**Student ID:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Full Names:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Software Development Practicum

(CS318)

(December 2019)

Instructor: O. Kalu

Project 2

--------------------------------------------------------------------------------------------------------------------

Make sure to include the screenshots of your results, as required.

--------------------------------------------------------------------------------------------------------------------

(CS318 - SDP)

(December 2019)

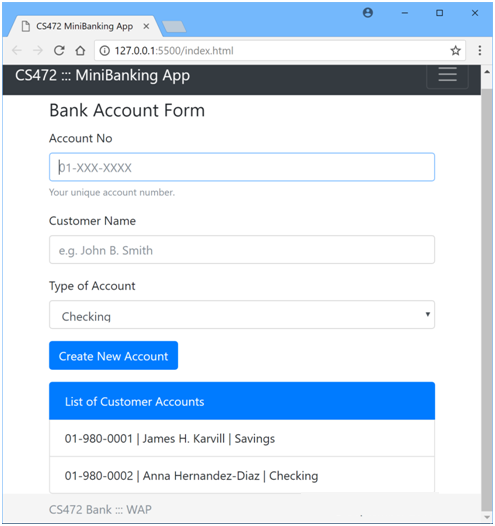
Project 2 – Mini-Banking Web App (40 points)

**JavaScript/Web Application Coding:** (40 points)

**Note:** *For the tasks in this project, where applicable, you are expected to take screenshot(s) of your UI(s), save into a .png or .jpg image file, placed inside a folder named, screenshots and include these in you repo commits.*

**Implementing a simple client-side JavaScript Single-Page web application**

Using HTML, CSS and JavaScript, implement a mini-BankingApp as a single-page web application, as shown in the UI screenshot below, with the following features and functionalities:



**Requirements / Specifications:**

* 1. Code the User interface of the webapp using standards-compliant, semantically-correct HTML5 markup, including all the form fields as shown in the UI screenshot above.
  2. Apply styling using Bootstrap or you may apply your own custom CSS styling to produce the same or similar look and layout. **Note:** Your UI does NOT necessarily have to be exactly the same as the sample shown above. But it should have all the necessary form data input fields, labels and buttons etc.
  3. Account Number, Customer Name and Type of Account are all required, to create a new Bank account.
  4. Add validation using appropriate regular expression to ensure that any Account Number entered must be in the specified format, 01-XXX-XXXX, as shown on the UI above.
  5. The user can create any one of three available types of Bank Accounts – Checking, Savings or Loan.
  6. Using Fetch API (or some other JS Library/API you know/prefer (e.g. XMLHttpRequest or JQuery or Axios etc) and AJAX, load and display a List of existing Customer Accounts data, by making an AJAX Http-GET request to a JSON-formatted data file located at the url – <http://yourApplicationHostAddress/data/customerData.json>. You are required to create this JSON data file using the following raw data:

Account 1:

Account Number: 01-980-0001,

Customer Name: James H. Karvell,

Type of Account: Savings

Account 2:

Account Number: 01-980-0002,

Customer Name: Anna Hernandez-Diaz,

Type of Account: Checking

* 1. Implement code for the ‘Create New Account’ functionality which reads the data entered into the Bank Account form and adds the new account data to the List on the webpage.
  2. Implement code to add all the Accounts data to the Web Browser's LocalStorage, such that when the browser is closed and re-opened afterwards, the mini-Banking webapp still displays all customer Account data, including any newly added accounts.

(**Please note:** Points will be awarded based on your completion of the above given specs).

**//-- The End --//**