

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ**

«Московский государственный индустриальный университет»

(ФГБОУ ВПО «МГИУ»)

**КАФЕДРА ИНФОРМАЦИОННЫХ
СИСТЕМ И ТЕХНОЛОГИЙ**

КУРСОВАЯ РАБОТА

по дисциплине

«Проектирование и разработка корпоративных информационных систем»

на тему

**«Разработка интерактивной обучающей Web среды для студентов
программистских направлений»**

Группа

111131

Студент

А.Ю. Дьячук

Руководитель работы

Н.А. Роганова

Москва 2015

Аннотация

Работа посвящена разработке интерактивной обучающей Web среде для студентов программистских направлений. Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться с помощью визуального графического интерфейса. Интерфейс системы должен быть понятен и удобен для использования. Он должен обеспечивать удобный доступ к основным функциям и операциям системы. Элементы навигации должны быть реализованы в удобной для пользователя форме. Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

Оглавление

Введение.....	4
Глава 1. Структура системы.....	8
1.1 Описание миграций.....	8
1.2 Описание ассоциаций.....	10
1.3 Создание вопроса для тестирования.....	11
1.4 Создание ответа для тестирования.....	12
Глава 2. Описание интерфейсов.....	14
1.1 Просмотр всех вопросов.....	14
1.2 Создание новых вопросов.....	15
1.3 Отображение созданного вопроса в презентации.....	16
1.4 Редактирование вопроса.....	17
Заключение.....	18
Список литературы	19

Введение

Курсовая работа бакалавра представляет собой разработку интерактивной обучающей Web среде для студентов программистских направлений, в которой представлены модели: «Пользователь», «Вопрос», «Презентация», «Группа», «Документ», «Разделы». Цель данной информационной системы заключается, в дистанционном обучении студентов программистских направлений обучающихся в МГИУ. Проверку полученных знаний за определенный период времени и формирование отчета по тестированию по каждому конкретному студенту для преподавателя.

Элементы навигации должны быть реализованы в удобной для пользователя форме. Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние, предшествовавшее недопустимой команде или некорректному вводу данных.

Для решения нам потребуется создать модели, в которых будет использоваться миграции, валидации и контроллеры в самом проекте. Для решения курсовой работы использовались объектно-реляционные преобразователи ActiveRecord, Javascript и основы CSS, Ruby on Rails [1,2], СУБД Post SQL, HAML, HTML[3], и базовые знания написания SQL запросов.

Используемые технологии.

Для создания проекта мы используем Ruby on Rails [6] - объектно-ориентированный программный каркас для создания веб-приложений, который основан на архитектуре Модель-Представление-Контроллер (Model-View-Controller, MVC).

Для решения поставленной задачи необходимо было создать пять новых моделей. Все модели должны храниться в директории app/models в отдельных файлах. В Rails модель реализована в виде ActiveRecord

Структура базы данных представлена пять моделей, которые построены следующим образом:

1. Презентация («Presentation») - одна из главных моделей в базе данных, с которой связано большинство моделей. В ней указывается комментарий, группа, последний видимый слайд, авто открытие, и операции;
2. Ответ («Answer») - это модель, которая создает и редактирует ответ на поставленный вопрос. В данной модели нужно указывать правильный ответ, чтоб перейти на следующий слайд;
3. Вопрос («Question») - это модель, которая создает вопрос, который включается в презентацию для прохождения тестирования;
4. Пользователь («User») – это модель, которая создает пользователя для прохождения тестирования;
5. Группа («Group») – это модель, в которой указывается группа в которой находится студент;

Представление [4] (View), вид отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами. Наиболее типичная реализация отделяет вид от модели путем установления между ними протокола взаимодействия, используя аппарат событий (подписка/оповещение). При каждом изменении внутренних данных в модели она оповещает все зависящие от неё представления, и представление обновляется. Для этого используется шаблон «наблюдатель». При обработке реакции пользователя вид выбирает, в зависимости от нужной реакции, нужный контроллер, который обеспечит ту или иную связь с моделью. Для этого используется шаблон «стратегия», или вместо этого может быть модификация с использованием шаблона «команда». А для возможности однотипного обращения с подобъектами сложно-составного иерархического вида может использоваться шаблон «компоновщик». Кроме того, могут использоваться и другие шаблоны проектирования, например, «фабричный метод», который позволит задать по умолчанию тип контроллера для соответствующего вида.

