|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| Due Date: | | November 25, 9AM (Two weeks from the time of issue) | |
| Percentage of ASP.NET 1 module mark: | | 20% | |
| Late Penalty: | | 20% deducted each day this assignment is late. | |
|  |  | |

**Application Setup**

**MVC**

* All code must be implemented using MVC.

**LINQ**

* All queries and database operations must be performed using LINQ and the entity framework.

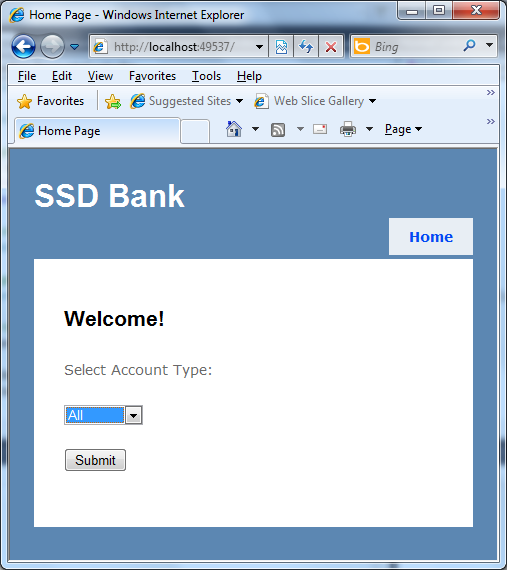
**SQL**

* Use the following SQL to generate your database named **Bank. (1 mark)**
* Do not use stored procedures or modify this SQL.

|  |
| --- |
| DROP TABLE ClientAccount;  DROP TABLE BankAccount;  DROP TABLE Client;  GO  CREATE TABLE Client(  clientID INT,  lastName VARCHAR(20),  firstName VARCHAR(20),  PRIMARY KEY(clientID)  );  INSERT INTO Client VALUES (1, 'Appleby', 'Alana');  INSERT INTO Client VALUES (2, 'Barr', 'Bob');  GO  CREATE TABLE BankAccount  (  accountNum INT,  accountType VARCHAR(15),  balance DECIMAL(18,2),  CHECK( accountType IN ('Savings', 'Chequing')),  PRIMARY KEY(accountNum)  );  INSERT INTO BankAccount VALUES (10, 'Savings', 10.10);  INSERT INTO BankAccount VALUES (20, 'Chequing', 20.20);  INSERT INTO BankAccount VALUES (30, 'Chequing', 30.30);  GO  CREATE TABLE ClientAccount  (  clientID INT FOREIGN KEY REFERENCES Client(clientID),  accountNum INT FOREIGN KEY REFERENCES BankAccount(accountNum),  PRIMARY KEY(clientID, accountNum)  );  INSERT INTO ClientAccount VALUES(1, 10);  INSERT INTO ClientAccount VALUES(2, 20);  INSERT INTO ClientAccount VALUES(1, 30); |

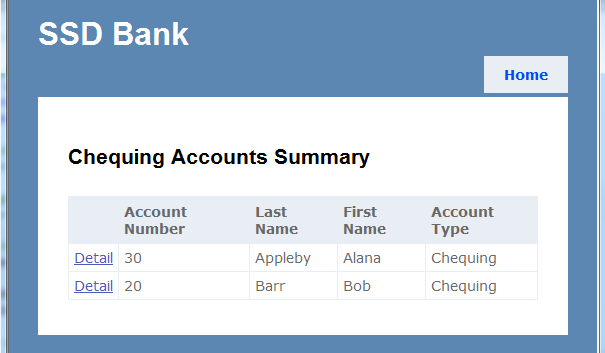
**Home/Index**

Allows you to choose from a drop-down list which provides the options ‘All’, ‘Chequing’, or ‘Savings’. You can then submit your choice. You can hard code in ‘All’, ‘Chequing’, or ‘Savings’ selection options. (2 marks)



