Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

**Пермский национальный** **исследовательский политехнический университет**

Факультет ПММ Кафедра ВМиМ

Специальность (направление) ИСТ

**Отчет по проектной работе на тему:**

**«Журнал преподавателя»**

Выполнили:

Студенты группы ИСТ-12

Проверил:

Елисеев А.С.

Пермь 2015 г.

Содержание

[Введение 3](#_Toc440831005)

[Описание проекта 3](#_Toc440831006)

[Основные требования 3](#_Toc440831007)

[Тип разрабатываемой ИС 3](#_Toc440831008)

[Состав команды разработчиков проекта 3](#_Toc440831009)

[Инструментальные средства на этапах жизненного цикла информационной системы 4](#_Toc440831010)

[Сбор требований к системе 4](#_Toc440831011)

[Проектирование 5](#_Toc440831012)

[Разработка 7](#_Toc440831013)

[Тестирование 7](#_Toc440831014)

[Результаты 7](#_Toc440831015)

# Введение

Цель лабораторной работы: познакомиться на практике с современными инструментальными средствами, использующимися при разработке информационных систем (ИС). Для этого требуется реализовать лабораторный проект ИС по индивидуальному техническому заданию (ТЗ).

# Описание проекта

Преподаватель в университете ведет несколько дисциплин. Каждая дисциплина включает лекционные и практические занятия, для которых определены соответствующие подразделы изучаемого предмета. Требуется разработать систему, предоставляющую на основе графика посещаемости конкретного студента по конкретной дисциплине список подразделов для вопросов на экзамене.

## Основные требования

1. Должна присутствовать форма просмотра истории посещения занятий студентами.

2. Должен быть разработан веб-интерфейс для просмотра и редактирования (в т.ч. создания) данных.  
3. Система должна предоставлять список тех подразделов дисциплины, которые студент не посетил / посетил меньше всего.

## Тип разрабатываемой ИС

Клиент-серверная ИС для хранения обработки информации.

## Состав команды разработчиков проекта

* Прототипирование интерфейса – Матвеева Анна, Ефимова Анастасия.
* Верстка – Шушаков Николай.
* Бизнес-логика – Паздерин Альберт.

# Инструментальные средства на этапах жизненного цикла информационной системы

## Сбор требований к системе

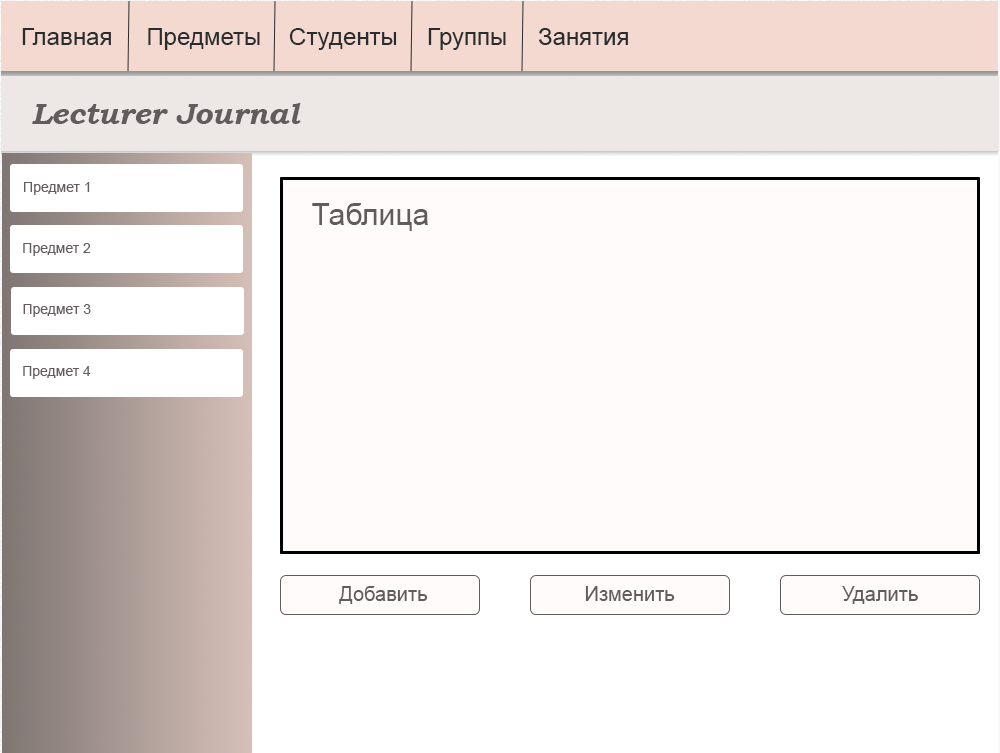
* детализация и фиксация требований к системе  
    
   Требуется реализовать журнал посещаемости в виде веб-интерфейса, включающего следующие страницы:
  + Главная – главная страница с информацией о сути системы.
  + Посещаемость – сводная таблица посещаемости по текущему предмету (строки – запланированные занятия по данному предмету, столбцы – имена студентов). На данной странице должны присутствовать элементы управления, позволяющие переключаться на любой из присутствующих в БД предметов. Далее при наличии в требовании словосочетания «текущий предмет» предполагается наличие данной возможности. На странице также требуется реализация механизма, позволяющего преподавателю отмечать посещаемость студентов.
  + Студенты – на данной странице генерируется список наименее посещаемых каждым студентом тем текущего предмета фиксированной длины.
  + Администрирование – интерфейс администратора.



