Explanatory note

«Love Calculator»

Продукт предназначен для вычисления совместимости между людьми по их датам рождения(Compatibility), а также для подбора идеального партнёра(Perfect match). Также пользователь имеет возможность поделиться полученным результатом в социальной сети [ВКонтакте](https://vk.com/). Еще одной особенностью наше проекта является то, что в вычислениях мы используем базу данных пар, собранную вконтакте с помощью его API.

Адрес репозитория:

<https://github.com/ZnamenskayaVictoria/Love-Calculator>

Члены команды и их роли:

Znamenskaya Victoria 152(2) - Server logic, client logic

Kononova Daria 152(2) - UI, Data collection, testing

* Изначально мы под Server Logic подразумевали просто хранение базы данных на сервере, а всю её обработку делать на клиентской стороне. Но в итоге большая часть обработки данных происходит тоже на сервере, а в клиенте только обращение к серверу и небольшие расчёты.

Классы:

**DataCollection** - здесь с помощью пакета NuGet VkNet происходит обращение к API вконтакте и сбор данных. Сначала проходит авторизация в аккаунт социальной сети, потом проходит поиск людей, у которых стоит семейное положение, затем среди друзей этих людей ищутся люди, у которых в профиле стоит партнёр, все эти пары сохраняются в отдельные файлы, а потом собираются в один .csv файл.

**ServerInteraction/Querier.cs** - класс, который осуществляет взаимодействие с нашим сервером. Обращения к серверу происходят через URL <http://lovecalculator.somee.com/api/PerfectDate> , в конце которого указываются месяц и день, например /11/1, итоговая ссылка ведёт на страничку с результатом запроса, который представляет из себя небольшой Json объект, который потом парсится регулярным выражением и из него вычленяется дата - день рождения идеального партнера для введённого.

**LoveMagic/Matcher.cs** - класс, который по полученным данным вычисляет конечный результат для пользователя. В случае с Perfect match он просто возвращает результат запроса к серверу.

Для совместимости(Compatibility):

Допустим введено две даты, A и B, сначала от сервера мы получаем Perfect match для A, и считаем, на сколько дней он отклоняется от B. Затем получаем Perfect match для B, и вычисляем отклонение от A в днях. Затем полученные дни суммируем. Если суммарное отклонение меньше 60 дней(мы сделали 60, а не 30, чтобы процент совместимости был выше реального и не был слишком низок, как это могло

получаться во многих случаях), то их совместимость 100%, затем совместимость равномерно снижается за каждый день, так как максимальное суммарное отклонение составляет 365 дней (если для каждого партнёра его идеал будет на другой стороне года).

**VKInteraction/Publisher.cs** - класс предназначен для публикации результата на стене вконтакте, также к записи прикрепляется картинка.

**Pages** - в этой папке лежат разметки страниц приложения, а так же файлы с обработчиками событий объектов, которые используют перечисленные выше классы.

Описание сервера

Сервер написан на базе технологии Web API 2 в ASP.NET от Microsoft. Для реализации базы данных использовался Entity framework.

Основные части сервера:

**Маршруты** сервера содержатся в папке **App\_start** в файле WebApiConfig

Маршруты это определяющая часть, они указывают серверу какой контроллер и с какими параметрами вызывать.

Например, конкретно для этого проекта существует маршрут "api/{controller}/{month}/{day}"

То есть мы указываем адрес сервера (в нашем случае http://lovecalculator.somee.com/), потом слово api, имя обработчика запроса и сами параметры (месяц и день)

**Контроллер** задает реакцию сервера на запрос. В нашем случае это анализ базы данных и выдача какого-то результата. Например, наш контроллер называется PerfectDate, значит обращение к нему будет осуществляться по адресу api/PerfectDate. После названия мы указываем параметры, и в контроллере PerfectDate выбирается метод с подходящей сигнатурой. В нашем случае там один метод GetDate(int month, int day).

* Еще доступен контроллер ZodiacSignsController, который выдает статистику по знакам зодиака. Он доступен по маршруту api/ZodiacSigns/{ZodiacId}.

Можно сказать, что он позволяет производить альтернативный анализ. Но это не реализовано в нашем проекте, скорее, это один из возможных вариантов реализации теста на совместимость, но наш тест реализован на основе даты рождения.

* Так же есть стандартный контроллер который позволяет получить инфу о записи в базе по id. Доступен по маршруту api/Partners/{id}.

**Модели** описывают логику данных, то есть, типы, которые возвращает наш API.

В проекте есть 3 модели:

**Partners** описывает формат базы данных, то есть нашу выборку.

**PerfectDate** при создании считает наиболее подходящую дату для данной.

Поля:

DateTime Date - искомая дата.

double MaxCoef - наибольший коэффициент совпадения с теми данными, что есть в базе.

DateTime Exists - наиболее подходящая дата из базы.

**ZodiacSigns** при создании собирает статистику для данного знака.

Интерфейс программы





















