

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики и  
информационных технологий

**Проектирование и разработка веб-приложения Расписание**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 5 курса 521 группы  
направления 09.03.01 «Информатика и вычислительная техника»

факультета компьютерных наук и информационных технологий  
Новикова Дмитрия Игоревича

Научный руководитель

к. ф.-м.н., доцент

должность, уч. степень, уч. звание

\_\_\_\_\_

дата, подпись

И.Д Сагаева

инициалы, фамилия

Заведующий кафедрой

к. ф.-м.н., доцент

должность, уч. степень, уч. звание

\_\_\_\_\_

дата, подпись

Л.Б. Тяпаева

инициалы, фамилия

Саратов 2017 год

## ВВЕДЕНИЕ

В современном мире у любого университета есть свой сайт и электронное расписание на нем. Век бумажных новостей на стенах прошел. А значит, способы электронного улучшения жизни должны развиваться и совершенствоваться. Расписание – то, без чего не может жить ни один студент. Что на завтра учить? В какой корпус идти? Где найти нужного преподавателя? Эту информацию позволяет получить данное веб-приложение с экрана мобильного телефона. Именно поэтому моя дипломная работа посвящена данной теме.

Целью выпускной квалификационной работы является проектирование и разработка веб-приложения Расписание.

Приложение предоставляет пользователям информацию о времени и месте проведения учебных занятий, справочную информацию для составления учебных планов и отчетов. Также приложение реализует функционал поиска для ускорения процесса обработки информации.

Предполагается, что пользователями приложения будут студенты и преподаватели университета. За поддержание информации в актуальном состоянии должны отвечать администраторы.

Реализация приложения выполнена с использованием новейших технологий и библиотек Spring, Hibernate, Vaadin. Данные технологии применяются в промышленной разработке программного обеспечения. Текущий проект является учебным и позволяет получить опыт создания самостоятельного программного решения, который будет полезен в дальнейшей работе по специальности.

Цели моей работы:

- изучить возможности фреймворков Spring, Hibernate, Vaadin
- разработать серверную и клиентскую часть приложения
- спроектировать и создать базу данных

## **1 Основные понятия**

Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами [1].

Бэкэнд и фронтэнд термины в программной инженерии, которые различают согласно принципу разделения ответственности между внешним представлением и внутренней реализацией соответственно [2].

## **2 Применение веб-приложений**

Веб-приложения позволяют посетителям быстро и легко находить требуемую информацию на веб-сайтах с большим объемом информации. Данный вид веб-приложений позволяет осуществлять поиск в содержимом, упорядочивать содержимое и перемещаться по нему удобным для посетителей способом.

Веб-приложение позволяет сохранять данные непосредственно в базе данных, а также получать данные и формировать отчеты на основе полученных данных для анализа. В качестве примера можно привести интерактивные страницы банков, страницы для контроля товарных запасов, социологические исследования и опросы, а также формы для обратной связи с пользователями [3].

## **3 Структура веб-приложения**

Чаще всего веб-приложения состоят как минимум из трёх основных компонентов:

- 1) Клиентская часть веб приложения - это графический интерфейс. Это то, что вы видите на странице. Графический интерфейс отображается в браузере. Пользователь взаимодействует с веб-приложением именно через браузер, кликая по ссылкам и кнопкам.

2) Серверная часть веб-приложения - это программа или скрипт на сервере, обрабатывающая запросы пользователя (точнее, запросы браузера). Чаще всего серверная часть веб-приложения программируется на PHP. При каждом переходе пользователя по ссылке браузер отправляет запрос к серверу. Сервер обрабатывает этот запрос, вызывая некоторый PHP-скрипт, который формирует веб-страничку, описанную языком HTML, и отправляет клиенту по сети. Браузер тут же отображает полученный результат в виде очередной веб-страницы.

3) База данных (БД, или система управления базами данных, СУБД) - программное обеспечение на сервере, занимающееся хранением данных и их выдачей в нужный момент. В случае форума или блога, хранимые в БД данные - это посты, комментарии, новости, и так далее. База данных располагается на сервере. Серверная часть веб-приложения (то есть, PHP скрипт) обращается к базе данных, извлекая данные, которые необходимы для формирования страницы, запрошенной пользователем.