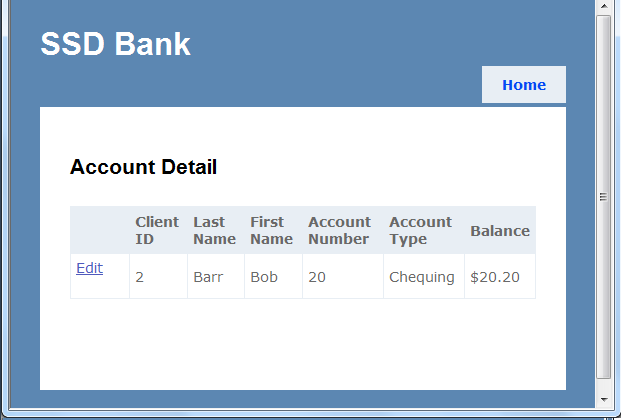
**Accounts/Index**

* Displays all accounts for the type that was selected in the home page. (1 mark)
* Show the same columns as displayed in the screenshot. (3 marks)
* The title indicates the type of summary according to the user’s selection in Home/Index. In this case the title says ‘Chequing Accounts Summary’. If the user had selected ‘All’ the title would be ‘All Accounts Summary’. (1 mark)
* The detail link allows users to navigate to Accounts/Detail to view details about the account. (1 mark)



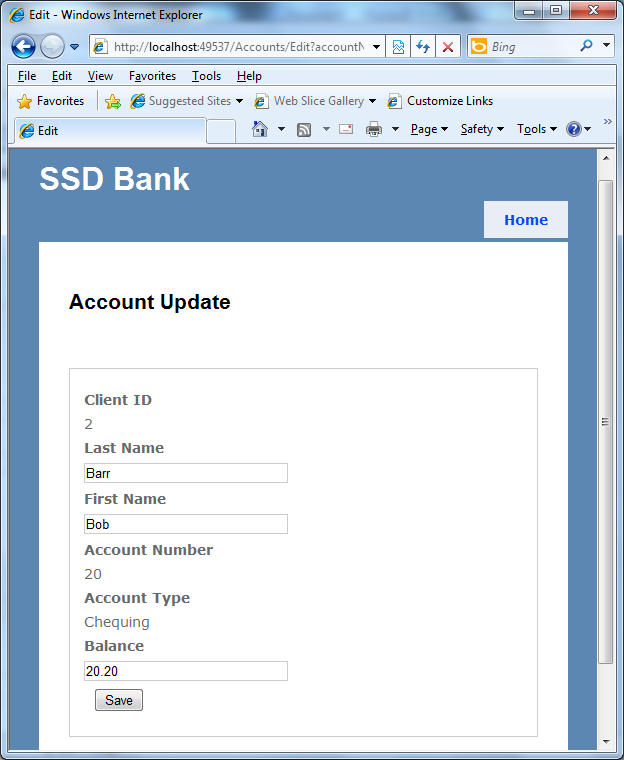
**Accounts/Detail**

* Shows all of the column information presented below. (2 marks)
* Has a link to Accounts/Edit where details for the account shown on the page can be modified. (2 marks)



**Accounts/Edit**

* Displays the fields as shown. (1 mark)
* Allows you to edit Last Name, First Name, and Balance only. (2 marks)
* Redirects the user back to the Accounts/Detail where the updated information is saved. (3 marks)
* Applies validation to: (4 marks)
  + ensure all enabled fields are not empty.
  + Last names and first names may only include alphabetical characters, spaces, and hyphens.
  + Last names and first names must start with an alphabetical character.
  + Balance is a valid number which must include two digits to the right of the decimal.



**View Models and Repository Models**

* Both the listing and detail views require you to query both tables to generate data for the view. Use a view model to do this. Also note, since the view model is not a real entity in the database you will have to parse out the submitted data during the update to edit each entity separately. (2 marks)
* Place most of your logic in repository models and not in the controller. No logic should exist in the vie w model. (2 marks)

**Layout Page**

* Include a Home Link that appears on all pages which directs you back the Home/Index.
* Include a nice looking Image logo in your layout.
* Use the same layout page for all of your views so you only have to add the code for the image and home link once. (3 marks)