1. СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ
   1. Структура программной модели тотализатора игровых видов спорта с возможностью предсказания результата игры

Приложение построено с использованием клиент-серверной архитектуры и представляет собой одностраничное приложение (single-page application – SPA), реализующее паттерн MVC.

Структурная схема программной модели тотализатора игровых видов спорта с возможностью предсказания результата игры представлена на чертеже ГУИР.400201.075.С1 и состоит из следующих компонентов:

* блок управления;
* блок работы с базами данных;
* блок клиентского интерфейса;
* блок квалиметрического анализа;
* блок нечетких предсказаний;
* блок анализа на основе нейроной сети;
* блок предсказания;
* блок корректировки коэффициентов;
* база данных пользователей;
* база данных игр.

Далее каждый компонент рассматривается более подробно.

* + 1. Блок управления

Блок управления является одним из основных в приложении, так как все решения о действиях программы принимаются в нем. Из блока клиентского интерфейса в блок управления поступают данные о действиях пользователя, в соответствии с которыми либо отправляется запрос в блок работы с базами данных на корректировку или получение имеющихся данных, либо запускается процесс предсказания результата события в блоке предсказания. После получения предсказания блок управления передает данные блоку корректировки коэффициентов для вычисления видимых всем пользователям коэффициентов.

* + 1. Блок работы с базами данных

Блок работы с базами данных предназначен для управления всеми имеющимися в приложении базами данных, выборки и привединя к необходимому формату данных от них. Все процессы в нем начинаются по команде от блока управления, результаты этих процессов так же передаются в него.

* + 1. Блок клиентского интерфейса

Блок клиентского интерфейса представляет собой непосредственно сам интерфейс приложения. В программе он взаимодействует только с блоком управления, от которого получает все необходимые данные для отображения пользователю , и куда отправляет всю информацию о действиях пользователя. Интерфейс представляет собой одностраничное приложение, поэтому никаких задержек такой подход не создает.

* + 1. Блок квалиметрического анализа

Блок квалиметрического анализа взаимодействует только с блоком предсказания. От него он получает данные, необходимые для анализа, и ему же отдает результат своей работы. Предсказание результата происходит квалиметрическим методом, данные для которого поставляются из базы данных игр через несколько других блоков. Далее полученные данные проходят через процедуру оценки, результатом которой является так называемая «сила» обеих команд. Сравнивая эти «силы», блок возвращает блоку предсказания три коэффициенты на игру.

* + 1. Блок нечетких предсказаний

Блок нечетких предсказаний взаимодействует только с блоком предсказания. От него он получает данные, необходимые для анализа, и ему же отдает результат своей работы. На основе этих данных строится таблица нечетких аргументов, которая далее анализируется с помощью лексических выражений «если -- то». Результатом данного анализа являются выбор одного события. Для поддержания консистентности возвращаемых блоками анализа данных. Оставшиеся коэффициенты приравниваются к нулю.

* + 1. Блок анализа на основе нейронной сети

Блок анализа на основе нейронной сети взаимодействует только с блоком предсказания. От него он получает данные, необходимые для анализа, и ему же отдает результат своей работы. До выставления программы пользователю в данном блоке производится обучение нейронной сети на основе статистических данных, в процессе которого формируется структура сети и задаются веса составляющих ее нейронов. Задача данного блока обобщенно сводится к задаче кластеризации. Варьируя число кластеров можно улучшать результат предсказания. Собственно результатом работы блока будет являться отнесение результата события к некоторым из кластеров с определенной долей вероятности.

* + 1. Блок предсказания

Блок предсказания отвечает за обработку результатов от блоков квалиметрического анализа, нечетких предсказаний и анализа на основе нейронной сети. Данный блок получает данные от них в виде трех коэффициентов от каждого, и далее он производит их компоновку. На выходе получается итоговый результат предсказания, который отдается в блок управления для дальнейших действий с ним. Основой для предсказания являются данные от нейронной сети, достоверность которых в большинстве случаев выше достоверности результатов оставшихся блоков. Данные этих блоков используются для корректировки результата блока нейронной сети в ту или иную сторону. Каждый из трех упомянутых выше блоков получает данные для анализа из блока предсказания. Блок предсказания отправляет запрос на эти данные в блок управления, а после передает их блокам анализа.

* + 1. Блок корректировки коэффициентов

Блок корректировки коэффициентов получает от блока управления данные о предсказании результата события, вычисленные в блоке предсказания. Далее в эти коэффициенты закладывается прибыль владельца тотализатора в размере 15-20%. Так же данный блок предоставляет возможность корректировать коэффициенты по мере поступления ставок от пользователей. Чем больше человек поставило на такой исход события, тем меньше становится коэффициент для последующих игроков. Скорректированные коэффициенты возвращаются блоку управления.