Браузер через Интернет отправляет HTTP-запросы веб-серверу. Веб-сервер вызывает PHP-скрипт, написанный разработчиком веб-приложения. PHP-скрипт обращается к базе данных, если это нужно. В результате PHP-скрипт возвращает клиенту веб-страницу, которую и отображает браузер [4].

#### **4 Популярные языки программирования для разработки**

Основной язык, которым описывается графический интерфейс веб-приложения - это HTML. Данный язык описывает структуру веб-страницы, размещение на ней компонентов.

Оформление веб-страниц, их стиль и цветовая схема описываются в таблицах стилей – CSS.

Для "оживления" графического интерфейса, придания ему динамичности, используются дополнительные технологии: скрипты JavaScript, а также встроенные в веб-страницу компоненты, созданные на Flash, Java или Silverlight.

Отсутствие необходимости полностью перезагружать страницу после каждого получения данных от сервера может существенно ускорить работу веб-приложения. Такая концепция имеет название *Asynchronous JavaScript and XML* (асинхронный JavaScript и XML, Ajax). При использовании данного подхода динамические запросы к серверу происходят без видимой перезагрузки веб-страницы: пользователь не замечает, когда его браузер запрашивает данные [5].

## **5 Доменные модели**

Доменные объекты — это объекты в объектно-ориентированных компьютерных программах, выражающие сущности из модели предметной области, относящейся к программе, и реализующие бизнес-логику программы. Доменные объекты инкапсулируют всю необходимую для программы информацию об объекте предметной области [6].

## **6 Роль БД при создании веб-приложений**

Согласно классическому определению, база данных — это упорядоченная совокупность информации, хранящейся в виде множеств, каждое из которых содержит записи унифицированного вида. Системы управления базами данных (СУБД) предоставляют программисту мощнейший инструментарий для создания, обновления и обработки больших объемов информации, имеющей сложную структуру.

В классической теории выделяют три типа, три структуры баз данных: иерархическую, сетевую и реляционную. В настоящее время доминирующее положение занимают реляционные базы данных.

Лидером среди баз данных, применяемых для разработки WEB-приложений, на сегодняшний день, безусловно, является MySQL. Главное достоинство MySQL (плавно переходящее в недостаток:) - ее простота. Как следствие - высочайшая скорость выполнения SQL-запросов и необходимость явного программирования основных правил поддержания целостности и непротиворечивости данных на уровне сервера приложений.

Среди других баз данных, применяемых для WEB-разработок, отметим Oracle и PostgreSQL. PostgreSQL - свободно распространяемая СУБД с открытым исходным кодом, ориентированная главным образом на работу в UNIX-подобных системах [7].

## **7 Популярные фреймворки для разработки веб-приложений**

Фреймворк (framework) — это программная оболочка, позволяющая упростить и ускорить решение типовых задач, характерных для данного языка программирования. Само слово framework означает «каркас» в переводе с английского. Действительно, фреймворки и призваны быть готовыми каркасами программ, на которые только и остаётся навесить стены и окна [8].

Сам фреймворк предлагает нам уже встроенные классы для: работы с базой данных, создания функциональных форм, валидации, логирования и др. Все эти классы можете легко использовать во всех ваших проектах, при этом их подключение и использование будет максимально простым.

Еще один из плюсов - структурирование архитектуры вашего приложения [9].

### **7.1 Maven**

Maven (мавен) — это инструмент для сборки Java проекта: компиляции, создания jar, создания дистрибутива программы, генерации документации.

Преимущества Maven:

- 1) Независимость от OS. Сборка проекта происходит в любой операционной системе. Файл проекта один и тот же.
- 2) Управление зависимостями. Редко какие проекты пишутся без использования сторонних библиотек (зависимостей). Эти сторонние библиотеки зачастую тоже в свою очередь используют библиотеки разных версий. Мавен позволяет управлять такими сложными зависимостями. Что позволяет разрешать конфликты версий и в случае необходимости легко переходить на новые версии библиотек.

