

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Разработка и реализация модуля «ЕИП СГУ»

для преподавателей, студентов и сотрудников

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 247 группы

направление 09.04.03 — Прикладная информатика

механико-математического факультета

Васильева Виталия Сергеевича

Научный руководитель
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2022

Введение. Магистерская работа посвящена решению проблем автоматизации работы университета и взаимодействия пользователей: сотрудников, студентов и преподавателей.

Создание платформы для онлайн взаимодействия сотрудников университета, преподавателей и студентов очень актуально, в связи с тем, что в настоящее время необходимо хранить и обмениваться большим объемом информации.

Платформа «Единое информационное пространство» - это портал, доступный в локальной сети СГУ им. Чернышевского. Платформа состоит из множества модулей, каждый из которых отвечает за работу определенной части функционала сайта.

Роль платформы не сводится лишь к хранению и обработке информации, а также её упрощает доступность и упрощает обмен информацией между пользователями системы.

Данная работа направлена на реализацию главного модуля платформы «ЕИП» - который, отвечает за хранение и обработку данных сотрудников, преподавателей и студентов.

Цель работы: разработка модуля хранения и обработки данных «ЕИП СГУ» для Саратовского национального исследовательского государственного университета имени Н.Г. Чернышевского.

Для достижения цели необходимо решение следующих задач:

1. Углубленное изучение языка Java и сопутствующих технологий:
 - Spring Framework 5 (Core, MVC, Data, Security)
 - Hibernate;
 - Maven;
 - Tomcat 8.5;
 - PostgreSQL.
2. Изучение языка JavaScript и библиотеки JQuery;
3. Изучение архитектуры Java - приложения, архитектуры «модульный монолит» и специфики работы с сервером приложений.

Основная часть. Магистерская работа состоит из введения, 3 разделов, заключения и списка используемой литературы. Во введении описывается необходимость автоматизации процессов университета в целом, а также формулируется цель работы, и ставятся задачи. Первая глава посвящена выбору инструментов для решения задачи. Проводится сравнительный анализ языков программирования и технологий, подходящих для решения задачи и их подробное описание.

Backend разработка - программирование сайта, в процессе которого web-ресурс наполняется функционалом. Одной из главных функций бэкэнд является работа с пользовательской информацией, получаемой из frontend. Задача разработки в этом случае заключается в преобразовании полученной информации в более понятный для frontend вид.

Существует большое множество языков программирования для backend разработки, каждый из которых специализирован под решение определенной задачи. В качестве языка программирования для backend - разработки мной был выбран язык Java, который идеально подходит под задачи автоматизации.

Frontend - это клиентская сторона пользовательского интерфейса. К этому понятию относят всё, что видит пользователь, загружая веб-страницу: дизайн б сайта, юзабилити, различные формы. Frontend – разработка не может похвастаться таким обширным выбором языков программирования. На текущий момент единственным языком является javascript. JavaScript.

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных. Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов, формирующих таблицу.

Перейдем описанию разработки на Java. Для связи объектов Java - приложения и записей в базе данных используется технология JPA. JPA – это

технология, обеспечивающая объектно-реляционное отображение простых Java - объектов и предоставляющая API для сохранения, получения и управления такими объектами. Сам JPA не умеет ни сохранять, ни управлять объектами, JPA только определяет правила, как что-то будет действовать. Также JPA определяет правила о том, как должны описываться метаданные отображения и о том, как должны работать провайдеры.

Нibernate — самая популярная реализация спецификации Java Persistence Api, предназначенная для решения задач объектно-реляционного отображения (ORM). Нibernate создаёт связь между таблицами в базе данных и Java-классами и наоборот. Это избавляет разработчиков от огромного количества лишней, рутинной работы, в которой крайне легко допустить ошибку и крайне трудно потом её найти.

Чтобы упростить разработку организацией Apache был создан фреймворк Spring. Spring Framework - универсальный фреймворк с открытым исходным кодом для Java-платформы. Фреймворк Spring обладает весьма широкими возможностями. Но в основе практически всех его особенностей лежат несколько фундаментальных идей, направленных на достижение главной цели – упрощение разработки приложений на языке Java.

