

УДК: 004.04

**Перцев Т. Е.**  
магистрант 1 курса  
Хакасский Государственный Университет им. Н.Ф. Катанова

## РАЗРАБОТКА ИНФОРМАЦИОННОГО ПРОФИЛЯ СТУДЕНТА ДЛЯ АИС «СТУДЕНТ» ХАКАССКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА ИМ. Н.Ф. КАТАНОВА

*В работе описывается существующая проблема поддержки автоматизированной информационной системы, основанной на устаревшей технологии Silverlight и перехода на технологию MVC6. Приводится обзор основных алгоритмов обработки данных и пользовательского интерфейса.*

**Ключевые слова:** студент, профиль, АИС, контроллер, процедура, данные.

В процессе ведения учебной деятельности, образовательное учреждение использует множество информации об обучающихся студентах, такие как документы по движению, различные приказы, информацию о стипендиях и прочие.

На текущий момент в Хакасском государственном университете им. Н. Ф. Катанова используется автоматизированная информационная система (АИС) «Студент» [1], разработанная с применением программной платформы для многофункциональных интернет-приложений Microsoft Silverlight 5 [2]. АИС «Студент» представляет собой базу данных в MSSQL Server 2012 [3] и веб-интерфейс. Применение Silverlight привело к усложнению программного кода и трудностям в разработке дополнительных функций, внедряемых в систему. Так же поддержка продукта компанией-производителем будет прекращена в 2021 году. С учетом вышеперечисленных факторов было принято решение об усовершенствовании АИС «Студент» и полном рефакторинге и разработке веб-оболочки с применением фреймворка ASP.NET MVC 6 [4]. Применение MVC 6 позволяет упростить программный код, облегчить модификацию продукта и внедрение новых функций, а так же увеличить производительность веб-интерфейса.

Одним из самых востребованных разделов приложения является профиль студента, в котором отображаются и редактируются все данные о студенте: информация об учебной программе, оплате за обучение, паспортные данные, данные о предыдущем образовании, проживании в общежитиях, контактная информация, льгота, воинская обязанность, членство в профсоюзе, результаты промежуточной аттестации, а так же отображается список приказов и документов, в которых фигурирует конкретный студент.

Переход к профилю студента реализован с помощью вкладки «Студенты», где отображаются списки академических групп. Внешний вид списка группы представлен на рисунке 1.

Перейдя в раздел «Студенты», пользователь попадает в список групп, где должен выбрать учебно-структурное подразделение (УСП); институт или колледж, а так же форму обучения. На этом этапе контроллер оболочки вызывает хранимую процедуру базы данных Students\_Get, передавая ей ID института и форму обучения. Процедура возвращает XML-файл в котором записаны студенты с краткой информацией, разделенные на группы. После чего вызывается конструктор модели StudentsModel, которая заполняет поля модели из XML. После этого будет отображен список групп данного института, обучающихся по выбранной форме (очная, заочная или очно-заочная). При переходе в группу будет показан список студентов данной группы. Для дальнейшей работы оболочка записывает небольшой фрагмент данных (в cookies браузера) с информацией о выбранном подразделении, чтобы отобразить то же УСП при повторном открытии.