- 3) Возможна сборка из командной строки. Такое часто необходимо для автоматической сборки проекта на сервере (Continuous Integration).
- 4) Хорошая интеграция с средами разработки. Основные среды разработки на java легко открывают проекты которые собираются с помощью maven. При этом зачастую проект настраивать не нужно - он сразу готов к дальнейшей разработке. Как следствие - если с проектом работают в разных средах разработки, то maven удобный способ хранения настроек. Настроечный файл среды разработки и для сборки один и тот же - меньше дублирования данных и соответственно ошибок.
- 5) Декларативное описание проекта [10].

## **7.2 Vaadin**

Vaadin — свободно распространяемый фреймворк для создания RIA-веб-приложений, разрабатываемый одноимённой финской компанией. В отличие от библиотек на Javascript и специфических плагинов для браузеров, Vaadin предлагает сервер-ориентированную архитектуру, базирующуюся на Java Enterprise Edition. Использование JEE позволяет выполнять основную часть логики приложения на стороне сервера, тогда как технология AJAX, используемая на стороне браузера, позволяет интерактивно взаимодействовать с пользователем, не отставая от аналогичных десктоп-приложений. Для отображения элементов пользовательского интерфейса и взаимодействия с сервером на стороне клиента Vaadin использует Google Web Toolkit [11].

Структурно Vaadin состоит из серверного API, клиентского API, набора компонентов пользовательского интерфейса с обеих сторон, механизма тем для оформления интерфейса и модели данных, позволяющей связывать серверные компоненты непосредственно с данными. Можно применять две основные модели разработки: на стороне сервера и на стороне клиента (браузера).

## **7.3 Spring Framework**

Spring Framework (или коротко Spring) — универсальный фреймворк с открытым исходным кодом для Java-платформы.

Spring предоставляет Java-разработчикам в проектировании; кроме того, он предоставляет хорошо документированные и лёгкие в использовании средства решения проблем, возникающих при создании приложений корпоративного масштаба.

Между тем, особенности ядра Spring применимы в любом Java-приложении, и существует множество расширений и усовершенствований для построения веб-приложений на Java Enterprise платформе. По этим причинам Spring приобрёл большую популярность и признаётся разработчиками как стратегически важный фреймворк.

Одним из компонентов фреймворка является Spring Security, который представляет из себя инструмент, предоставляющий механизмы построения систем аутентификации и авторизации, а также другие возможности обеспечения безопасности для корпоративных приложений, созданных с помощью Spring Framework. [13].

## **8 Практическая часть**

На данный момент разработан функционал серверной и клиентской стороны, который позволяет администрировать и просматривать данные с помощью веб-браузера.

Приложение включает в себя:

Таблицу дисциплин

Таблицу преподавателей

Таблицу студентов

Таблицу учебных групп

Таблицу аудиторий

Таблицу расписания

Таблицу учебного плана

Для каждого элемента таблиц реализовано 4 операции:

Добавление



Чтение

Редактирование

Удаление

В приложении есть возможность отображения расписания, учебного плана, поиска по преподавателю, поиска по группе.

Дальнейшим развитием приложения будет реализация системы регистрации, доработка пользовательского интерфейса, интеграция с мобильными мессенджерами.

Исходных код программы находится на приложенном электронном носителе.

### **8.1 Создание сущностей**

Для установления соответствия между элементами базы данных и POJO классами (определение смотри далее) с помощью фреймворка Hibernate используется термин сущность.

Сущность (Entity) – простой POJO объект, который требуется сохранить в реляционной базе данных. Чтобы объявить класс POJO сущностью, его нужно отметить аннотацией @Entity

Каждая сущность имеет свой первичный ключ. Задать сохраняемому полю свойство первичного ключа можно с помощью аннотации @Id. Сущность управляет его состоянием, используя либо поля, либо get и set методы. Это зависит от того, где используется аннотация.