* + 1. База данных пользователей

База данных пользователей хранит в себе всю информацию о зарегистрированных приложении пользователях и их правах. Для каждого пользователя сохраняется информаци о сделанных им ставках и полученных выплатах.

* + 1. База данных игр

База данных игр хранит в себе всю накопленную приложением статистическую информацию о прошедших играх, структурированную по видам спорта, а внутри них по турнирам. Здесь же хранится информация о предсказанных результатах событий с информацией о ее соответствии действительности.

* 1. Обоснование выбора программных средств

Клиент-серверные [9] приложения обычно принято называть трехуровневыми, так как они расположены на трех различных системах: клиентский компьютер, сервер ASP.Net и сервер базы данных или обычный сервер. Трехуровневые приложения, работающие данным способом, расширяют стандартную архитектуру клиент-сервер, добавляя многопоточный сервер приложений между клиентской частью и сервером базы данных. Эта модель решает, в основном, задачу развёртывания приложения. С использованием клиент-серверной модели созданы многие приложения для работы с базами данных, электронной почтой и для доступа к веб-ресурсам. Основные принципы данного архитектурного стиля:

* клиент инициирует один или несколько запросов, ожидает ответа на них, а затем обрабатывает ответы;
* в определённый момент времени клиент подключён к одному серверу для обработки запросов (реже – к небольшой группе серверов);
* клиент работает с пользователем напрямую, применяя графический интерфейс;
* сервер не инициирует запросов;
* обычно для выполнения запросов клиенты проходят аутентификацию на сервере.

Главными преимуществами клиент-серверной модели являются:

* все данные хранятся на сервере, обеспечивающем больший уровень безопасности, нежели отдельный клиент (высокая безопасность);
* так как данные хранятся только на сервере, ими легко управлять (централизованный доступ к данным);
* роль сервера могут выполнять несколько физических компьютеров, объединённых в сеть (устойчивость и лёгкость сопровождения).

Благодаря этому клиент не замечает сбоев или замены отдельного серверного компьютера. В системах клиент-очередь-клиент сервер исполняет роль очереди для данных клиентов. Пиринговые приложения– это вариация системы клиент-очередь-клиент, в которой любой клиент может играть роль сервера. Сервера приложений служат для размещения и выполнения программ, которыми управляет клиент.

Данный дипломный проект разрабатывался в рамках ASP.NET Framework. Это набор спецификаций и соответствующей документации для языка C# и платформы .Net, описывающей архитектуру серверной платформы для задач средних и крупных предприятий.

Спецификации детализированы настолько, чтобы обеспечить переносимость программ c одной реализации платформы на другую. Основная цель спецификаций — обеспечить масштабируемость приложений и целостность данных во время работы системы. .Net во многом ориентирована на использование её через веб как в интернете, так и в локальных сетях.

.Net является промышленной технологией и в основном используется в высокопроизводительных проектах, в которых необходима надежность, масштабируемость, гибкость.

ASP.Net — технология, позволяющая веб-разработчикам динамически генерировать HTML, XML и другие веб-страницы. Является составной частью единой технологии создания бизнес-приложений .Net. Технология позволяет внедрять .Net-код в статичное содержимое страницы.

Встроенная защита от различных видов нападений, предоставляет следующие возможности: SQL Injection, переполнение буфера, XSS, изменение скрытых полей и прочие. Технология ASP.NET повышает степень устойчивости к вредоносным действиям и различным видам хакерских атак сайтов, построенных на ней.

В качестве фреймворка для клиентской части был выбран AngularJS. AngularJS представляет собой opensource JavaScript-фреймворк, использующий шаблон MVC. Собственно, использование MVC является его одной из отличительных особенностей. Для описания интерфейса используется декларативное программирование, а бизнес-логика отделена от кода интерфейса, что позволяет улучшить тестируемость и расширяемость приложений. Другой отличительной чертой фреймворка является двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом. Таким образом, AngularJS синхронизирует модель и представление. Кроме того, AngularJS поддерживает такие функциональности, как Ajax, управление структорой DOM, анимация, шаблоны, маршрутизация и так далее. Мощь фреймворка, наличие богатого функционала во многом повлияла на то, что он находит свое применение во все большем количестве веб-приложений, являясь на данный момент наверное одним из самых популярных javascript-фреймворков.

В качестве базы данных была выбрана MySQL. Это одна из самых популярных СУБД в современных интернет-технологиях. К основным плюсам MySQL можно отнести высокую скорость работы, быстроту обработки данных и оптимальную надежность. Немаловажно и то, что данная СУБД распространяется бесплатно и представляет собой программное обеспечение с открытым кодом. За счет этого Вы можете вносить свои изменения и модифицировать код, что весьма полезно для веб-мастеров.