Use-case диаграмма

* разработка прототипа интерфейса пользователя

Для создания макета веб-интерфейса выл выбран фотошоп. Данный инструмент обладает всем необходимым для создания макетов функционалом и является одним из наиболее часто используемых веб-дизайнерами.



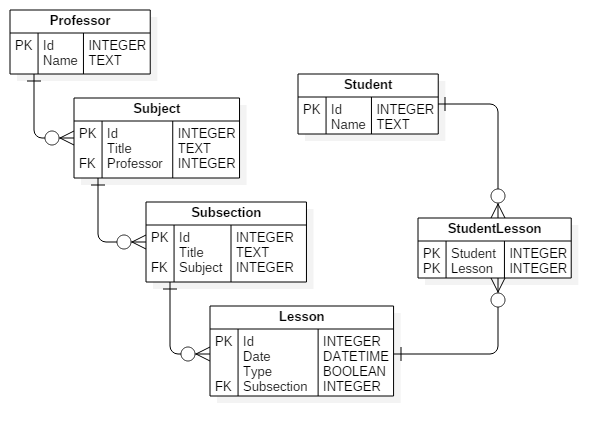
Прототип интерфейса пользователя, спроектированный в фотошопе

## Проектирование

* разработка высокоуровневой архитектура системы, основных компонентов

Выделим основные сущности предметной области и кратко поясним, что они из себя представляют в контексте проектного решения.

* + Профессор – главный актер системы. На данном этапе реализации системы предполагается единственный профессор. Данная сущность добавлена в систему предполагая будущее расширение в виде возможности работы с системой нескольких профессоров.
  + Предмет. Включает ссылку на профессора и название.
  + Тема. Включает ссылку на предмет и заголовок темы.
  + Занятие. Включает ссылку на тему и дату занятия.
  + Студент. Включает данные о студенте.
  + Пометка о посещении – данная сущность является реализацией связи «многие ко многим» между сущностями «студент» и «занятие».



ER-диаграмма

* выбор основных технологий реализации системы;
  + Все диаграммы были нарисованы в CASE-средстве StarUML.
  + Верстка макета осуществлялась в текстовом редакторе Sublime Text 3.
  + Бизнес-логика проекта была реализована с помощью MVC-фреймворка Django. Данный фреймворк используется во многих современных крупных сайтах (например, instagram.com). Несмотря на некоторые отличия от большинства других крупных MVC-фреймворков (например, обработчики URL в Django конфигурируются явно при помощи регулярных выражений, а не выводятся автоматически из структуры моделей контроллеров, другими словами, сущность «контроллер» в привычном понимании в Django отсутствует), фреймворк позволяет реализовывать веб-приложения практически любой сложности.
* выбор системы хранения данных;

Было решено использовать СУБД SQLite ввиду ее простоты и поддержки ее в Django «из коробки».

## Разработка

* выбор IDE;

Для написания кода проекта была применена IDE PyCharm ввиду своей функциональности и удобства. Альтернативами могли служить:

Eclipse + PyDev (медленный и не самый функциональный вариант)

SublimeText 3 + SublimePythonIDE (долгая настройка, но отличная скорость работы, небольшой функционал)

* выбор системы контроля версий VCS;

Выбор системы версий пал на GitHub т.к. проект не является строго приватным, а простота и возможности GitHub очень привлекательны. Также PyCharm поддерживает GitHub.

* выбор инструментов анализа качества кода;

Анализ кода производился встроенными средствами PyCharm.

* выбор инструментов профилирования и анализа производительности;

В данном проекте профилирование и анализа производительности не проводилось, ввиду его простоты и малого размера.

Однако для больших проектов можно применять модуль Django-extensions (коллекция кастомных расширений на Гитхабе).

Анализ результатов профилирования можно проводить при помощи:

* + Pstats - встроенный в Python модуль, выводит результаты в консоль.
  + RunSnakeRun - программа для визуализации результатов профайлинга.
  + Gprof2dot - генерирует картинку с деревом вызовов функций и информацией о времени их выполнения.
* формирование требований к производительности не проводилось.

## Тестирование

* выбор систем автоматического тестирования;

Тестирование проекта производилось при помощи юнит-тестов Django.

* написание модульных и интеграционных тестов;

Были написаны следующие автоматические тесты:

* + добавление занятия и рендеринг страницы «студенты»
  + добавление студента и рендеринг страницы «посещаемость»
  + тест генерации списка тем
  + добавление отметки о посещаемости и рендеринг страницы «посещаемость»
* написание «ручных» тестов;

Ручные тесты системы не писались ввиду достаточности автоматических тестов.

* результаты нагрузочного тестирования и профилирования.

Нагрузочное тестирование и профилирование системы не проводилось.

# Результаты

* Текущая версия проекта реализует все необходимые требования.
* Из положительных особенностей IDE PyCharm следует отметить крайне удобный редактор написания кода на языке Python и наличие средств рефакторинга и анализа кода. Из недостатков: в версии Community отсутствует поддержка Django (подсветка кода языка шаблонов, автозаполнение и т.д.), вследствие чего пришлось идти на некоторые ухищрения.