Контроллер (Controller) обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции. Не затрагивая реализацию видов, можно изменить реакции на действия пользователя (нажатие мышью на кнопке, ввод данных), для этого достаточно использовать другой контроллер. Важно отметить, что как представление, так и контроллер зависят от модели. Однако модель не зависит ни от представления, ни от контроллера. Тем самым достигается назначение такого разделения: оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

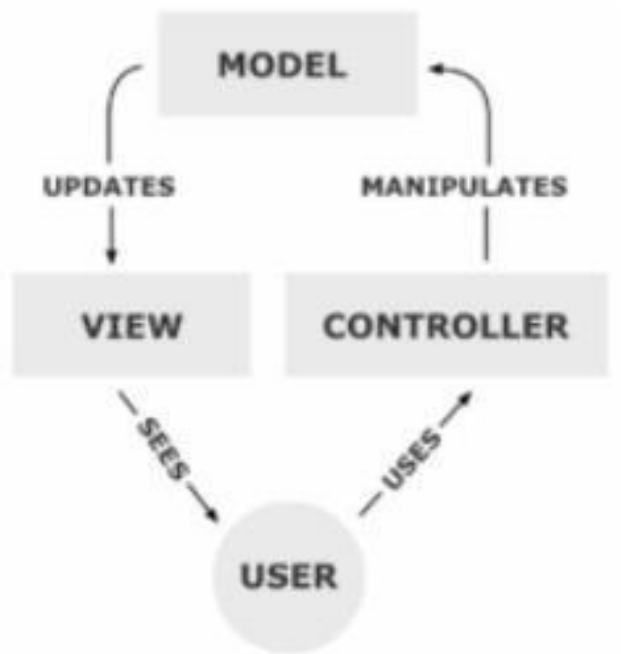


Рис. 1. Диаграмма контроллера

Глава 1. Структура системы

Описание миграций

Миграции – это обновление структуры базы данных от одной версии до другой. Каждую миграцию можно рассматривать как новую 'версию' базы данных. Схема изначально ничего не содержит, а каждая миграция изменяет ее, добавляя или убирая таблицы, столбцы или записи. Active Record знает, как обновлять вашу схему со временем, перенося ее из определенной точки в прошлом в последнюю версию. Active Record также обновляет ваш файл `db/schema.rb`, чтобы он соответствовал текущей структуре вашей базы данных.

Некоторые миграции имеют поля, которые отвечают за ограничение целостности: NOT NULL, UNIQUE, CHECK, PRIMARY KEY, FOREIGN KEY.

Файл миграции для модели «Вопросы»:

```
class CreateQuestions < ActiveRecord::Migration
  def change
    create_table :questions do |t|
      t.text :name
      t.integer :answer_type
      t.text :answer_array
      t.integer :position
      t.references :presentation, index: true
      t.timestamps
    end
  end
end
```


Эта миграция добавляет таблицу *Questions* с текстовыми полями *name*, *answer_array* и числовым полям *position*. Первичный ключ, названный *id*, будет добавлен по умолчанию, поэтому его специально не нужно указывать.

Миграция *Questions* («Вопрос») также содержит внешний ключ, который указывает, что данное поле будет связующим и в нем хранится внешний ключ:

```
t.references :presentation, index: true
```

Данная миграция добавляет вопрос в презентацию.

Миграции для модели «Ответы»:

```
class CreateAnswers < ActiveRecord::Migration
  def change
    create_table :answers do |t|
      t.text :text
      t.references :user, index: true
      t.references :question, index: true
      t.timestamps
    end
  end
end
```

Данная миграция добавляет ответ на вопрос в презентации. Эта миграция добавляет таблицу *Answers* с текстовым полем *text*. Первичный ключ(*id*), будет добавлен по умолчанию, поэтому его не нужно указывать.

Миграция ответ также содержит внешний ключ, который указывает, что данное поле будет связующим и в нем хранится внешний ключ:

```
t.references :user, index: true
t.references :question, index: true
```

Миграции для модели «Пользователи»:

```
class CreateUsers < ActiveRecord::Migration
  def change
    create_table :users do |t|
      t.string :login
      t.string :fio
      t.text :groups
      t.string :roles
      t.timestamps
    end
  end
end
```

Данная миграция добавляет таблицу User со строковыми столбцами(логин, пароль, ФИО, группа, роль).

По аналогий создаются миграции *Presentations*.