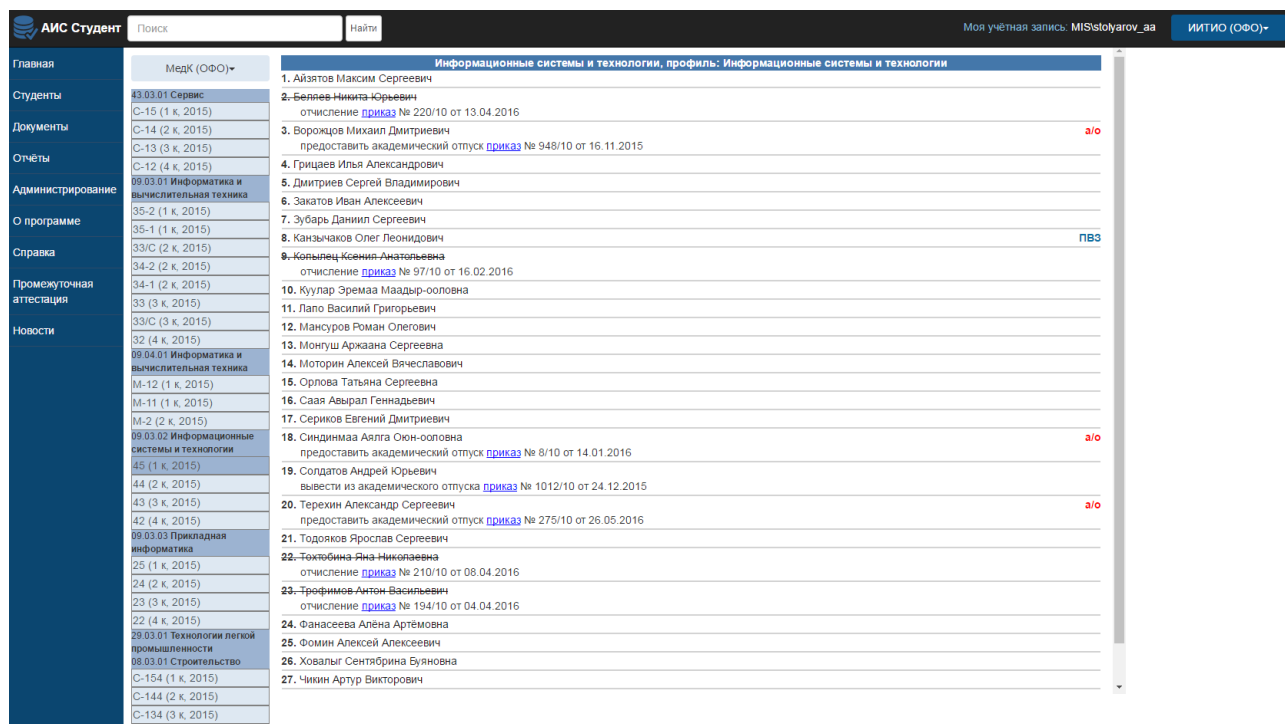


Рисунок 1. – список групп института

В списке студентов группы отображаются все текущие студенты, а так же при наличии таких данные, как отпуск студента, информация о платной форме обучения и приказ об отчислении. При переходе в профиль студента, контроллер вызывает хранимую процедуру StudentProfile\_Get, передавая ей только ID студента. Процедура возвращает XML-файл, содержащий все имеющиеся данные о студенте, перечисленные выше. Каждый раздел профиля заворачивается в отдельный узел в XML, в котором уже располагаются вложенные элементы. Для работы с данными, возвращенный процедурой XML также разделяется конструктором модели StudentProfileModel, заполняя свойства модели. Все незаполненные поля имеют значение NULL. После чего модель возвращается в представление StudentProfile.

Веб-страница представления написана с использованием языков HTML5 и JQuery. Скрипты JQuery используются для валидации данных при вводе в изменяемые поля. Внешний вид пользовательского интерфейса показан на рисунке 2.

Защита данных является важной задачей при предоставлении данных пользователям АИС «Студент». Проверка прав доступа осуществляется на уровне хранимых процедур и в контроллере оболочки. Права пользователя хранятся в таблицах базы данных. Контроллер, вызывая хранимую процедуру Security\_CheckAccessLeftPanel проверяет доступ пользователя к пункту меню «Студенты». При отсутствии доступа, пользователь получит соответствующее информационное сообщение. Данные о группах и студентах не будут переданы.

Если же пользователь имеет доступ к разделу, то при открытии профиля студента, хранимая процедура StudentProfile\_Get проверяет доступ пользователя к каждому из разделов профиля и записывает их в XML. Права доступа разделяются на: «Нет доступа», «Чтение» и «Запись». Если доступа к какому-либо разделу у пользователя нет, то соответствующие данные не будут вложены в узел передаваемого XML-файла и вкладка на странице профиля не будет отрисована. Если права только на чтение, то у пользователя будет отсутствовать возможность редактирования полей. Если же присутствуют права на запись, то скрипты страницы при отрисовке снимут блокировку редактируемых полей и кнопок «Изменить», «Сохранить» и «Отменить».

