# Credit Card System

# Project architecture

**Credit Card Microservice**

(<http://localhost:8080/CreditCardSystem/api/v1/home>)

Springboot, Lombok, RestController, Entity, ExceptionHandler, Validators, Junit Test

**Angular 13 UI**

(<http://localhost:4200/>)

Modules, Components, Routes, Services, Material

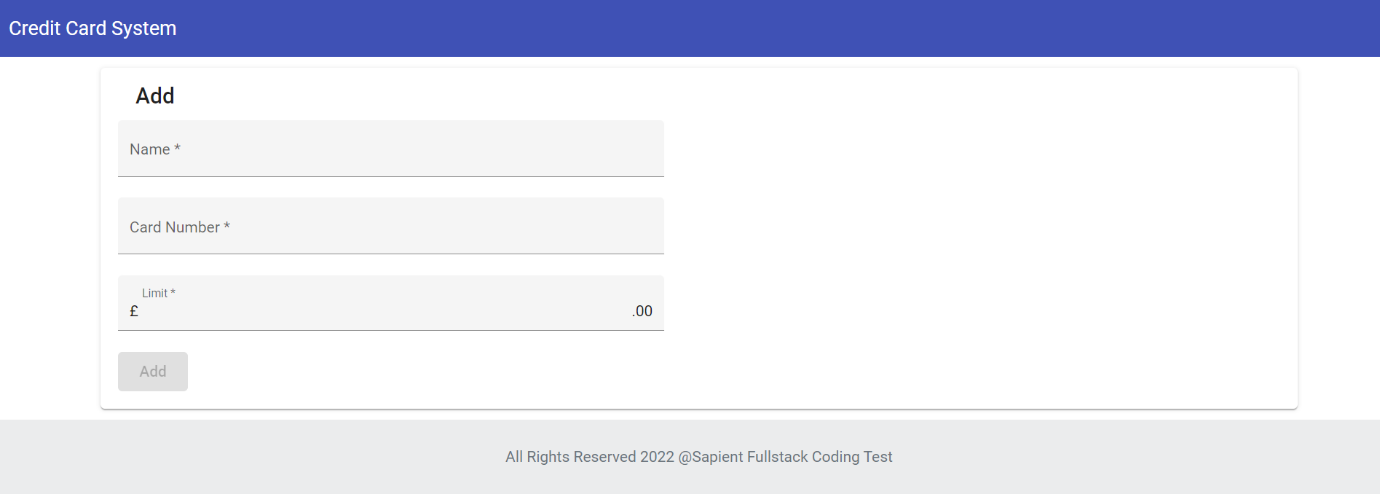
Spring data JPA

H2

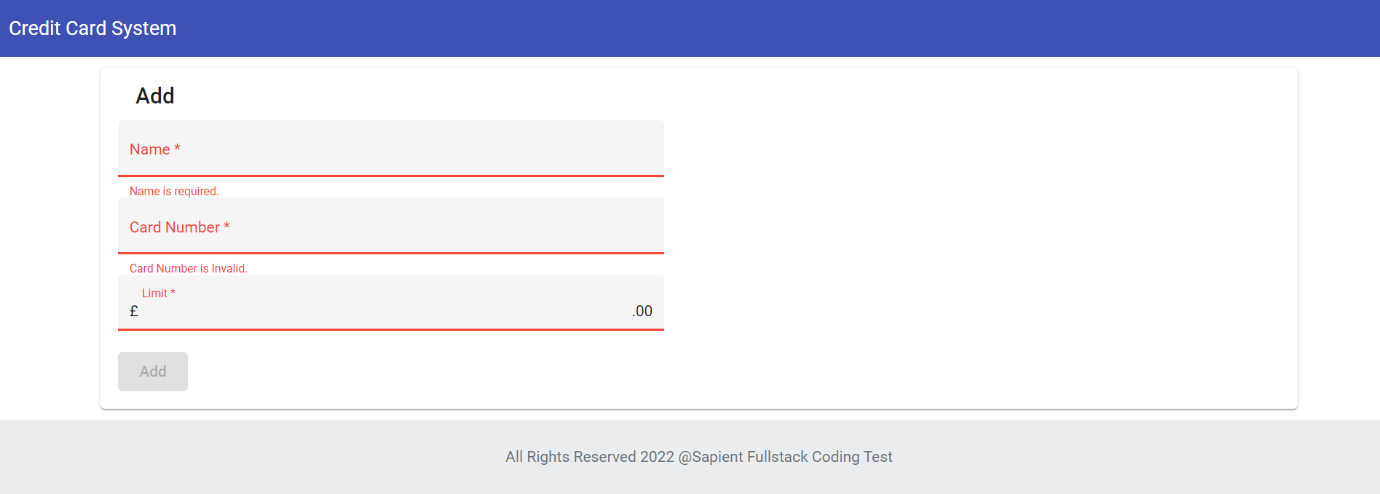
(In Memory DB)