Описание ассоциаций

В Rails, ассоциация – это связь между двумя моделями Active Record. В проекте представлены три вида связей: один к одному, один ко многим и многие ко многим. Модели которые были созданы используют связь один ко многим. Связь один ко многим в проекте реализуется с помощью `belongs_to` и `has_many`.

-`belongs_to` ассоциация, которая устанавливает соединение с другой моделью, один-к-одному, так что каждый экземпляр модели «принадлежит» один экземпляр другой модели.

- `has_many` ассоциация, которая указывает связь с другой моделью один-ко-многим. Эта ассоциацию находится на «другой стороне» `belongs_to` ассоциации. Она показывает, что каждый экземпляр модели имеет ноль или более экземпляров другой модели.

1.1 Создание вопроса для тестирования

Свойствами объектов данной модели является: название (title), тип ответа (answer_type), выбор ответа (answer_array), позиция (position) и какой презентации относится вопрос.

Структура представлена на рис. 2.

Приложения Переход Система

Edu-online - Mozilla Firefox

Файл Правка Вид Журнал Закладки Инструменты Справка

ФГБОУ ВПО «МГИУ». Корпо... Образовательный портал М... Edu-online

localhost:3000/questions/new?presentation_id=2

Institute Control Engineering and Information Technology, Stud-online

Новый вопрос

презентации Test (2)

Вопрос

После слайда

Тип ответа

ручной ввод

Добавить ответы

Сохранить

Назад

Разработано в ИВЦ МГИУ

© ИВЦ МГИУ, 2014–2015.

Firefox автоматически отправляет некоторые данные в Mozilla, чтобы мы могли улучшить вашу работу в браузере. Выбрать, чем мне поделиться

Edu-online - Mozilla Firefox stud_online_30_12_2014 Терминал

Рис.2 Создание нового вопроса

Модель вопрос (Questions) связана с ответом (answer) связью один ко многим (*has_many*). Тип вопроса определяется способом ответа и может быть одним из следующих:

- 1) выбрать один ответ;
- 2) выбрать несколько ответов;
- 3) выбрать один ответ из нескольких вариантов или написать свой;
- 4) написать свой ответ.

Свойствами объектов данной модели является:

1. вопрос (question),
2. ответ, с которой связан вопрос (question),
3. позиция вопроса в презентации (presentation),
4. тип вопроса (type).

1.2 Создание ответа для тестирования

Модель ответы (Answer) связана с моделью вопрос (Questions) как один ко многим (has_many). В функционале реализовано, что после определенного ответа можно перейти к следующему ответу, а в случае неверного ответа попробовать снова ответить на заданный вопрос или перейти на предыдущий слайд и ознакомиться с материалом презентации лучше или завершить анкетирование досрочно. Добавить ответ к вопросу можно будет, только если тип вопроса это позволяет.

Свойствами объектов данной модели является:

1. вопрос, с которым связан ответ (question),
2. тип ответа (question_type) может быть как 'ручной ввод' или 'выбор из списка'.

Структура представлена на рис. 3.