Для каждой сущности доменной модели создается соответствующий его POJO класс на языке java.

POJO (англ. Plain Old Java Object) — «простой Java-объект в старом стиле», простой Java-объект, не унаследованный от какого-то специфического объекта и не реализующий никаких служебных интерфейсов сверх тех, которые нужны для бизнес-модели.

В текущей реализации POJO класс состоит из:

полей

конструкторов

методов get

методов set

переопределенного метода toString

## 8.2 Проектирование базы данных

Для каждой сущности создается MySQL таблица с колонками, соответствующими полям POJO класса. Отношения между полями показаны на диаграмме отношений.

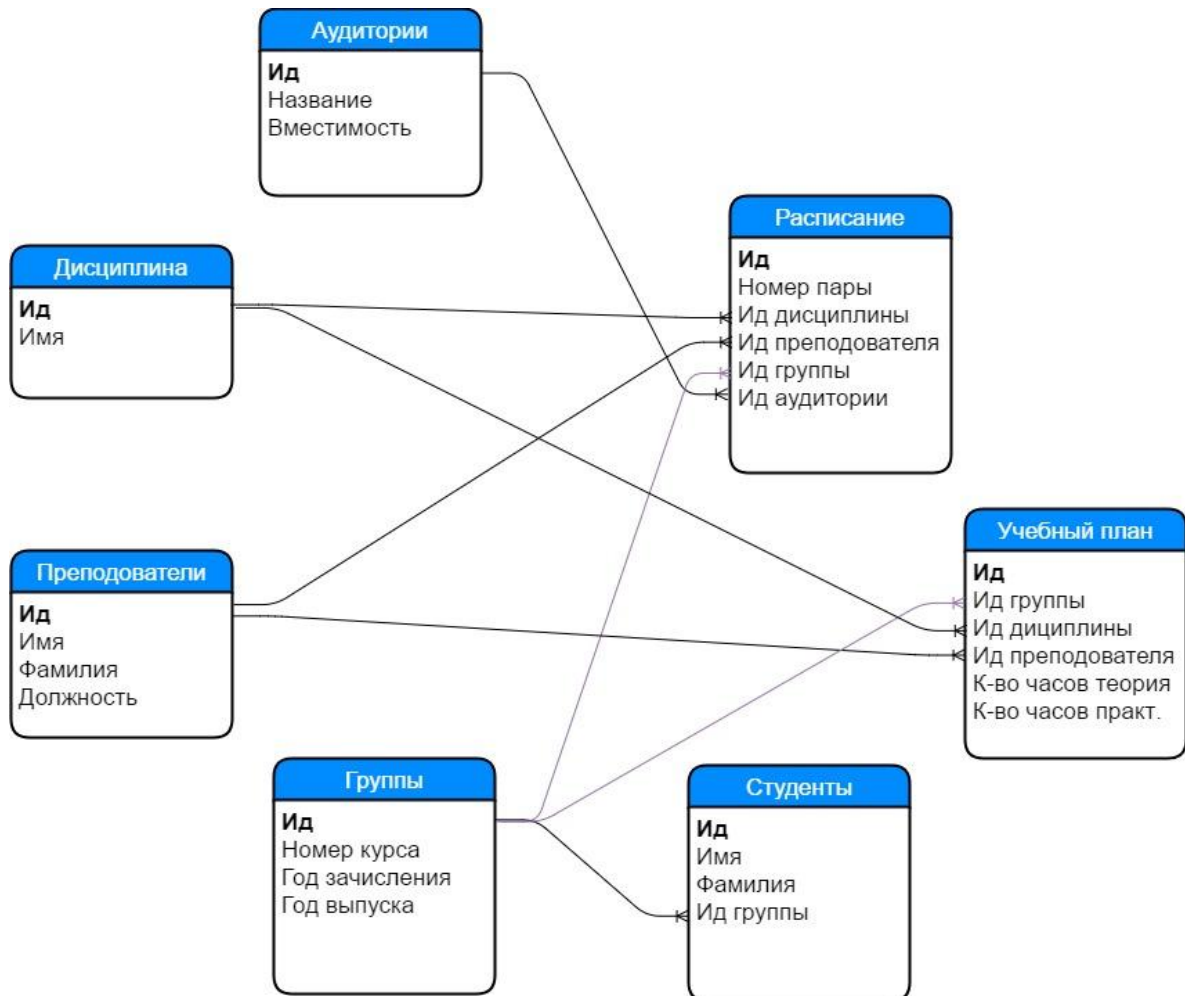


Рисунок 1 – Диаграмма отношений.

