page(Page 2):

HTML, CSS, and JavaScript are used on this page.

HTML Functionalities:

- Utilized <h1> for a bold title "Credit / Debit Card Validator".
- Added an input field labeled "Credit / Debit Card Number".
- Inserted a placeholder "Enter 16-digit card number" to guide and ensure 16-digit entry.
- Included a "Validate Card" button for validation.

CSS Functionalities:

- Ensured proper centering of the page content.
- Styled the title "Credit / Debit Card Validator" with a big, bold font.
- Set the color of the "Validate Card" button to blue by default and changed it to green when the user hovers over it.

JAVASCRIPT Functionalities :

- Restricted users to type only numbers for the 16-digit Credit/Debit card number.

- Automatically insert spaces after every 4 digits to enhance readability.

- Display an error message if the entered number is not exactly 16 digits, prompting users to input a exact 16 digits Credit/Debit card number.

- The error message is automatically removed after the user enters exactly 16 digits in the "Credit/Debit Card Number" input field.

Pan Card Validation(Page 3):

HTML, CSS, and JavaScript are used on this page.

HTML Functionalities:

- Created a title "Pan Card Validation" using <h1> with bold font.
- Implemented input fields labeled "PAN Number" and "Pan Holder Name".
- Provided placeholder messages 'Enter PAN Number' inside the "PAN Number" input field and 'Enter Pan holder's Last name / Surname' inside the "Pan Holder Name" input field to guide the user and facilitate validation.
- Developed a "Validate Card" button for validation purposes.

CSS Functionalities:

- Ensured proper centering of the page content.
- Styled the title "Pan Card Validation" with a big, bold font.
- Set the color of the "Validate Card" button to blue by default and changed it to green when the user hovers over it.

JAVASCRIPT Functionalities :

- Ensure the "Pan Number" input field has exactly 10 alphanumeric characters.If the user enters more or less than 10 characters, an error indication will be displayed, stating "Please enter a valid 10-character PAN card number.ensuring clear and prompt validation.

- The error message is automatically removed after the user enters exactly 10 characters in the input field "Pan Number".

- Restrict the "Pan Holder Name" input field to accept only alphabetic characters, preventing the input of non-alphabetic characters for accurate validation.

- Display an error message if the "Pan Holder Name" field is left blank, guiding users to enter their name for clear and prompt validation.

Valid Page(Page 4 and 5):

- This page serves as the Validation Results Page. If the validation is successful, it displays "Your Card Number is Valid" for the Credit/Debit Card Validator and "Your PAN Card Number is Valid" for the PAN Card Validation.

On the Valid Page, there are two buttons:

- "Validate Another Card": This button allows users to validate another card.
- "Back to Home Page": This button takes users back to the Home Page.

Invalid Page(Page 6 and 7):

- This page serves as the Validation Results Page. If the validation fails, it displays "Your Card Number is Invalid" for the Credit/Debit Card Validator and "Your PAN Card Number is Invalid" for the PAN Card Validation.

On the Invalid Page, there are two buttons:

- "Try Again": This button allows users to attempt validation again.
- "Back to Home Page": This button takes users back to the Home Page.

Technical Documentation

Created Totally 7 classes for code to "Card Validator".

1) Created an "input_collect" class to store user inputs.

2) Developed a "string_to_Array" class to convert string characters to arrays, then passed to "Credit_or_Debit_Validation" class.

3) Separated credit/debit card validation into "Credit_or_Debit_Validation" class.

4) Isolated PAN card validation in "PanValidation" class.

5) Spring Boot automatically generated "CardValidatorApplication" class.

6) Linked all classes in "Card_Controller"(Controller Class) for smooth operation.

Maven Dependencies:

- **spring-boot-starter-web:** Starter for building web, including RESTful, applications using Spring MVC.
- **tomcat-jasper:** Provides JSP and JSTL support for a Spring Boot application.

- **spring-boot-starter-test:** Starter for testing Spring Boot applications.

Here clear details about all the class code,

Package Name is "CardValidate".

CardValidate Package

This package contains classes and components related to validating PAN (Permanent Account Number) cards, credit or debit cards, and collecting input data.

Class input_collect:

Description:
This class represents a POJO (Plain Old Java Object) for collecting input data related to Credit or Debit Cards and PAN cards.
Attributes:

UserPanName: String - Represents the name associated with the PAN card.
PanNo: String - Represents the PAN number.
digit: String - Represents Credit or debit card number.

Methods:

Getter, Setter Methods:

- getUserPanName(): Getter for retrieving the user's PAN name.
  setUserPanName(String userPanName): Setter for setting the user's PAN name.