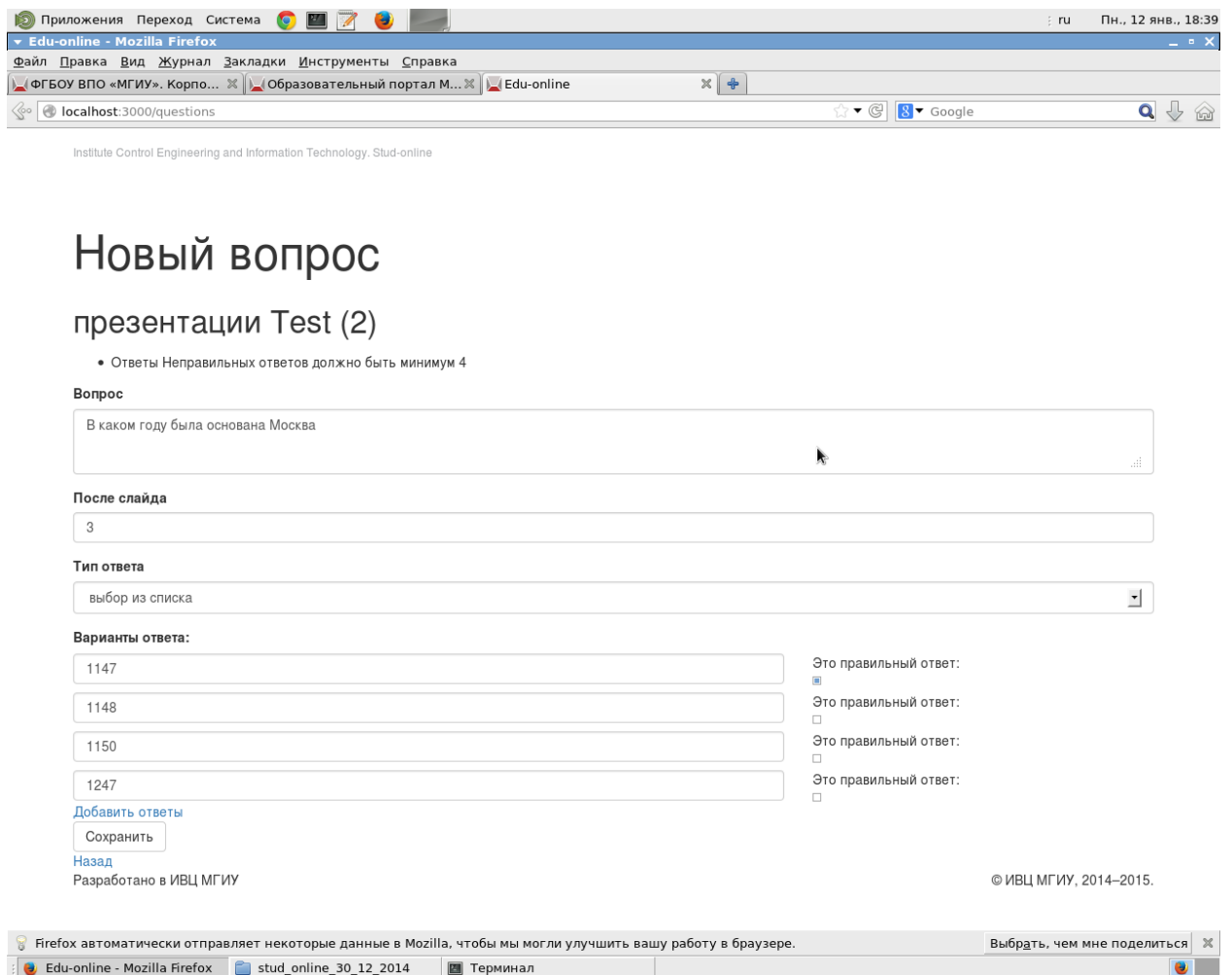


Рис. 3 Создание ответа на вопрос

Глава 2.

1.1 Просмотр всех вопросов

При просмотре всех вопросов отображается «Название», «Презентация», «Дата создания».

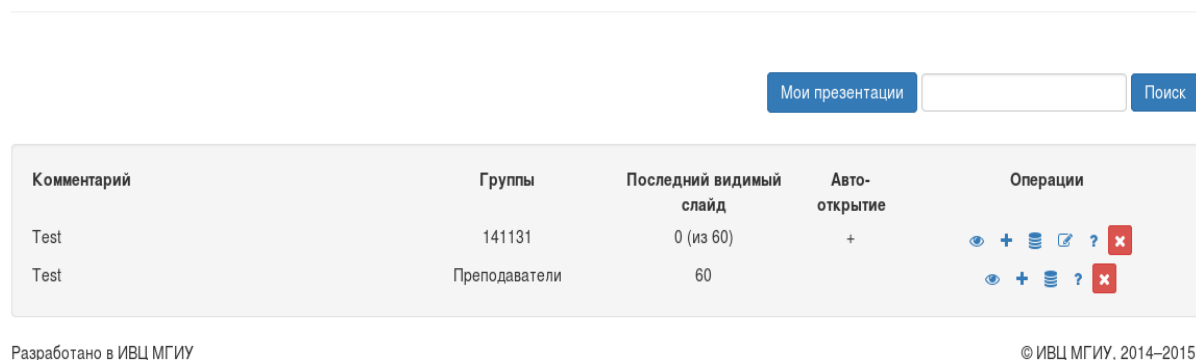
Функционал зависит от пользователя. Преподаватель может:












- 1) создать новый вопрос;
- 2) просматривать вопросы;
- 3) удалять/редактировать свои вопросы;
- 4) получать отчет по тестам (вопросам);
- 5) проходить тесты, если ему назначены.

Студент может только проходить назначенные ему тесты в заданный срок.

На рисунке 4 представлен интерфейс преподавателя.

