1- Changes to application logic

       1-1- Based on single responsibility principle, separated data access logic from controller     classes and put them in classes specifically responsible for handling data access corresponding to each entity type in the application model.

       1-2- Implemented input validation for values posted from the client side to make sure they cannot violate business requirements. Added a custom “PositiveNumberAttribute” for this purpose.

2- Changes to the view model

   2-1- Renamed model classes with more self-explanatory names.

   2-2- Organised view model classes in categorically cohesive bundles.

   2-3- Changed “the amount” property from int to decimal to allow accurate calculations and to avoid unnecessary conversions for factional math purposes.

3- Changes to Razor view templates

    3-1 Renamed them to clearly and simply reflect their purpose

    3-2 Further developed their mark-up to wrap input elements in forms with proper labels, validation message placeholders, etc.

4- type-safety and name-safety

    4-1 Used type-safe references in view files instead of hard-coded references to view model names.

    4-2 Used type-safe references in RouteConfig instead of hard-coded references to controller names.

    4-3- Couldn’t afford time to implement dependency injection for maximum type-safety and avoidance of tight-coupling between components of the solution.

5- Testing

5-1- Added a test project with unit tests.

                5-2 Added unit tests to target controller actions and logic calculation separately

6- Added style sheets with minor aesthetic changes as a foundation for further UI development.

============================================