Утверждаю

Руководитель предприятия

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, печать предприятия)

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_г.

Учреждение образования

«Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра «Информационные системы и технологии»

Специальность «Информационные системы и технологии»

Направление специальности 1.40-05 01-03 Издательско-полиграфический комплекс

**ОТЧЕТ**

по производственной общеинженерной практике

в БГТУ”

20.03.17 - 16.04.17

(наименование предприятия, сроки практики)

Исполнитель

студент 4 курса 2 группы Прокопович Д. В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись, дата)

Руководитель практики от предприятия

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Смелов В. В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(печать предприятия) (Ф.И.О.) (подпись, дата)

Руководитель практики от университета

Савинко А.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись, дата)

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Минск 2017

Содержание

[Общая характеристика предприятия 3](#_Toc480064220)

[**Общие сведения о предприятии, его структуре.** 3](#_Toc480064221)

[**Применяемые информационные технологии.** 3](#_Toc480064222)

[**Разрабатываемое или применяемое ПО** 3](#_Toc480064223)

[Ключевые возможности Oracle Database 4](#_Toc480064224)

[Введение 5](#_Toc480064225)

[**2.1.** **Аналитический обзор литературы** 6](#_Toc480064226)

[**2.1.1.** **СУБД** 6](#_Toc480064227)

[**2.1.2.** **Веб-сервер** 6](#_Toc480064228)

[**1.2.** **Стек технологий LAMP** 7](#_Toc480064229)

[**1.3.** **Стек технологий JEE** 7](#_Toc480064230)

[**1.4.** **Стек технологий Node.js** 8](#_Toc480064231)

[**1.5.** **Стек технологий .NET** 8](#_Toc480064232)

[1. Патентный поиск и аналоги программного обеспечения 8](#_Toc480064233)

[3.1 Программное обеспечение мониторинга заданий резервного копирования (Simple Backup Monitor IT) 9](#_Toc480064234)

[3.2 Программа для разработки и администрирования баз данных InterBase/Firebird "EMS SQL ManagerforInterBase/Firebird 9](#_Toc480064235)

[3.3 Программное средство «Система управления базами данных Синтез» 10](#_Toc480064236)

[Заключение 11](#_Toc480064237)

[Список используемых источников 12](#_Toc480064238)

# Общая характеристика предприятия

## **Общие сведения о предприятии, его структуре.**

Белорусский государственный технологический университет - это престижный и динамично развивающийся учебно-научно-производственный центр Республики Беларусь, строящий свою деятельность на инновационных технологиях мирового уровня.

Университет готовит инженеров-программистов, web-дизайнеров, редакторов-технологов, инженеров-технологов, инженеров-экономистов, менеджеров, маркетологов, инженеров по стандартизации и сертификации, инженеров-экологов, специалистов по автоматизации процессов производств и в сфере туризма, инженеров-энергоменеджеров, инженеров-механиков; специалистов для следующих отраслей промышленности: химической и нефтехимической; фармацевтической и целлюлозно-бумажной; строительных материалов; ландшафтной архитектуры, лесной и деревообрабатывающей, лесному хозяйству; электрохимической; энергетической; полиграфической; парфюмерно-косметической; туристической индустрии; биотехнологии и биоэкологии; охране окружающей среды и промышленной экологии.

## **Применяемые информационные технологии.**

Белорусский государственный технологический университет применяет следующие относящиеся к теме дипломного проекта технологии:

* БД Oracle;
* БД MSSQL;
* ASP.NET.

## **Разрабатываемое или применяемое ПО**

## **Microsoft Visual Studio**

MicrosoftVisualStudio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии WindowsForms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, WindowsMobile, Windows CE, .NET Framework, Xbox, WindowsPhone .NET CompactFramework и Silverlight [4].

В учреждении используется VisualStudioCommunity 2015, преимуществом которой является бесплатная лицензия на использование, а также полнофункциональная и расширяемая интегрированная среда разработки. На момент написания отчета актуальной версией данного ПО является VisualStudio 2017.