Все презентации



Комментарий	Группы	Последний видимый слайд	Авто-открытие	Операции
Test	141131	0 (из 60)	+	     
Test	Преподаватели	60		    

Разработано в ИВЦ МГИУ

© ИВЦ МГИУ, 2014–2015.

Рис. 4 Список презентаций (тестов).

1.2 Создание нового вопроса

Создание вопроса происходит максимально просто и удобно для пользователя. Первым шагом создается название вопроса. Далее выбирается способ ответа (question_type) может быть как 'ручной ввод' или 'выбор из списка'.

Добавление вопросов происходит при нажатии кнопки «Добавить вопрос». В вопросе нужно написать в соответствующее поле сам вопрос, выбрать тип вопроса, от выбранного типа вопроса будет зависеть дальнейшие функции, доступные при добавлении вариантов ответа.

На рисунке 5 представлен интерфейс создание новой вопроса.

Новый вопрос

презентации Test (2)

Вопрос

После слайда

Тип ответа

[Добавить ответы](#)

[Назад](#)

Разработано в ИВЦ МГИУ

© ИВЦ МГИУ, 2014–2015.

Рис. 5 Интерфейс создания нового вопроса.

1.3 Отображение созданного вопроса в презентации

При отображении созданного вопроса показывается презентация в которой будет представлен интерфейс под сам вопрос. Способ ответа может быть как 'ручной ввод' или 'выбор из списка'.

На рисунке 6 представлен интерфейс просмотра вопроса и интерфейс.

В каком году была основана москва?

Ответить

Рис. 6 Интерфейс просмотра вопроса

1.4 Редактирование вопроса

Редактирование вопроса позволяет изменить название вопроса, способ ответа, выбрать для какого слайда будет вопрос.

На рисунке 7 представлен интерфейс редактирование вопроса.

Редактирование вопроса

презентации Test (2)

Вопрос

В каком году основана Москва

После слайда

3

Тип ответа

выбор из списка

Варианты ответа:

1147

1148

1150

1247

1231

Это правильный ответ:



Это правильный ответ:



Это правильный ответ:



Это правильный ответ:



Это правильный ответ:



[Добавить ответы](#)

[Сохранить](#)

[Показать](#) | [Назад](#)

Разработано в ИВЦ МГИУ

© ИВЦ МГИУ, 2014–2015.

Рис. 7 Редактирование вопроса

Заключение

Данная работа была полностью реализована, в процессе решения поставленной задачи. Научились работать с множеством новых объектов, таких как: модели, контроллеры, view-файлы, применили на практике теорию нормальных форм и знания по SQL запросам в базу данных и основу CSS. В результате проведенной работе информационная система была успешно создана. Она позволяет пользователю пройти тестирование, как по определенной теме, так и выбрать самому тест. Преподавателю позволяет добавлять новые материалы, которые будут преобразованы с помощью языка разметки (Pandoc markdown) в слайды, которые будут добавлены тесты с определенным количеством вопросов. Администратору информационной системе предоставляется полный доступ ко всей информационной системе:

- 1) может как добавлять студентов или преподавателей в информационную систему, так и их удаление из нее.
- 2) может добавлять презентации, а так же в них и тесты.

Элементы навигации реализованы в удобной для пользователя форме. Система обеспечивает корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входными данными. В указанных случаях система должна выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной команде или некорректному вводу данных.

В процессе работы были применены навыки работы с Ruby on Rails, LaTeX[3], Postgres, объектно-реляционным преобразователем ActiveRecord и написания SQL запросов и материалы которые были взяты из книги С.М.Львовского[5] и книги Д.Е.Кнута.

Информационная система была внедрена, существует акт о внедрении.

Список литературы:

- [1] <http://ru.wikipedia.org/wiki/Ruby> — Википедия о языке Ruby.
- [2] <http://rusrails.ru/active-record-basics> — Перевод Ruby on Rails Guides для версии Rails 4.0.
- [3] D. E. Knuth. The TEXbook. — Addison-Wesley, 1984. Русский перевод: Дональд Е. Кнут. Все про TEX. — Протвино, РДТЕХ, 1993.
<http://www.cs.rhul.ac.uk/~adrian/typesetting/lbook.pdf>
- [4] Тексты дипломных работ, защищенных в МГИУ
(<http://36.msiu.ru/pages/36>).
- [5] Фултон Х. «Программирование на языке Ruby» - М.:ДМК Пресс, 2007. – 688с.: ил.
- [6] Статья о среде Ruby on Rails. -
URL:http://ru.wikipedia.org/wiki/Ruby_on_Rails