

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра Математического и компьютерного моделирования

Разработка информационной системы поддержки

работы Диссертационного совета

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студентки 2 курса 247 группы

направление 09.04.03 — Прикладная информатика

механико-математического факультета

Зиновьева Валентина Сергеевича

Научный руководитель

профессор, д.ф.-м.н., доцент

Д.В. Кондратов

Зав. кафедрой

зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2023

Введение. Диссертационные советы являются важнейшим звеном в подготовке кадров высшей научной квалификации.

Диссертационные советы создаются Высшей аттестационной комиссией Минобрнауки России и в своей деятельности подчинены ей. Вместе с тем диссертационные советы находятся в прямой связи с руководящими органами организаций, при которых они функционируют. Ответственность диссертационных советов одновременно является и ответственностью ректоров учебных заведений и дирекций научных организаций. В данном случае руководители этих учреждений не вмешиваются в деятельность диссертационных советов, по оценке диссертаций. Вместе с тем они не только имеют право контролировать работу кафедр и советов в части соблюдения нормативных документов Правительства, Минобрнауки России и ВАК, но и несут ответственность за строгое следование государственным документам.

В диссертационных советах существует проблема учета документов необходимых для защиты диссертационных работ. Для защиты диссертации необходимо собрать определенный пакет документов. Некоторые документы не предоставляются в нужный срок, из-за чего появляется проблема в учете документов еще до защиты диссертационных работ. Для того, чтобы исключить вероятность просрочки сдачи документов, следует разработать веб-приложение, которое будет осуществлять контроль над сроками сдачи этого пакета документов.

Данная магистерская работа посвящена разработке информационной системы и веб-приложению учета документов и их сроков сдачи, необходимых для процедуры защиты диссертационных работ. Веб-приложение будет написано на языке программирования Java с использованием Spring Framework. Фронтенд разработка будет выполнена с помощью фреймворка Vaadin. Веб-приложение будет общаться с базой данных с помощью spring data JPA. Весь проект собирается с помощью Maven - инструмента для автоматической сборки проектов на основе описания их структуры в специальных файлах на языке POM (Project Object Model) — подмножестве XML.

Целью работы является разработка информационной системы поддержки

работы диссертационного совета.

Для этого должны быть решены следующие задачи:

- анализ предметная область;
- анализ возможных средств разработки, на основе которого выбрать инструментарий для реализации задачи;
- выбрать БД подходящего типа, спроектировать и реализовать ее структуру;
- выполнить реализацию проекта.

Во **введении** формируется цель работы и решаемые задачи.

Первый раздел посвящен объяснению разницы между защитой в ВАКовском Диссертационном совете и Диссертационном совете, созданном ВУЗом.

Диссертационный совет при ВАК – это особая структура, предназначенная для предварительной оценки научных работ, оценки соискателя и представления его кандидатуры на присвоение ученой степени. Она создается при учебном заведении (ВУЗе), где действует аспирантура и/или докторантура. Состав этого аттестационного органа формируется и утверждается высшей аттестационной комиссией. Членами Диссовета могут быть только квалифицированные эксперты в конкретной области: кандидаты и преимущественно доктора наук.

Структура Диссовета при ВАК включает в себя председателя, ученого секретаря, заместителя председателя и членов Диссовета. Притом количество компетентных экспертов должно быть не менее 19. У каждого из них должно быть профильное образование и «научная репутация». В одном Диссовете могут числиться профессионалы по трем специализациям (не более того) притом по каждой «оцениваемой области» должно быть не менее 7 ученых. Далек не каждый желающий может претендовать на «пост» в Диссовете. В это ведомство не попадут лица, занимающие рабочее место в муниципальных и государственных структурах, госслужащие. ВАК тщательно контролирует деятельность Диссоветов: утверждает его состав, изучает аттестационные дела соискателей, проверяет грамотность и легальность действий ВАК, аргументированность его решения относительно судьбы научной работы и ее автора и пр. Раз в 3-5 лет члены Диссовета проходят переаттестацию, подтверждая тем самым соответствие требованиям высшей

аттестационной комиссии: наличие актуальных работ в рамках определенной научной плоскости, публикационная активность (научные статьи, монографии, учебники и методички и пр.), работа в научной организации и пр. В большинстве случаев Диссоветы действуют в изначально утвержденном составе около 5 лет: каждый эксперт продлевает срок работы путем успешной переаттестации.

Один ученый может работать одновременно в трех Диссоветах. Деятельность каждого из них тщательно изучается и контролируется, при выявлении нарушений может быть возобновлено досрочное прекращение полномочий эксперта и «переутверждение» состава Диссовета. Как мы видим, попасть в диссертационный совет, подведомственный ВАК, непросто. Необходимо постоянно развиваться, пополнять свою «научную копилку», делиться опытом с молодыми специалистами, грамотно и объективно оценивать научные работы, аргументировать свою позицию, следовать четким регламентам и стандартам. На сегодняшний день в России насчитывается около 2000 действующих Диссертационных советов, созданных и контролируемых ВАК.