## **Microsoft SQL Server**

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка [6].

## **Oracle 12c**

**Oracle Database** - это объектно-реляционная система поддерживающая некоторые технологии, реализующие объектно-ориентированный подход, то есть обеспечивающих управление создания и использования баз данных [2].

**Ключевые возможности Oracle Database**

* Real Application Cluster (RAC) обеспечивает работу одного экземпляра базы данных на нескольких узлах grid, позволяя управлять нагрузкой и гибко масштабировать систему в случае необходимости.
* Automatic Storage Management (ASM) позволяет автоматически распределять данные между имеющимися ресурсами систем хранения данных, что повышает отказоустойчивость системы и снижает общую стоимость владения (TCO).
* Производительность. Oracle Database позволяет автоматически управлять уровнями сервиса и тиражировать эталонные конфигурации в рамках всей сети.
* Простые средства разработки. Новый инструмент разработки приложений HTML DB позволит простым пользователям создавать эффективные приложения для работы с базами данных в короткие сроки.
* Самоуправление. Специальные механизмы Oracle Database позволяют самостоятельно перераспределять нагрузку на систему, оптимизировать и корректировать SQL-запросы, выявлять и прогно- зировать ошибки.
* Большие базы данных. Теперь максимальный размер экземпляра базы данных Oracle может достигать 8 экзабайт.
* Недорогие серверные системы. Oracle Database может использовать недорогие однопроцессорные компьютеры или модульные системы из “серверов-лезвий”.
* В новой версии базы данных реализована поддержка переносимых табличных пространств, система управления потоками данных Oracle Streams и модель распределенных SQL-запросов. Для переноса существующих баз данных в среду Grid в них не потребуется вносить изменений, что позволяет быстро начать использовать все преимущества Oracle Database.

# Введение

В деканате информационных технологий Белорусского государственного технологического университета осуществляется учёт защиты лабораторных работ студентов, анализ текущей успеваемости студентов. Учёт защиты лабораторных работ производится путём получения от преподавателей отчетов по сданным студентами лабораторным работам в бумажном виде.

Дипломный проект посвящён автоматизации процесса документооборота в деканате высшего учебного заведения.

Задачи:

* Написание первой главы дипломного проекта.
* Изучить литературу по теме дипломного проекта.
* Изучить технологические процессы, программные средства, информационные технологии, относящиеся к теме дипломного проекта.

1. **Анализ и выбор технологий и инструментария для разработки автоматизированной системы «Электронный деканат»**
   1. **Аналитический обзор литературы**
      1. **СУБД**

Система управления базами данных (СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных [1].

Основные функции СУБД

* управление данными во внешней памяти (на дисках);
* управление данными в оперативной памяти с использованием дискового кэша;
* журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
* поддержка языков БД (язык определения данных, язык манипулирования данными).

Обычно современная СУБД содержит следующие компоненты:

* ядро, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию,
* процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода,
* подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД
* а также сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.
  + 1. **Веб-сервер**

Понятие "Веб-сервер" может относиться как к железу, так и к программному обеспечению.

С точки зрения железа, `Веб-сервер` это компьютер, который хранит ресурсы сайта (HTML документы, CSS стили, JavaScript файлы и другое) и доставляет их на устройство конечного пользователя (веб-браузер и т.д.).

С точки зрения ПО, Веб-сервер включает в себя некоторые вещи, которые контролируют доступ веб-пользователей к размещенным на сервере файлам, это минимум HTTP сервера. HTTP сервер это часть ПО которая понимает URL’ы (веб-адреса) и HTTP (протокол который использует ваш браузер для просмотра веб-станиц).

Простыми словами, когда браузеру нужен файл размещенный на веб-сервере, браузер запрашивает его через HTTP. Когда запрос достигает нужного веб-сервера (железо), сервер HTTP (ПО) передает запрашиваемый документ обратно, также через HTTP [5].