В связке с Java наиболее распространенной системой сборки является Maven. Apache Maven - фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM (англ. Project Object Model), являющемся подмножеством XML. Maven позволяет выполнять все стадии сборки проекта: компиляция, создание артефактов, создания дистрибутива программы, генерации документации.

После того, как приложение готово, его необходимо упаковать в артефакт и загрузить на удаленное хранилище, которое называется сервером приложений. Сервер приложений - это набор физического и программного обеспечения, которое способно обеспечить доступ клиентов к программам, выполняющимся непосредственно на серверном оборудовании.

Во второй главе описывается процесс разработки приложения «Единое информационное пространство СГУ». Приводится описание архитектуры приложения, разработка базы данных, описание слоев всей системы и java-приложения в частности. Также описана настройка конфигурации готового приложения.

Для организации компонентов всего приложения используется классическая трёхуровневая архитектура. Трёхуровневая архитектура — архитектурная модель веб - приложения, которая состоит из трёх слоёв: слой клиента, слой бизнес логики и слой данных.

Рассмотрим каждый слой более подробно:

1. Клиентский слой — интерфейс пользователя. Это может быть веб-браузер, которому отправляются HTML-страницы, мобильное или графическое приложение, написанное с помощью JavaFX или Swing. Главное назначение этого слоя - отправка запросов на сервер и обработка ответов от него.

2. Слой логики — сервер, на котором происходит обработка запросов и ответов клиента. Здесь происходят все логические операции: математические расчеты, операции с данными, обращения к другим сервисам или хранилищам данных.

3. Слой данных — сервер баз данных. В этом слое сохраняется вся необходимая информация, которой пользуется приложение при работе. реализуется, как правило, средствами систем управления базами данных, подключение к этому компоненту обеспечивается только с уровня сервера приложений.

По требованиям проекта необходимо хранить в базе информацию о студентах, преподавателях и сотрудниках, а также принадлежность студентов и преподавателей к учебным группам.

Таким образом, таблица «Groups» содержит следующие поля:

- поле «group_id» - первичный ключ таблицы;
- поле «title» - наименование группы.

Таблицы «Students», «Teachers» и «Employees» имеют общую структуру полей:

- поле «id» - первичный ключ таблицы;
- поле «fam» - фамилия;
- поле «name» - имя;
- поле «patronym» - отчество;
- поле «dateOfBirth» - дата рождения;
- поле «phoneNumber» - номер телефона.

Однако для связей таблиц «Students» и «Teachers» с таблицей «Groups» необходимо создание двух дополнительных таблиц:

1. Таблица «student_groups» - хранит информацию о принадлежности студента к определенной группе. Каждый студент может числиться только в одной группе, но в группе, как правило числиться несколько студентов. Связь один ко многим.

- поле «group_id» - идентификатор группы;
- поле «student_id» - идентификатор студента.

2. Таблица «teachers_groups» - хранит информацию о принадлежности преподавателя к определенной группе. Каждый преподаватель может числиться в нескольких группах и в каждой группе может быть несколько преподавателей. Связь многие ко многим.

- поле «group_id» - идентификатор группы;
- поле «teacher_id» - идентификатор преподавателя.

Перейдем к описанию слоёв Java - приложения.

Каждый Java проект имеет определенную структуру файлов - папки и пакеты именованы специальным образом и имеют определенную иерархию. Хорошей практикой является использование стандартной иерархии, создав проект с помощью одного из артефактов Maven, например - «maven-archetype-webapp». Это нужно для поддержания единого стиля разработки, а также для того, чтобы было проще ориентироваться разработчикам, которые будут продолжать работу

над проектом. Если файловая структура имеет стандартный вид, то система сборки уже знает, где искать нужные файлы конфигурации, исходные коды и т.д. Таким образом разработчику не придется прописывать никаких дополнительных настроек. По умолчанию, проект на Java с использованием Maven имеет папку «src» - место, где хранятся исходные коды приложения, а также файл «pom.xml» - файл системы сборки Maven. Внутренняя структура папки с исходными кодами может очень сильно различаться в зависимости от проекта.

В папке java хранятся исходные коды приложения. Исходный код для удобства разбит по пакетам, которые называются слоями. Каждый слой отвечает за решение определенных задач.

