Readme

Добрый день тут я объясняю, что это такое и с чем его едят :-D
Как вы заметили это не простая программа, а решение в чем-то схожее на CMS? так вот вам не показалось :)
Говорю сразу, писал все я но для другого боевого проекта, для вашего проекта было изменено и дописано процентов 40%, в качестве основного подхода используется "инверсия зависимостей (IoC)" + библиотека для IoC от MS - Unity

А теперь пройдемся более подробно по системе
Первое на что вам нужно обратить внимание это папка Install\bitcoin-0.15.1-win64\bin как вы догадываетесь это локальная нода биткоин с помощью команды nsssm.exe install "BitcoinNodeServer" и содержимого файла start-bitcoin-node-server.bat организуем windows службу для локальной ноды биткоин чтобы она нас не беспокоила своим присутствием :), естественно запускаем созданную службу :) если вы все сделали правильно то по сути мы не будем часто лазить сюда повторно, только при рестарте службы для повторной генерации биткоинов на опыты :)

Теперь рассмотрим более подробно проект :), для справки все делалось в окружении "vasiliy.stan" это наследник от окружения "Debug" но персонализированное :)
1) библиотека Sources\BitcoinNet.Client - базовая библиотека которая организует взаимосвязь биткоин ноды и основного приложения по протоколу json-rpc, была переписана с исходников 2011 года которые были найдены на bitcoinWiki, в старых исходниках 99% занимали исходники библиотеки Json.NET :-D, а о преобразовании json в обьекты своих классов старый автор имел ну ооооочень смутное представление :-D
Идем дальше по степени важности
2) проекты Sources\Dal\Dal и Sources\Dal\Los - библиотеки на основе EF6 и подхода CodeFirst для доступа к данным и логирования, через библиотеку log4net, соответственно :)
3) проект DataGenerator - проект для развертывания баз данных и инициализации их первичными значениями, сделанно это потому, что постоянно переключаться на сервак и делать bak копии 2-3 баз меня задолбало + ведь не потащишь develop базы с ошметками тестов на production :) заказчик не поймет :) а тут 1 раз настроил корректно строки соеденения и параметры для sql скриптов и забыл :), также она служит не просто развертованию баз, а выполняет роль генератора первичных данных для bitcoind сервиса. Запускаеться она с администраторскими правами, это было нужно для корректного управления bitcoind сервисом, в ней выполняеться генерация 1000 поколений вычислений хеш сумм, и рассылка первичных сумм биткойнов на разные кошельки, в виде 8 генераций из которых подтверждается 7 из них

4) Sources\WinServices\Core.Bitcon.Service - windows служба на основе пакета Topshelf, основное ядро по взаимодействию WebApi и биткоин нодой через BitcoinNet.Client, она предоставляет для каждой WebApi функции легковесный TcpListener который выдерживает 30000 асинхронных коннектов. Также были проработаны механизмы кеширования, сбора данных от bitcoind и централизованного выполнения bitcoin send запросов, централизованное ядро выполняет данные функции с переодичностью раз в минуту, в промежутках между выполнением данных функций идет набор данных для следущего раунда и выдача закешированных результатов. число 30000 выбрано не спроста при 1000 коннектов в секунду, на работу одного коннекта можно выделить всего 1мс что по сути очень мало для работы, если же принять что для работы 1 коннекта требуется в среднем 30мс то получаем соответствующий размер асинхронного пула коннектов.
5) модуль Sources\Frontend.Web.Core\Modules\Modules.WebApi, основной модуль WebApi системы т. к. вам нужна производительность (от 1000 запросов в секунду) асинхронен и посути представляет трубу на сокетах в из сторону Sources\WinServices\Core.Bitcon.Service он тподдерживает кеширование результатов полученных из Core.Bitcon.Service, а также каждая функция WebApi имеет возможность выдержать 1000 коннектов в секунду к ядру системы
6) Sources\Frontend.Web.Core\Libraries\Libraries.Core.Backend - основная бекенд библиотека для всей системы хранит в себе классы с базовой функциональностью
7) Sources\Frontend.Web.Core\Libraries\Libraries.Core.Frontend - основная фронтэнд библиотека по логике хранит в себе базовые модули на CSS и Jquery которые используються на фронтэнде
8) Sources\Frontend.Web.Core\Themes\Themes.Core - библиотека базовой темы сайта на основе Bootstrap
9) Sources\Frontend.Web.Core\Modules\Modules.Core - базовый модуль который хранит повторяемые страницы web сайта на подобии страницы входа, чтобы не верстать их заново в каждом новом приложении
10) Sources\Frontend.Web.Core\Modules\Modules.Site - основной модуль который хранит основные SPA для веб сайта
11) Sources\Frontend.Web.Core\Modules\Modules.WebApi.Shared - библиотека которая расшаривает классы запросов и ответов на основе которых и построен протокол WebApi текущего приложения
12) Sources\Frontend.Web.Core\Modules\WebApi.Client - посути содержит в себе класс - оболочку которая из C# может работать с WebAPI информационной системы
13) Sources\Frontend.Web - основное приложение построенное на ASP .NET MVC4 которое динамически связыывает все остальные части
14) Sources\Translations - библиотека мультиязычных переводов для сайта основанная на стандартном подходе с использованием *.resx файлов от MS

по возможности все "магические" элементы, как то строки или числа, были вынесены в файлы настроек

P. S. если у вас будут вопросы по проекту или просто пообщаться, то со мной можно связаться по email: vasiliyStankevich@gmail.com