* 1. **Стек технологий LAMP**

Рассмотрим свободно распространяемый и практически бесплатный набор программного обеспечения под названиям LAMP. Как правильно расшифровывается данная аббревиатура? Она состоит из 4-ех популярных технологий, которые состоят в данной связке: Linux — операционная система; Apache — веб-сервер; MySQL — СУБД; PHP — язык программирования, с помощью которого создают веб-ресурсы.

Данный набор ПО устанавливают на сервер для отображения динамических веб-сайтов и веб-приложений. Эта платформа позволяет установить (развернуть) большинство типов веб-сайтов и сетевого программного обеспечения на вашем сервере.

**Комбинация LAMP – это**:

1. Серверная **ОС Linux** для выполнения нужных задач;
2. веб-сервер **Apache**, для которого создано множество дополнительных модулей, которые решают вопрос совместной работы веб-сервера и сценариев, написанных на самых разных языках программирования;
3. серверный язык динамических сценариев **PHP** (либо Perl);
4. **СУБД** **MySQL**, которая демонстрирует отличную скорость выполнения SQL-запросов и идеально подходит для малых и средних проектов. MySQL работает на Unix и Windows, отличается особенным удобством в работе.

**Для чего может понадобиться LAMP?**

1. в случае необходимости окружения для веб-разработки и тестирования приложений, которые написаны для LAMP. Это как и для собственных приложений, так и для любых необходимых Вам CMS. Вы можете работать на Windows и на Ubuntu;
2. если Вам необходимо сделать на виртуальном VPS сервере или на выделенном сервере производительный веб-сервер для хостинга своих проектов;
3. если Вам нужен сервер для систем контроля версий;
4. для самообучения администрирования;
5. в экономических целях, для создания собственного сервера.

Как весомое преимущество данной сборки: она замечательно подходит для быстрого развертывания приложения, из-за простой конфигурации, но все же она дает мало функций в плане масштабируемости и изоляции компонентов.

Недостаток подобной сборки: приложение и база данных используют одни и те же ресурсы сервера (CPU, память, I/O и т.д.), что дает низкую производительность и затрудняет определение источника (приложение или база данных) этой проблемы. Так же есть помехи в осуществлении горизонтального масштабирования[7].

* 1. **Стек технологий Java**

**1.3.1. JEE**

Сервер приложений (англ. application server) — это программная платформа (фреймворк), предназначенная для эффективного исполнения процедур (программ, скриптов), на которых построены приложения. Сервер приложений действует как набор компонентов, доступных разработчику программного обеспечения через API (Интерфейс прикладного программирования), определённый самой платформой [8].

Для веб-приложений основная задача компонентов сервера — обеспечивать создание динамических страниц. Однако современные серверы приложений включают в себя и поддержку кластеризации, повышенную отказоустойчивость, балансировку нагрузки, позволяя таким образом разработчикам сфокусироваться только на реализации бизнес-логики.

В случае сервера приложений Java, сервер приложений ведёт себя как расширенная виртуальная машина для запуска приложений, прозрачно управляя соединениями с базой данных с одной стороны и соединениями с веб-клиентом с другой.

**Преимущества серверов приложений**

1. Целостность данных и кода

Выделяя бизнес логику на отдельный сервер, или на небольшое количество серверов, можно гарантировать обновления и улучшения приложений для всех пользователей. Отсутствует риск, что старая версия приложения получит доступ к данным или сможет их изменить старым несовместимым образом.

1. Централизованная настройка и управление

Изменения в настройках приложения, таких как изменение сервера базы данных или системных настроек, могут производиться централизованно.

1. Безопасность

Сервер приложений действует как центральная точка, используя которую, поставщики сервисов могут управлять доступом к данным и частям самих приложений, что считается преимуществом защиты. Её наличие позволяет переместить ответственность за аутентификацию с потенциально небезопасного уровня клиента на уровень сервера приложений, при этом дополнительно скрывая уровень базы данных.