Второй раздел посвящен выбору платформы для разработки веб-приложения.

В качестве платформы для разработки веб-приложения был выбран Java.

Java – строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems.

Это один из наиболее известных и популярных языков, используемый во многих компаниях, в том числе AliExpress, Google, Intel, eBay и во многих других. С момента появления в 1995 году он только набирал популярность, став лидером к концу 2010-х наряду с Python.

Написать на языке можно практически все что угодно:

- приложения для Android;
- промышленные программы;
- десктопные приложения;
- банковские программы;
- программы для работы с Big Data;
- научные программы;

- корпоративный софт;
- веб-сервера, веб-приложения, сервера приложений;
- встроенные системы – от небольших чипов до специальных компьютеров.

Мультифункциональность языка Java заключается в том, что она не привязана ни к одной из популярных платформ. С одинаковым успехом его можно использовать на Windows или iOS, Linux или Android.

Строгая (сильная) типизация не позволяет смешивать в выражениях различные типы и не выполняет автоматически неявные преобразования.

Java создан по модели объектно-ориентированного программирования. Здесь есть классы и объекты. Классы – типы данных, а объекты – представители классов. Разработчик создает их и сам, дает названия и присваивает им свойства и операции, которые можно выполнять с ними. Это напоминает конструктор, который позволяет построить то, что хочется.

У Java выделяется следующий ряд плюсов:

- мультифункциональность;
- достаточно простой синтаксис;
- независимость (код сможет работать на любой платформе, поддерживающей Java);
- надежность, благодаря строгой статической типизации;
- возможность параллельной разработки;
- хорошую организацию кода и его легкую поддержку;
- универсальность использования различных классов.

Если говорить о конкурентах Java, то в первую очередь вспоминают Python. Языки соревнуются много лет, постоянно опережая друг друга в рейтингах. Но различия все же есть.

У Java типизация статическая – при вводе переменной необходимо прописывать тип данных. У Python – динамическая: типы данных определяются автоматически. Кроме того, в Python можно смешивать разные типы переменных.

Еще одно отличие – читаемость. Это отличие связано с предыдущим, потому, что то же указание типов оказывает влияние на конечный вид кода. У Java простой синтаксис по сравнению с другими языками, но у Python – еще проще.

У синтаксиса в Python ясная и простая структура, его можно даже назвать интуитивным. Язык подойдет новичкам. В то время, как код в Java более комплексный, со множеством знаков и слов: на английский подобный синтаксис называют verbose, то есть многословный код. Он сложнее читается и может вызвать трудности у новичков, хотя многие разработчики чувствуют себя более комфортно со строгим синтаксисом.

Python отстает от Java по производительности и скорости. В обоих языках программирования приложения переводятся в байт-код, чтобы быть кроссплатформенными, но различия в том, когда это происходит. Java компилирует с помощью JIT-компиляции, заранее, а Python – при выполнении программы. Поэтому Java намного быстрее.

Второй возможный конкурент Java – язык программирования C#. У них есть явное сходство: они оба объектно-ориентированные со статической типизацией. Кроме того, имеют похожий синтаксис.

Один из важных параметров языков программирования – типобезопасность (безопасность системы типов). В безопасном языке после проверки согласования типов в них не возникнет ошибок во время выполнения.

Третий раздел посвящен описанию фреймворка Spring для языка программирования Java.

Spring, или Spring Framework – фреймворк для языка программирования Java. Он нужен для того, чтобы разработчикам было легче проектировать и создавать приложения. Spring не связан с конкретной парадигмой или моделью программирования, поэтому его могут использовать как каркас для разных видов приложений.

Чаще всего Spring используется в крупных корпоративных проектах: это характерно для Java и связанных с ним инструментов. Но благодаря универсальности применять Spring можно и в других случаях. Его задача – дать разработчикам больше свободы в проектировании и реализации.

Spring пользуются следующие лица:

- Java-разработчики, занимающиеся созданием десктопных, мобильных или веб-

приложений, - Spring можно использовать в любом из направлений. Чаще всего речь идет о проектах enterprise-масштаба.

- Kotlin-разработчики, так как этот язык написан на основе Java и использовался для создания Spring.

- Системные архитекторы и инженеры, основная задача которых – проектировать будущие проекты. Spring можно использовать как каркас, шаблон для приложения, поэтому он важен в проектировании и анализе.

- иногда – бэкенд-разработчики, так как в некоторых случаях серверная часть сайта может быть написана на Java.