JSON

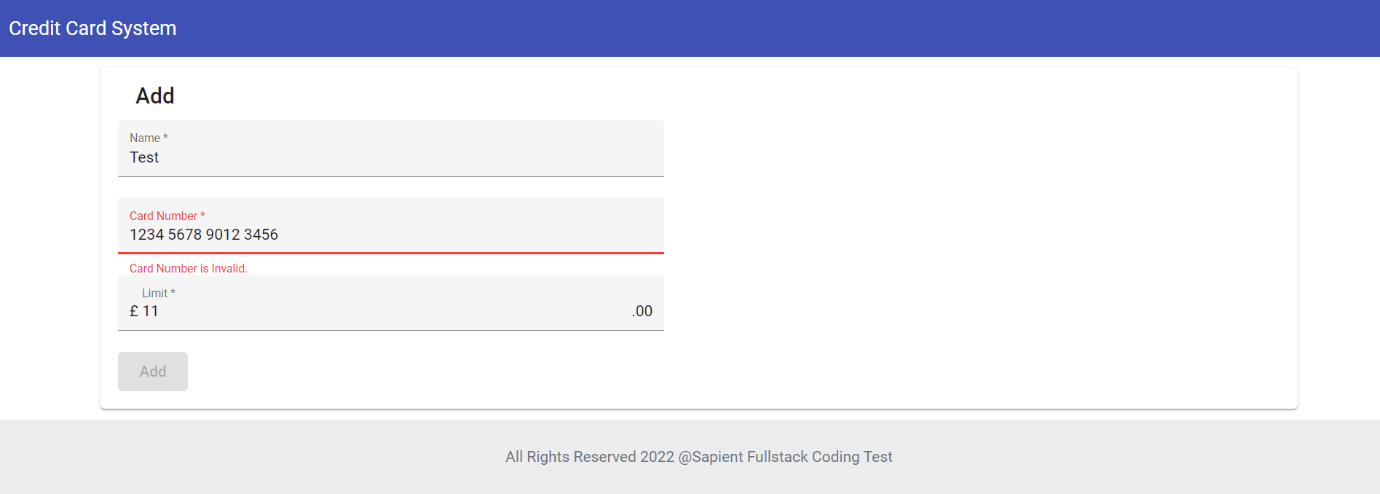
# Screen flow



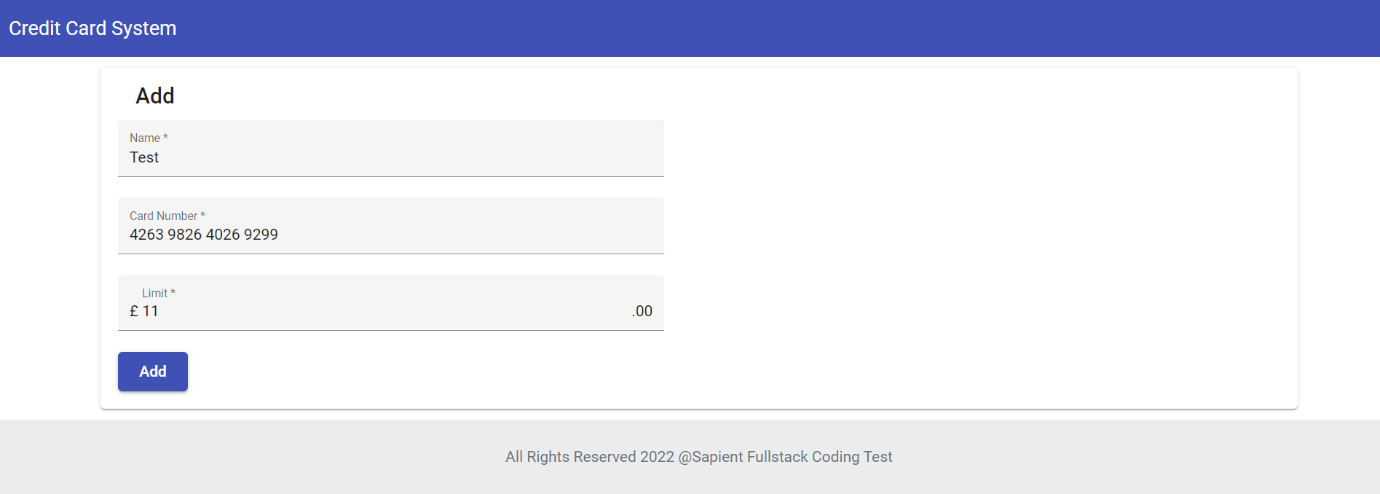
*Screen to add new card details in system*