1. Поддержка транзакций

Транзакция представляет собой единицу активности, во время которой большое число изменений ресурсов (в одном или различных источниках) может быть выполнено атомарно (как неделимая единица работы). Конечные пользователи при этом могут выиграть от стандартизованного поведения системы, от уменьшения времени на разработку и от снижения стоимости. В то время как сервер приложений выполняет массу нужного генерирования кода, разработчики могут сфокусироваться на бизнес-логике.

Примеры реализации

* + 1. Под сервером приложений в случае Java EE подразумевается комплекс программ, реализующих концепцию Java EE и позволяющих запускать в себе Java EE приложения. К классу серверов приложений относятся такие продукты как Sun GlassFish, IBM WebSphere, RedHat JBoss Application Server, Apple WebObjects (англ.), Oracle Weblogic Server и др.

1. Zope, развитый сервер web-приложений.
2. Терминальные серверы, например поставляемые компанией Citrix

**1.3.2. Spring Framework**

Spring Framework (или коротко Spring) — универсальный фреймворк с открытым исходным кодом для Java-платформы. Также существует форк для платформы .NET Framework, названный Spring.NET [10].

Первая версия была написана Родом Джонсоном, который впервые опубликовал её вместе с изданием своей книги «Expert One-on-One Java EE Design and Development

Фреймворк был впервые выпущен под лицензией Apache 2.0 license в июне 2003 года. Первый стабильный релиз 1.0 был выпущен в марте 2004. Spring 2.0 был выпущен в октябре 2006, Spring 2.5 — в ноябре 2007, Spring 3.0 в декабре 2009, и Spring 3.1 в декабре 2011. Текущая версия — 4.3.7.

Несмотря на то, что Spring не обеспечивал какую-либо конкретную модель программирования, он стал широко распространённым в Java-сообществе главным образом как альтернатива и замена модели Enterprise JavaBeans. Spring предоставляет бо́льшую свободу 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.

Spring обеспечивает решения многих задач, с которыми сталкиваются Java-разработчики и организации, которые хотят создать информационную систему, основанную на платформе Java. Из-за широкой функциональности трудно определить наиболее значимые структурные элементы, из которых он состоит. Spring не всецело связан с платформой Java Enterprise, несмотря на его масштабную интеграцию с ней, что является важной причиной его популярности.

Spring, вероятно, наиболее известен как источник расширений (features), нужных для эффективной разработки сложных бизнес-приложений вне тяжеловесных программных моделей, которые исторически были доминирующими в промышленности. Ещё одно его достоинство в том, что он ввел ранее неиспользуемые функциональные возможности в сегодняшние господствующие методы разработки, даже вне платформы Java.

Этот фреймворк предлагает последовательную модель и делает её применимой к большинству типов приложений, которые уже созданы на основе платформы Java. Считается, что Spring реализует модель разработки, основанную на лучших стандартах индустрии, и делает её доступной во многих областях Java.

**Модули**

Spring может быть рассмотрен как коллекция меньших фреймворков или фреймворков во фреймворке. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании. Эти фреймворки делятся на структурные элементы типовых комплексных приложений:

* Inversion of Control-контейнер: конфигурирование компонентов приложений и управление жизненным циклом Java-объектов.
* Фреймворк аспектно-ориентированного программирования: работает с функциональностью, которая не может быть реализована возможностями объектно-ориентированного программирования на Java без потерь.
* Фреймворк доступа к данным: работает с системами управления реляционными базами данных на Java-платформе, используя JDBC- и ORM-средства и обеспечивая решения задач, которые повторяются в большом числе Java-based environments.
* Фреймворк управления транзакциями: координация различных API управления транзакциями и инструментарий настраиваемого управления транзакциями для объектов Java.
* Фреймворк MVC: каркас, основанный на HTTP и сервлетах, предоставляющий множество возможностей для расширения и настройки (customization).
* Фреймворк удалённого доступа: конфигурируемая передача Java-объектов через сеть в стиле RPC, поддерживающая RMI, CORBA, HTTP-based протоколы, включая web-сервисы (SOAP).
* Фреймворк аутентификации и авторизации: конфигурируемый инструментарий процессов аутентификации и авторизации, поддерживающий много популярных и ставших индустриальными стандартами протоколов, инструментов, практик через дочерний проект Spring Security (ранее известный как Acegi).
* Фреймворк удалённого управления: конфигурируемое представление и управление Java-объектами для локальной или удалённой конфигурации с помощью JMX.
* Фреймворк работы с сообщениями: конфигурируемая регистрация объектов-слушателей сообщений для прозрачной обработки сообщений из очереди сообщений с помощью JMS, улучшенная отправка сообщений по стандарту JMS API.
* Тестирование: каркас, поддерживающий классы для написания модульных и интеграционных тестов.