- getPanNo(): Getter for retrieving the PAN number.
  setPanNo(String panNo): Setter for setting the PAN number.

- getDigit(): Getter for retrieving the Credit or debit card number with spaces.
  setDigit(String digit): Setter for setting the Credit or debit card number spaces.

Normal Method:

- removeSpace(String string)
  Description: Method to remove spaces from Credit or Debit card number (digit).
  Parameters:
  string: String – Credit or Dedit card number with spaces.
  Return Type: String
  Return Value: Credit or Dedit card number without spaces.

Class string_to_Array:

Description: This class provides a method to convert a string of digits to an integer array.

Methods:

convertStringToIntArray(String number):

Description: Method to convert a string of digits to an integer array.
The String values from removeSpace() from class input_collect.
Parameters:
number: String - The string of digits.
Return Type: int[]
Return Value: An integer array containing the digits from the input string.

Class Credit_or_Debit_Validation:

Description: This class provides methods for validating credit or debit card numbers.

Algorithm: Luhn Algorithm.

Dependencies:

string_to_Array: Autowired dependency for converting strings to integer arrays.

Methods:

valida(input_collect inc):

Description: Method to validate credit or debit card numbers.

Parameters:
inc: input_collect – The Int array input for this method.
Return Type: int
Return Value: The sum of the digits of the credit or debit card number.

Class Pan_Validation:

Description: This class provides methods for validating PAN card numbers.

Algorithm: Indian PAN card Algorithm.

Methods:

pan_valida(input_collect inc):
Description: Method to validate PAN card numbers.
Parameters:
inc: input_collect – String pan name and String pan card number input for this method.
Return Type: ModelAndView
Return Value: A ModelAndView object representing the validation result.

Class CardValidatorApplication

Description: This class serves as the entry point for the application.

Methods:

main(String[] args):
Description: The main method to start the Spring Boot application.

Class Card_Controller;

Description: This class serves as the controller for handling HTTP requests related to card validation.
Dependencies:

Pan_Validation: Autowired dependency for PAN card validation.
input_collect: Autowired dependency for collecting input data.
Credit_or_Debit_Validation: Autowired dependency for credit or debit card validation.

Methods:

- mainHome_page():
  Description: Method to handle the GET request for the main home page.
  Return Type: String
  Return Value: The name of the HTML file for the main home page.

- credit_debit_Homepage():
  Description: Method to handle the GET request for the credit card search page.
  Return Type: String
  Return Value: The name of the HTML file for the credit card search page.

- pan_card__Homepage():
  Description: Method to handle the GET request for the PAN card search page.
  Return Type: String
  Return Value: The name of the HTML file for the PAN card search page.

- CreditDebit_validation(input_collect inc):

  Description: Method to handle the validation of credit or debit cards based on the input data.
  Parameters:

inc: input_collect - The input data collected.
Return Type: ModelAndView
Return Value: A ModelAndView object based on the validation result.

- pan_validation(input_collect inc):

Description: Method to handle the validation of PAN cards based on the input data.
Parameters:
inc: input_collect - The input data collected.
Return Type: ModelAndView
Return Value: A ModelAndView object based on the PAN card validation result.

User Documentation

How To Use This Web Application:

- First Run the “Card Validator” Application.
- Then you navigate to localhost:8080/cardvalidator in your browser, you will be directed to the Home Page of the Card Validator web application.
- From here, you have the option to choose between "Credit or Debit Card Validator" or "PAN Card Validator".

Credit or Debit Card Validator:

- The "Credit or Debit Card Validator" app prompts you to input a 16-digit card number. Upon entering the exact number, validation begins. If valid, it shows "Your Card is Valid" Page ; otherwise, "Your Card is Invalid" Page.
- If your card validation results are "Valid", the Result page will feature two buttons at the bottom center: "Validate Another Card" and "Back to Home Page", enabling you to either validate another card or return to the Home Page of the Card Validator web application.

- If your validation results indicate that the card is invalid, the options displayed at the bottom center "Try Again" and "Back to Home Page".

PAN Card Validator:

- Upon selecting "PAN Card Validator" on the Home Page, you will be directed to the next web page where you will be asked to input your PAN Number and PAN Holder Name. After entering your inputs, you will receive the validation results.

- If the validation results for your PAN card indicate "Valid", the Result page will present two buttons at the bottom center: "Validate Another Card" and "Back to Home Page", allowing you to validate another PAN card or return to the Home Page of the Card Validator web application.

- If the validation results for your PAN card indicate "Invalid", the options displayed at the bottom center will include "Try Again" and "Back to Home Page".

Note:

This site provides results based on the Credit Card Luhn Algorithm and the Indian PAN Card Algorithm.