Samurai/Battle project

Please restore the provided samurai.bak and add to your localdb

Please run the program in Debugging mode, not start without debugging

This wpf project has two parts to the main window, Samurai and Battles that Samurai are involved in.

Please Click the Fetch Data buttons to load the data, it is a bit slow to due having to match samurai and battle.

There is a list of all Samurai in the database. When selecting an item in the list, name, town, age and picture are displayed. Also if the program is running in database mode, the list of all battles the samurai is fighting/fought in is displayed (I will refer to this as the secondary Battle list). Selecting an item in the secondary battle list will populate the battle details on the right side.

There is a list of Battles in the database. When selecting an item in the list, name, date, city and country are displayed. Also if the program is running in database mode, the list of all samurai fighting/fought in is displayed (I will refer to this as the secondary Samurai list). Selecting an item in the secondary samurai list will populate the samurai details on the left side.

Add a samurai to a battle button uses the selected samurai and the selected battle. Please re-fetch to see changes.

The program has two modes: database and csv. If a connection to a database is available, the database will be seeded if blank. If the database is unavailable the program will run in csv mode with functionality limited to reading data from the csv file

In the database mode, each samurai and battle can have their fields updated. And a samurai can be added to a battle.

The program saves all data to csv files every 5 minutes

Ben: build/design the ui. Link all fields/commands to the view model with data binding, modify the view model as required. Write the repository for csv mode

Tao: Work with Ben to Create ViewModel, provide UI data binding sources and Call data access repository, monitor and respond to  all Command buttons events, implement data fetching with Async/await.

Pei-Hua:

Create the data model and initial database creation using entity framework code first approach

Worked with Dong to do the initial seeding of the database

(run into an issue where .net core project can't be referenced by .net framework project the rest of the team is using.

the team ended up doing database first to create the service using the database I had created)

Write the code to write all samurais and battles from database to CSV every 5 min

Nico and Aaron: Created the Samurai and and Battle Services that queries the original database from Pei-Hua. The services returns the data from database, except for the Samurais in Battle and Battles of Samurai. Workaround was to make an operation contract that returns all the Battles of a Samurai and Samurais in a Battle. Created the repository that stores all the data from the Samurai and Battle services. The problem is a slow loading because the data contract does not include Samurais in Battle and vice versa and had to do an operation for each Battle and each Samurai.