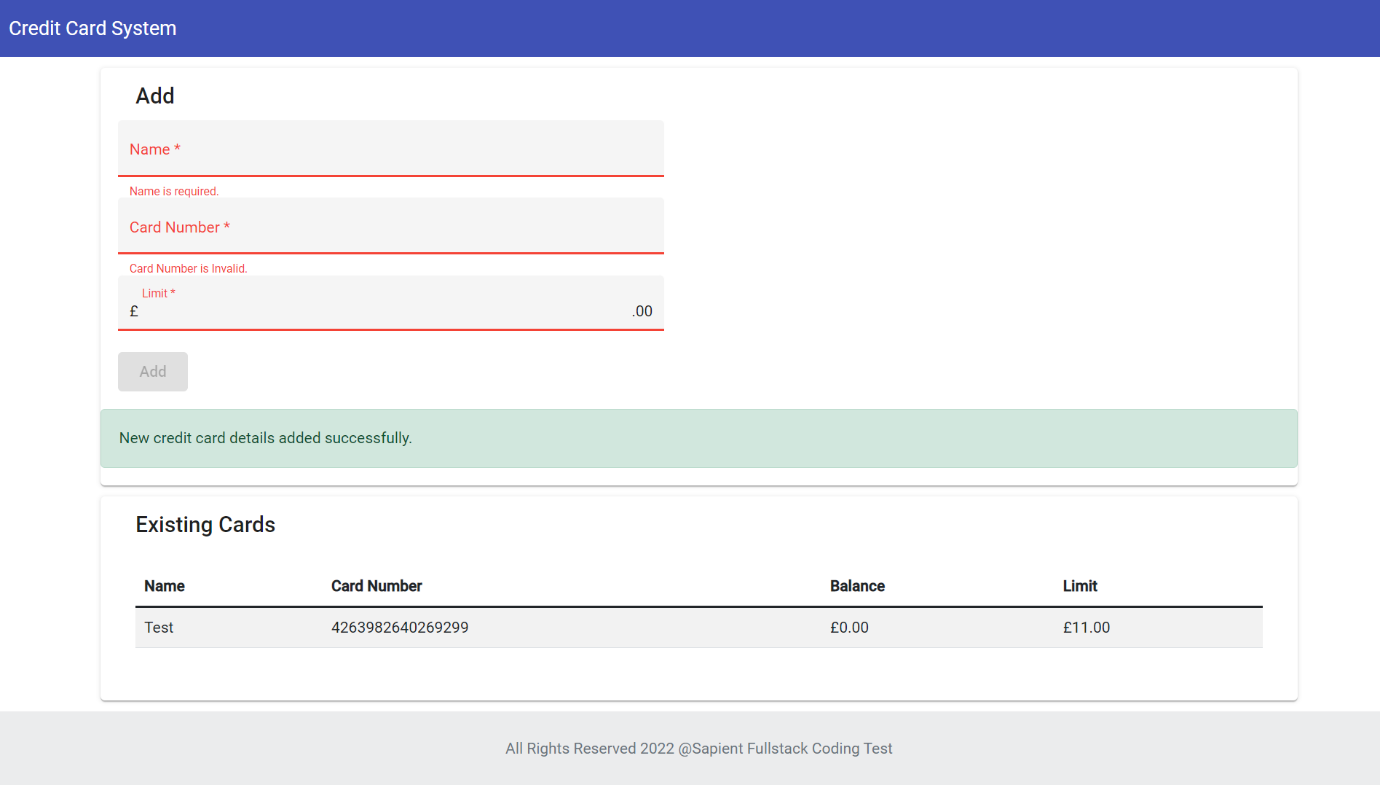
*Front end form field validations*



*Luhn Algo to validate card details*



*Add button enabled post validating input fields*



*New Card details added successfully with existing card details table*

# Code Structure

GitHub URL - <https://github.com/lalith-pandey/credit-card-processing-repo>

1. CreditCardSystem - Rest endpoints written in Java 8 using Springboot 2.6.7 framework.
2. CreditCardSystemUI - Angular 13 UI components for view and add credit cards. Also, used angular material and bootstrap.

**A. Starting Tomcat for BE API**

Download Java code and run using command line or any IDE - mvn clean install . The tomcat container would start on default port 8080.

API end points –

List of existing cards - GET - http://localhost:8080/CreditCardSystem/api/v1/home

Add new Card details - POST - <http://localhost:8080/CreditCardSystem/api/v1/add>

Request body – {"balance":0,"cardLimit":11,"cardNumber":"4263982640269299","name":"Test"}

**B. Starting angular server**

Download UI code and install packages using npm install.

Run angular server locally using ng serve on port 4200.

http://localhost:4200/

**C. H2 in memory DB**

login to H2 console using details provided in application.properties - http://localhost:8080/CreditCardSystem/h2