После того, как приложение готово, его необходимо упаковать в артефакт и загрузить на удаленное хранилище, которое называется сервером приложений.

Сервер приложений - это набор физического и программного обеспечения, которое способно обеспечить доступ клиентов к программам, выполняющимся непосредственно на серверном оборудовании. Это некий комплект, к которому можно получить доступ при помощи специальных API и обеспечить не только вывод каких-либо данных, но и оптимизировать исполнение программного кода на любых устройствах.

Основное назначение сервера приложений приведено ниже:

- Механизм уменьшения размера и сложности клиентских программ.
- Для необходимости кэширования и контроля потока данных для повышения производительности.
- Механизм для обеспечения безопасности данных, а также трафика конечного пользователя.

Хотя они могут иметь различное назначение, не все серверы приложений используются для одного и того же набора функций.

Сервера приложений отличаются по новизне, стоимости и набору поддерживаемых технологий.

Для относительно небольших проектов на Java используется бесплатный сервер приложений Tomcat. Рассмотрим его поподробнее.

Tomcat — это контейнер сервлетов с открытым исходным кодом, который также выполняет функцию веб-сервера.

Реализует спецификацию сервлетов, спецификацию JavaServer Pages (JSP) и JavaServer Faces (JSF). Написан на языке Java.

Tomcat позволяет запускать веб-приложения и содержит ряд программ для самоконфигурирования.

Для разработчиков контейнеры сервлетов имеют огромную роль, так как решают целый спектр проблем. Для коммерческой Enterprise на Java Tomcat обеспечивает несколько серьезных преимуществ:

Tomcat может выполнять роль полноценного web-сервера. Благодаря этому развертывание приложений происходит гораздо быстрее, чем было раньше. Также Tomcat может выступать в роли классического контейнера сервлетов, который можно встроить в более крупный веб-сервер.

Tomcat способствует обмену данными между клиентом и сервлетом, обеспечивает программный контекст для веб-приложений, берет на себя функции идентификации и авторизации клиентов, создание сессий для каждого из них.

Предоставляет очень удобное управление JSP-страницами. Фактически веб-сервер может отдавать только html. Jasper, компонент Tomcat, компилирует код, содержащийся в шаблонных тегах JSP-страниц в корректный HTML-файл, который получает клиент. Вся логика шаблонизации находится внутри Tomcat, позволяя добавлять даже сложные конструкторы в классическую форму HTML.

Итак, приложение готово, далее необходимо настроить конфигурацию запуска. Перед тем как начать настройку запуска проекта, необходима установка компонентов, без которых наше приложение не сможет запуститься, а именно:

- Java 8;
- Система сборки Maven 3;
- Среда разработки IntelliJ Idea;
- Сервер приложений Tomcat 8.5;
- СУБД PostgreSQL.

После того, как все необходимые программы установлены, можно приступить к конфигурированию запуска готового приложения.

1. Исходные коды приложения хранятся в репозитории на GitHub, поэтому, для начала необходимо загрузить проект на компьютер. Для этого, в IntelliJ Idea нужно выбрать пункт меню «File» - «Get From Version Control».

2. В поле «URL» необходимо ввести адрес репозитория с проектом - <https://github.com/vytsalo/testproject.git>.

3. В поле «Directory» нужно указать путь, где будет храниться проект на вашем компьютере. Далее необходимо нажать кнопку «Clone».

Откроется новое окно, где будет загружаться проект и его зависимости. Это займет некоторое время. По умолчанию, при выгрузке из репозитория пользователь будет находиться в ветке «master», если необходимо сменить ветку, то это можно сделать в нижнем правом углу, нажав на соответствующую иконку. После выбора ветки необходимо выбрать команду «checkout» и изменения будут загружены поверх текущей ветки, т.е. Git перезапишет нашу локальную копию.

Далее, необходимо настроить конфигурацию запуска. Для этого нужно нажать на кнопку «Add Configuration», которая находится в верхнем меню. В появившемся окне выбираем «Add new run Configuration», выбираем Tomcat Server (Local). Откроется окно конфигурации запуска.