**Inversion of Control**

Центральной частью Spring является контейнер Inversion of Control, который предоставляет средства конфигурирования и управления объектами Java с помощью рефлексии. Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путём связывания их между собой.

Объекты, создаваемые контейнером, также называются управляемыми объектами (beans). Обычно конфигурирование контейнера осуществляется путём загрузки XML-файлов, содержащих определение bean’ов и предоставляющих информацию, необходимую для создания bean’ов.

Объекты могут быть получены одним из двух способов:

* Поиск зависимости — шаблон проектирования, в котором вызывающий объект запрашивает у объекта-контейнера экземпляр объекта с определённым именем или определённого типа.
* Внедрение зависимости — шаблон проектирования, в котором контейнер передает экземпляры объектов по их имени другим объектам с помощью конструктора, свойства или фабричного метода.

**MVC**

Spring имеет собственную MVC-платформу веб-приложений, которая не была первоначально запланирована. Разработчики Spring решили написать её как реакцию на то, что они восприняли как неудачность конструкции (тогда) популярного Apache Struts, а также других доступных веб-фреймворков. В частности, по их мнению, было недостаточным разделение между слоями представления и обработки запросов, а также между слоем обработки запросов и моделью.[4]

Класс DispatcherServlet является основным контроллером фрэймворка и отвечает за делегирование управления различным интерфейсам, на всех этапах выполнения HTTP-запроса. Об этих интерфейсах следует сказать более подробно.

Как и Struts, Spring MVC является фреймворком, ориентированным на запросы. В нем определены стратегические интерфейсы для всех функций современной запросно-ориентированной системы. Цель каждого интерфейса — быть простым и ясным, чтобы пользователям было легко его заново имплементировать, если они того пожелают. MVC прокладывает путь к более чистому front-end-коду. Все интерфейсы тесно связаны с Servlet API. Эта связь рассматривается некоторыми как неспособность разработчиков Spring предложить для веб-приложений абстракцию более высокого уровня. Однако эта связь оставляет особенности Servlet API доступными для разработчиков, облегчая все же работу с ним. Наиболее важные интерфейсы, определенные Spring MVC, перечислены ниже:

* HandlerMapping: выбор класса и его метода, которые должны обработать данный входящий запрос на основе любого внутреннего или внешнего для этого запроса атрибута или состояния.
* HandlerAdapter: вызов и выполнение выбранного метода обработки входящего запроса.
* Controller: включен между Моделью (Model) и Представлением (View). Управляет процессом преобразования входящих запросов в адекватные ответы. Действует как ворота, направляющие всю поступающую информацию. Переключает поток информации из модели в представление и обратно.
* View: ответственно за возвращение ответа клиенту в виде текстов и изображений. Некоторые запросы могут идти прямо во View, не заходя в Model; другие проходят через все три слоя.
* ViewResolver: выбор, какое именно View должно быть показано клиенту.
* HandlerInterceptor: перехват входящих запросов. Сопоставим, но не эквивалентен сервлет-фильтрам (использование не является обязательным и не контролируется DispatcherServlet-ом).
* LocaleResolver: получение и, возможно, сохранение локальных настроек (язык, страна, часовой пояс) пользователя.
* MultipartResolver: обеспечивает Upload — загрузку на сервер локальных файлов клиента.

