

## АДМИНИСТРАТОРСКАЯ СОСТАВЛЯЮЩАЯ ТЕОРЕТИЧЕСКОГО ТЕСТИРОВАНИЯ

Свиржевский Артем Юрьевич

*УО «Брестский государственный университет им. А.С. Пушкина»*

Многие современные сетевые протоколы построены на архитектуре клиент-сервер. В их основе обычно лежат одинаковые или схожие принципы взаимодействия, а разница обнаруживается в деталях, обусловленных особенностями и спецификой области, для которой разрабатывался тот или иной сетевой протокол.

Разрабатываемая нами система представляет собой полноценный веб-сервер, ориентированный на организацию теоретического тестирования. Система разрабатывается средствами фреймворка asp.net core. Для хранения данных используется SQL Server. Приложение обменивается с ним информацией при помощи Entity Framework Core. При построении приложения применялись принципы многоуровневой архитектуры. Приложение разделено на несколько слоев:

1. слой хранилища. На этом уровне производятся действия, связанные с обращением к базе данных;
2. слой логики. На этом уровне производится обработка информации и основные вычисления. Этот уровень является связующим звеном между слоем хранилища и внешним уровнем – слоем представления;
3. слой представления. Основная задача данного уровня – обеспечить взаимодействие пользователя и системы.

Для прохождения тестирования клиенту необходимо иметь в своем распоряжении любой браузер и находиться в сети, из которой будет доступен WEB-сервер с установленным приложением.

Разрабатываемая система контроля знаний имеет три основных варианта использования: *тестирование студентов, просмотр результатов преподавателем и администрирование системы*. Названные варианты, в свою очередь, порождают соответствующие модули системы. Мы наделяем разрабатываемую систему возможностью протоколирования каждого сеанса тестирования с сохранением подробной информации о клиенте, его ответах и итогах тестирования.

При указанном подходе обработка всей информации осуществляется на стороне сервера. В частности, обработка запроса клиента на сеанс тестирования. Результат запроса – формирование html-размеченного текста. Заполняя формы в html-документе, клиент выполняет задание. Ответив на нужное количество вопросов или по достижению иного события, согласно сценария тестирования, клиент получает результат,

который сохраняется в системе, как и ответы на все вопросы теста в протоколе.

К преимуществам предлагаемой модели взаимодействия клиент-сервер можно отнести то, что программные коды клиентского и серверного приложений разделены. Это обстоятельство позволяет решать проблемы временного «падения» сервера за счет его оперативного перезапуска.