В секции «Before launch» нажимаем на «плюс», и выбираем «Build artifacts». Выбираем артефакт «eip-core-module:war exploded». Нажимаем «ok» и «apply». Далее, переходим на вкладку «Deployment». В секции «Deploy at the server startup» выбираем артефакт «eip-core-module:war exploded». В поле Application context ставим «/». Нажимаем «apply» и «ok». Настройка завершена.

Третья глава посвящена описанию процесса взаимодействия пользователя с определенной страницей веб – приложения.

Страницы для обработки информации по преподавателям, студентам и сотрудникам имеют единый интерфейс, поэтому, для примера рассмотрим только страницы студентов.

Описание состоит из следующих страниц:

1. Главная страница;

С этой страницы можно перейти на одну из трех вкладок – «Студенты», «Группы» или «Преподаватели», для этого нужно нажать на кнопку «Перейти» в соответствующей категории. Также, на каждой странице в её верхней части находится «шапка», в которой доступны две функциональные кнопки - левом углу можно нажать на надпись и логотип - это ссылка, которая ведет на главную страницу сайта, а нажав на кнопку «Войти в систему» - можно авторизоваться, если вы зарегистрированы в системе.

2. Страница «Список студентов»;

На странице представлены данные в виде таблицы со следующими полями: фамилия, имя, отчество, дата рождения, телефон, группа. Название группы является ссылкой, нажав на которую пользователь попадет на страницу редактирования и просмотра информации о конкретной группе. Далее следует столбец «Действия», в которой находятся функциональные ссылки для каждой из записей в таблице: редактировать, удалить, информация. Слева от таблицы находится функциональный блок для работы с таблицей студентов, там находятся следующие кнопки: добавить, поиск, импорт, экспорт. Также на странице доступна пагинация - на каждой странице выводится по 10 записей с информацией по студентам. Переключение осуществляется нажатием на соответствующую кнопку с номером страницы.

3. Страница «Добавление студента»;

На данной странице расположена форма для добавления нового студента в базу данных. К форме подключена клиентская валидация на проверку правильности ввода.

4. Страница «Личное дело студента»;

На странице редактирования можно изменить любые данные по студенту, если они удовлетворяют бизнес требованиям и проходят валидацию. Также на данной странице можно определить студента в группу. Для этого необходимо нажать на поле «Группа», откроется окно с поиском, куда необходимо ввести

наименование группы, а потом сделать выбор из предложенных вариантов. После завершения редактирования для принятия изменений, необходимо нажать кнопку «Отправить».

5. Страница авторизации.

Для получения разрешения на любые действия, связанные с редактированием данных, например - изменение данных или удаление студентов из базы необходима авторизация под учетной записью администратора. Авторизация доступна с каждой страницы в верхней панели, справа от логотипа «Единое информационное пространство». Можно авторизоваться сразу с главной страницы, либо система сама предложит вам авторизоваться, когда вы попытаетесь выполнить действие, для которого не хватает ваших привилегий. После ввода логина и пароля, в случае, если авторизация прошла успешно, пользователь вернётся на страницу, откуда пришел. В случае ввода неверных данных, появится уведомление об ошибке.

Заключение. В ходе работы были рассмотрены следующие инструменты для решения задачи: инструменты для Backend и Frontend разработки, системы управления базами данных, спецификация JPA, библиотека Hibernate, фреймворк Spring и его составляющие, система сборки проектов Maven, сервер приложений Tomcat.

Было произведено сравнение существующих баз данных и JavaScript - библиотек. Описана архитектура проекта, назначение слоев приложения, схема базы данных. А также предоставлено подробное описание настройки конфигурации проекта и описание работы готового приложения. Все поставленные задачи были выполнены, разработан главный модуль хранения и обработки данных приложения «ЕИП СГУ» для преподавателей, студентов и сотрудников Саратовского национального исследовательского государственного университета имени Н.Г. Чернышевского. При разработке были использованы следующие технологии: Java 8, Spring 5, Hibernate 4, PostgreSQL, Maven 3, Tomcat 8.5, JavaScript, JQuery.

Разработанное приложение «Единое информационное пространство СГУ» позволит автоматизировать работу университета, упростить взаимодействие между пользователями, а также предоставит удобный интерфейс для работы системы электронного документооборота.