В начале работы с фреймворком можно перепутать Spring и Spring Boot. Это не отдельный фреймворк, а дополнение к Spring, которое облегчает работу с ним. Spring нужно конфигурировать для каждого нового проекта. Конфигурация может занять много времени и не дать ощутимые преимущества в дальнейшей работе. Чтобы исправить проблему, был создан Spring Boot. Он включает в себя комплекс утилит для автоматизации настройки.

Spring Boot:

- автоматически конфигурирует проекты на основе одного из стартовых пакетов для них;

- облегчает создание и развертывание приложений на Spring;

- быстро и легко управляет зависимостями и подгружает необходимые модули;

- поддерживает встроенный сервер для запуска приложений;

- может автоматически создать и настроить базу данных для приложения.

При необходимости настройки Spring Boot можно изменить, чтобы он конфигурировал и настраивал компоненты иначе. Набор утилит облегчает работу со Spring, поэтому некоторые tutorиалы по умолчанию написаны с учетом использования Spring Boot.

Благодаря огромному количеству компонентов и технологий, которые поддерживает Spring, он универсален. Его можно использовать комплексно, например, для разных частей архитектуры MVC или решения других сложных задач. Не нужно собирать большой стек технологий. Если чего-то окажется недостаточно,

помогут дополнительные инструменты и другое ПО экосистемы Spring.

Цель любого фреймворка – делать работу эффективнее и быстрее. То, на что бы ушло несколько месяцев на «чистом» языке, с фреймворком делается за считанные дни, а в коммерческой разработке время имеет ключевое значение. Фреймворк – как каркас, вокруг которого пишется остальной код программы. Некоторые компоненты и логика уже реализованы, и задача программиста – в том, чтобы грамотно ими воспользоваться.

Четвертый раздел посвящен описанию фреймворка Vaadin для языка программирования Java.

Vaadin специально ориентирован на доступность UX. Он позволяет программисту создавать настраиваемые веб-приложения, а пользователям – получать доступ к DOM непосредственно с виртуальной машины Java. Приложение на Vaadin запускается как обычное Spring Boot приложение, без каких-либо дополнительных действий.

Как Java, так и HTML, можно использовать для построения представлений во фреймворке Vaadin. У него интуитивно понятный пользовательский интерфейс с поддержкой взаимодействий перетаскивания. Это значительно упрощает весь процесс веб-разработки для программистов любого уровня подготовки. Также Vaadin – платформа с открытым исходным кодом, включающая в себя веб-фреймворк Java, набор веб-компонентов, средства запуска приложений и инструменты.

Maven – это инструмент для автоматической сборки проектов на основе описания их структуры в специальных файлах на языке POM (Project Object Model) — подмножестве XML.

Чаще всего Maven используют Java-разработчики, однако после установки дополнительных плагинов его также можно применять для работы на PHP, Ruby, Scala, C/C++ и Groovy.

Функциональность системы сборки Maven шире, чем компилятора исходного кода. В процессе работы приложения Apache Maven вызывает компилятор и при этом автоматически управляет зависимостями и ресурсами.

Пятый раздел посвящен разработке информационной системы.

В диссертационных советах существует проблема, связанная со сроками сдачи определенных документов. Для того, чтобы исключить вероятность превышения сроков сдачи документов, было разработано веб-приложение.

Данное веб-приложение позволяет вести учет защищающих работу для контроля сроков сдачи документов. Приложение позволяет вести собственную базу защищающихся. Для каждого защищающегося создана собственная страница для просмотра документов, которые он сдал или не донес. Напротив каждого типа документа, с помощью математических вычислений, выставлен срок – до какого числа, месяца и года не приложенный документ должен быть сдан. На экране также можно скачать памятку по срокам для докторской и кандидатской диссертаций. Существует возможность сформировать списки защищающихся по любому из тех полей для заполнения:

- ФИО защищающегося;
- Специальность;
- Дата защиты.

Для того, чтобы добавить карточку нового защищающегося необходимо для начала создать список специальностей. Это сделано для того, чтобы в ходе многократного создания карточек защищающихся, специальность можно было проставить, а не набирать заново. Также предусмотрена возможность не только добавить карточку, но и удалить ее.

Заключение. В рамках магистерской работы решена задача по реализации информационной системы, представляющей собой веб-приложение учета документов и сроков их сдачи, необходимых для процедуры защиты диссертационных работ. Веб-приложение написано на языке программирования Java с использованием Spring Framework. Разработка фронтенда выполнена с помощью фреймворка Vaadin.

Были решены следующие задачи:

- проанализирована предметная область;
- выполнен анализ возможных средств разработки, на основе которого выбран

инструментарий для реализации задачи;

- выбрана БД подходящего типа, спроектирована и реализована ее структура;

- выполнена реализация проекта.

Спроектированная информационная система имеет весь заявленный в задании функционал.