Spring MVC предоставляет разработчику следующие возможности:

* Ясное и прозрачное разделение между слоями в MVC и запросах.
* Стратегия интерфейсов — каждый интерфейс делает только свою часть работы.
* Интерфейс всегда может быть заменен альтернативной реализацией.
* Интерфейсы тесно связаны с Servlet API.
* Высокий уровень абстракции для веб-приложений.
* В веб-приложениях можно использовать различные части Spring, а не только Spring MVC.

**Доступ к данным**

Spring предоставляет свой слой доступа к базам данных и поддерживает все популярные СУБД.

JDBC, iBatis / MyBatis, Hibernate, JDO, JPA, Oracle TopLink, Apache OJB, Apache Cayenne и т. д.

Для всех этих фреймворков Spring предоставляет такие особенности:

* Управление ресурсами — автоматическое получение и освобождение ресурсов базы данных
* Обработка исключений — перевод исключений при доступе к данным в исключения Spring-а
* Транзакционность — прозрачные транзакции в операциях с данными
* Распаковка ресурсов — получение объектов базы данных из пула соединений
* Абстракция для обработки BLOB и CLOB

**Управление транзакциями**

Фреймворк управления транзакциями в Spring привносит механизм абстракций для платформы Java. Основные возможности этих абстракций:

* работа с локальными и глобальными транзакциями
* работа с вложенными транзакциями
* работа с точками сохранения в транзакциях

**«Соглашение прежде конфигурации» в разработке приложения**

Spring Roo — фреймворк, поддерживающий принцип «соглашение прежде конфигурации» для быстрого создания бизнес-приложений.

Может использоваться со:

* Spring Framework
* Spring Security
* Spring Web Flow

**Интеграция**

Spring Integration — фреймворк для JEE, который предоставляет функции, необходимые для отправки сообщений или для построения событийно-ориентированной архитектуры.

* роутеры
* трансформеры
* адаптеры для интеграции с другими технологиями и системами (HTTP, AMQP, JMS, XMPP, SMTP, IMAP, FTP (FTPS/SFTP), файловые системы и т. д.)
* фильтры
* активаторы сервисов
* аудит и управление
  1. **Стек технологий MEAN**

В последнее время в веб-сообществе много говорят о новом стеке MEAN: MongoDB, Express, AngularJS, Node.js. Стек технологий MEAN отражает современный подход к веб-разработке: когда на каждом уровне приложения, от клиента до сервера и персистентности, применяется один и тот же язык (JavaScript).

MEAN — это больше, чем простая перестановка начальных букв и новых технологий. Смещение базовой платформы с ОС (Linux) к среде исполнения JavaScript (Node.js) несет с собой независимость от ОС: Node.js работает на Windows® и OS X так же, как и на Linux.

Node.js заменяет Apache из стека LAMP. Но Node.js — это гораздо больше, чем просто веб-сервер. На самом деле готовое приложение не развертывается на отдельном веб-сервере; вместо этого сам веб-сервер включается в приложение и автоматически устанавливается в составе стека MEAN. В результате процесс развертывания значительно упрощается, так как требуемая версия веб-сервера явно определена вместе с остальными зависимостями времени выполнения.

Переход от традиционной базы данных, такой как MySQL, к бессхемному, документо-ориентированному NoSQL-хранилищу, такому как MongoDB, представляет собой фундаментальный сдвиг в стратегии персистенции. Программист тратит меньше времени на написание операторов SQL и больше — на написание функций map/reduce на JavaScript. При этом исключаются огромные пласты логики преобразования, так как MongoDB изначально выдает формат JavaScript Object Notation (JSON). В результате предельно упрощается написание веб-сервисов REST.

Но главный сдвиг между LAMP и MEAN заключается в переходе от традиционного генерирования страниц на стороне сервера к ориентации на одностраничные приложения (SPA) на стороне клиента. Express позволяет управлять и маршрутизацией / генерацией страниц на стороне сервера, но теперь — благодаря AngularJS — упор делается на представления на стороне клиента. Эта перемена означает не просто перенос ваших артефактов модель-представление-контроллер (MVC) с сервера в клиентское устройство. Это также скачок от менталитета синхронности к менталитету, носящему событийно-управляемый, принципиально асинхронный характер. И, пожалуй, самое главное, — это движение от странично-ориентированных приложений к компонентно-ориентированным.