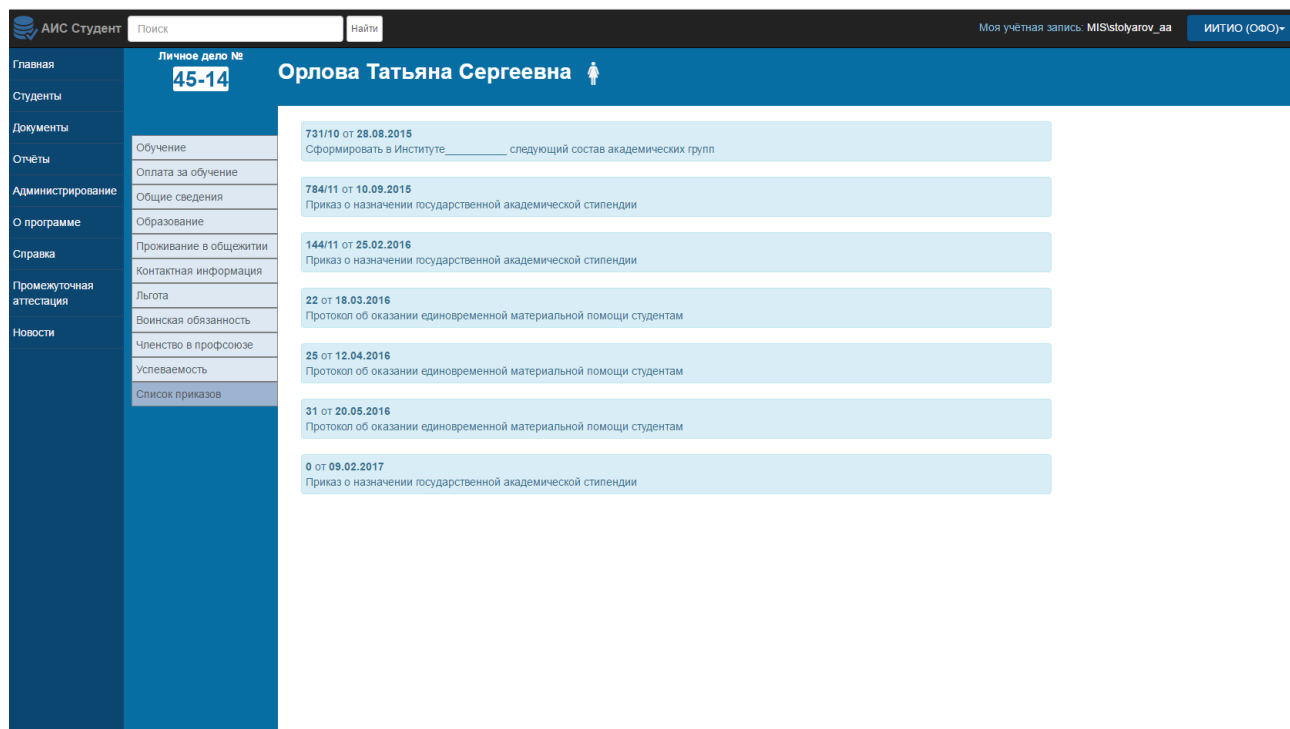


Рисунок 2. – профиль студента

Для защиты от случайных нажатий отрисовка страницы происходит со всеми заблокированными полями ввода. При необходимости внести изменения и наличии соответствующих прав записи на конкретный раздел, пользователь может нажать кнопку «Изменить», после чего поля ввода будут разблокированы. В процессе ввода данных, скрипты на странице проверяют корректность ввода и форматы данных, адресов, телефонов, электронной почты и других.

По завершении ввода, пользователь должен нажать кнопку «Сохранить». Для сокращения сетевого трафика и увеличения скорости работы было принято решение отправлять только часть модели, соответствующую редактируемому разделу профиля. В контроллере ActionResult «Save» получает экземпляр класса StudentProfileModel с частично заполненными данными для сохранения и в зависимости от параграфа, который необходимо сохранить, перенаправляет обработку в один из методов сохранения параграфов. В каждом из методов сохранения параграфов, из полученных данных собирается XML-файл, в который вкладываются только те поля, которые были изменены. После чего собранный XML-файл передается в хранимую процедуру StudentProfile\_Save. Процедура вновь проверяет права доступа пользователя к сохраняемому разделу и производит запись данных в соответствующие таблицы. Если сохранение прошло успешно, пользователь получит соответствующее сообщение на странице ввода и сможет продолжить просматривать или редактировать другие разделы профиля.

При реализации профиля студента на технологии MVC, удалось существенно упростить компоновку модулей, увеличить скорость работы, снизить влияние человеческого фактора: избавиться от случайных нажатий и исключить ввод некорректных данных, а так же скрыть от пользователей информацию, которая им не должна быть доступна. Использование более современной и постоянно развивающейся технологии позволяет устанавливать более короткие сроки внедрения новых модулей.

### Литература

1. Сопроводительная документация на автоматизированную информационную систему «Студент» Хакасского государственного университета им. Н.Ф. Катанова.

2. Мэтью Мак-Дональд. Silverlight 5 с примерами на С# для профессионалов. – М. : ООО И.Д. Вильямс, 2013. — 849 с. — 4-е изд. — ISBN: 5845917840, 9785845917843.

3. Ицик Бен-Ган. Microsoft SQL Server 2012. Основы T-SQL. – М. : Эксмо, 2015. – 401 стр. – ISBN: 978-5-699-73617-1.

4. Adam Freeman. Pro ASP.NET MVC 6 / Springer, 2016. – 1018 стр. – ISBN: 9781484203989.