3. Тихомиров А.В. Концепция развития систем энергообеспечения и повышения энергоэффективности использования ТЭР в сельском хозяйстве / А.В. Тихомиров // Вестник ВИЭСХ, 2016.-№1 (22). –С.11-17.
4. Расстригин В.Н. Развитие электрических систем и технических средств теплообеспечения животноводческих предприятий / В.Н. Расстригин, Д.А. Тихомиров // Техника в сельском хозяйстве.- 2010.-№2.- С.3-6.
5. Тихомиров Д.А. Энергоэффективные электрические средства и системы теплообеспечения технологических процессов в животноводстве Д.А. Тихомиров // Вестник Всероссийского научно-исследовательского института механизации животноводства. 2016. –№4 (24). – С.15-23.
6. Кузьмичев А.В. Теплоэнергетические показатели ИК облучателей для молодняка животных / А.В. Кузьмичев, А.К. Лямцов, Д.А. Тихомиров // Светотехника.- 2015.– №3.– С. 57–58.
7. Васильев А.Н. Разработать методы и модели построения систем электроснабжения сельских потребителей. Выполнить научное обоснование параметров и режимов работы новых системообразующих технических средств теплоэнергообеспечения сельскохозяйственных объектов /А.Н. Васильев, Е.В. Халин, А.И. Некрасов и др. / Отчет о НИР/НИОКР.- Федеральный научный агроинженерный центр ВИМ. –2017. – 104 с.
8. Клычев Е.М. Влаготепловая обработка комбикормов перед экспандированием / Е.М. Клычев, Д.А. Тихомиров, С.Г. Карташов, А.А. Мансуров // Механизация и электрификация сельского хозяйства. 2003.–№1.–С.19.
9. Машков А.Н. Разработка электротехнологии термической обработки сои / А.Н. Машков, Д.А. Тихомиров, А.М. Шувалов // Наука в центральной России. 2018. – № 1 (31). – С. 67-75.
10. Kuzmichyov A.V., Malyshev V.V., Tikhomirov D.A. Efficiency of the combined pasteurization of milk using UV and IR irradiation / Light & Engineering. 2011. Т. 19. № 1. С. 74-78.
11. Тихомиров Д.А. Энергосберегающая система горячего паро- и водообеспечения животноводческих объектов / Д.А. Тихомиров // Техника и оборудование для села. – 2013. № –10.– С.14-15.
12. Тихомиров Д.А. Энергосберегающие электрические средства и системы теплообеспечения основных технологических процессов в животноводстве / Д.А. Тихомиров // Вестник НГИЭИ.– № 8 (63), 2016.- С.129-138.
13. Дудин С.Н. Устройства аккумуляторного типа для нагрева воздуха / С.Н. Дудин Д.А. Тихомиров // Вестник ВИЭСХ.–2012.–№3.– С. 73-74.
14. Тихомиров Д.А. Программный проект для расчета потребной мощности теплоэнергетического оборудования и годового расхода тепловой энергии на объектах животноводства / Д.А. Тихомиров // Вестник ВИЭСХ. –2013.– Вып. 1(10).– С. 47–50.

Цыденов А.Б.

Архитектура модуля «Индивидуальный учебный план студента»

*Восточно-сибирский государственный университет технологий и управления
(Россия, Улан-Удэ)*

doi:10.18411/lj-31-03-2018-26

idsp: 000001:lj-31-03-2018-26

Введение

Модуль «Индивидуальный учебный план студента», рассчитан на студента, который сможет самостоятельно формировать свою траекторию обучения. При этом предполагается, что количество пользователей-студентов неограниченно, поэтому модуль разрабатывается как веб-приложение.

В автоматизированном информационном комплексе (АИК) ВСГУТУ все веб-приложения разрабатывались с помощью SpringFramework [1]. Так как модуль «Индивидуальный учебный план студента» входит в АИК ВСГУТУ, то он тоже разрабатывается на этом же фреймворке.

Архитектура модуля

В связи с тем, что мы имеем ограниченное число ресурсов, к которым требуется ограниченный правами доступ большого числа потребителей, то удобно реализовать клиент-серверную архитектуру [2], что и сделано при разработке АИК ВСГУТУ. Рассматриваемое приложение тоже разработано на основе клиент-серверного шаблона (Client-Server pattern, рис. 1).

Рисунок 1 – Трехуровневая архитектура приложения

Клиентская часть – средство, которое принимает данные и показывает их в виде понятном человеку. Этим средством является браузер, мобильное приложение, работающее с данными с сервера, компьютер, телефон, планшет, телевизор или радиоприёмник. Перечисленные выше средства помогают отрисовать в виде понятном для человека данные, которые получены с сервера.

Так как, в данной работе рассматривается веб-приложение, то данные с сервера будут приходить в виде:

- Html разметки;
- Css стилей;
- Javascript (дополнительный незначительный функционал который можно реализовать на стороне клиента).

Сервер приложения - на нём сосредоточена большая часть бизнес-логики. Основываясь на архитектуре типичного java-spring приложения, структура будет делиться на следующие компоненты:

1. *Controlllers* – компоненты, которые получает запросы и передает их всем остальным компонентам приложения. В самом конце компонует ответ на запрос, отправляет на определенное представление (*views*).
2. *DAO* – это слой объектов которые обеспечивают доступ к данным, которые связывают базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».
3. *Services* - это компоненты, которые предоставляют основную (Бизнес-Логика). В основном сервис использует готовые DAO или же другие сервисы (обработка, преобразование или расчет данных), для того чтобы предоставить конечные данные для пользовательского интерфейса.
4. *Views* – это набор шаблонов возвращаемых клиенту. Здесь формируется Html страницы, заполненные данными полученными с контроллера.

Сервер БД - обеспечивает хранение данных и выносятся на отдельный уровень. Реализуется, как правило, средствами систем управления базами данных, подключение к данному компоненту обеспечивается только с уровня сервера приложений.

Структурная схема модуля «Индивидуальный учебный план студента» АИК ВСУТУ

На рисунке 2 показана структурная схема вышеназванного модуля.

Рисунок 2 – Структурная схема модуля

Краткое описание структурных элементов:

1. Контроллер(controller) получает https-запрос от клиента. В зависимости от типа запроса срабатывает нужный контроллер.
2. Контроллер передает параметры в нужный сервис (Service). Каждый сервис отвечает за определенную обработку или бизнес-функцию приложения.
3. Если сервису нужны данные из БД, то он вызывает DAO сервис, который подгружает данные из БД и хранит в памяти.
4. После обработки данных, в контроллере формируется динамическая модель, где помещают обработанные данные, и наименование представления (view).
5. Шаблонизатор по наименованию представления выбирает определенный готовый шаблон, и наполняется данными из модели.
6. Готовый шаблон, наполненный данными, отправляется Клиенту в формате html-страницы.

Заключение. В статье рассмотрена трехуровневая архитектура приложения, а также его функциональная схема. Предложенный подход позволит неограниченному количеству студенту пользоваться модулем и строить индивидуальную траекторию обучения.

1. Крейг Уоллс, Spring в действии. 3-е изд., ДМК Пресс, 2015.— 754 с.
2. Леон Шкляр, Рич Розен, Архитектура веб-приложений. Перев. Михаил Р., Отдельное изд., Эксмо, 2010. — 640 с.