## 8.3 Разработка серверной части.

В серверной части реализован запуск веб-сервера, содержащего таблицы для каждой из сущности.

Для проверки работоспособности таблиц, используется протокол REST, представляющий собой обычный http запрос и возвращающий данные таблиц в формате JSON.

#### **8.4 Разработка клиентской части.**

В рамках дипломной работы графический интерфейс представлен для ролей Гость, Администратор. Гость имеет возможность просматривать таблицу расписания и использовать функционал поиска по группе и преподавателю. Администратор имеет возможность редактировать и удалять записи в таблицах, просматривать информацию в таблице учебный план.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения дипломной работы была подробно изучена теория на тему создания веб-приложения, серверной и клиентской его частей. Описаны популярные фреймворки, которые могут использоваться в создании промышленных приложений с высокой степенью загруженности. Так же были изучены фреймворки языка java такие как Spring, Hibernate, Vaadin. При создании приложения использовался сборщик проектов Maven.

Опираясь на изученный материал, была разработана серверная и клиентская часть приложения Расписание, реализован графический интерфейс, а также разработан код на языке java, реализовывающий общение клиент-сервер.

В дальнейшем планируется доработка графического интерфейса, создание системы регистрации и интеграция с мобильными мессенджерами, что повысит актуальность приложения.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Особенности тестирования веб-приложений [Электрон.ресурс] – URL <http://quality-lab.ru/key-principles-of-web-testing/>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 03.12.2016).
- 2 Front end и back end [Электрон.ресурс] –URL [https://ru.wikipedia.org/wiki/Front\\_end\\_%D0%B8\\_back\\_end](https://ru.wikipedia.org/wiki/Front_end_%D0%B8_back_end), свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 05.12.2016).
- 3 О веб-приложениях [Электрон.ресурс] –URL <https://helpx.adobe.com/ru/dreamweaver/using/web-applications.html>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 24.02.2017).
- 4 Структура веб-приложения [Электрон.ресурс] –URL <http://labaka.ru/likbez/struktura-veb-prilozheniya>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 24.02.2017).
- 5 Обзор средств создания интерфейсов веб-приложений на языке Java [Электрон.ресурс] –URL <http://masters.donntu.org/2013/fknt/riabinin/library/article1.htm>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 12.01.2017).
- 6 Доменные объекты [Электрон.ресурс] –URL <http://www.arisfera.ru/glossary/web/Domennye-obekty.html>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 12.01.2017).
- 7 Базы данных для WEB [Электрон.ресурс] –URL <https://www.wwwmaster.ru/bazy-dannyh-dlya-web>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 12.01.2017).
- 8 Фреймворк — что это такое? Определение, значение, перевод [Электрон.ресурс] –URL <https://что-это-такое.ru/framework>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 12.01.2017).
- 9 Что такое фреймворк [Электрон.ресурс] –URL <http://www.dbhelp.ru/what-is-framework/page/>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 12.01.2017).

10 Apache Maven: что это такое и как с ним работать [Электрон.ресурс] –URL <http://brtrg.by/blog/post/54>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 12.01.2017).

11 Vaadin [Электрон.ресурс] –URL <https://ru.wikipedia.org/wiki/Vaadin>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 12.01.2017).

12 Изучаем Vaadin [Электрон.ресурс] –URL <https://habrahabr.ru/company/hacker/blog/244477/>, свободный. – Загл. с экрана. – Яз.рус. – (Дата обращения 12.01.2017).