Стек MEAN не «заточен» на мобильные приложения — AngularJS одинаково хорошо работает на настольных компьютерах и ноутбуках, смартфонах и планшетах и даже на смарт-телевизорах, — но он и не относится к мобильным устройствам как к гражданам второго сорта. И тестирование больше не откладывается на потом: с помощью платформ тестирования мирового класса, таких как MochaJS, JasmineJS и KarmaJS, можно писать тщательные и всеобъемлющие наборы тестов для своих MEAN-приложений [9].

* 1. **Стек технологий .NET**

.NET Framework — это про­грамм­ная плат­фор­ма, выпу­щен­ная ком­па­нией Microsoft, кото­рая под­хо­дит для раз­ных язы­ков про­грам­ми­ро­ва­ния. ASP.NET — тех­но­ло­гия созда­ния веб-при­ло­же­ний и веб-сер­ви­сов от ком­па­нии Microsoft. Она явля­ется состав­ной частью плат­формы Microsoft. NET и раз­ви­тием более ста­рой тех­но­ло­гии Microsoft ASP.

Счи­та­ет­ся, что плат­форма. NET Framework яви­лась отве­том ком­па­нии Microsoft на набрав­шую к тому вре­мени боль­шую попу­ляр­ность плат­форму Java. ASP.NET осно­вы­ва­ется на Common Language Runtime: раз­ра­бот­чики могут писать код для ASP.NET, исполь­зуя прак­ти­че­ски любые языки программирования, неко­то­рые из кото­рых вхо­дят в ком­плект. NET Framework (C#, Visual Basic.NET и JScript. NET), а дру­гие могут быть уста­нов­лены допол­ни­тельно (IronRuby, IronPython, PHP, Perl, Smalltalk, Haskell и др.). Тем не менее, нередко исполь­зу­ется не вполне кор­рект­ный тер­мин «.NET-про­грам­мист», кото­рый, види­мо, харак­те­ри­зует раз­ра­бот­чика на любом языке про­грам­ми­ро­ва­ния, кото­рый может быть исполь­зо­ван в. NET-про­ек­те.

Неко­то­рые осо­бен­но­сти ASP.NET:

* Ком­пи­ли­ру­е­мый код выпол­ня­ется быст­рее, а боль­шин­ство оши­бок отлав­ли­ва­ется ещё на ста­дии раз­ра­бот­ки.
* Рас­ши­ря­е­мый набор эле­мен­тов управ­ле­ния и биб­лио­тек клас­сов, уско­ря­ю­щий раз­ра­бот­ку.
* Воз­мож­ность кэши­ро­ва­ния всей стра­ни­цы, её частей или дан­ных, исполь­зу­е­мых на стра­ни­це.
* Воз­мож­ность раз­де­ле­ния визу­аль­ной части и биз­нес-логики по раз­ным фай­лам, есть воз­мож­ность выде­лять часто исполь­зу­е­мые шаб­лоны поль­зо­ва­тель­ских эле­мен­тов управ­ле­ния, таких как меню сай­та, нали­чие master-стра­ниц для зада­ния шаб­ло­нов оформ­ле­ния, поддержка AJAX (расширение ASP.NET AJAX).
* Рас­ши­ря­е­мые модели собы­тий, обра­ботки запро­сов и сер­вер­ных эле­мен­тов управ­ле­ния.
* Под­держка CRUD-опе­ра­ций при работе с таб­ли­цами через GridView.
* Воз­можно созда­ние веб-при­ло­же­ний, кото­рые реа­ли­зуют шаб­лон Model-View-Controller (ASP.NET MVC Framework).

.NET доста­точно широко рас­про­странён в сфере раз­ра­ботки внут­ри­кор­по­ра­тив­ных про­грамм­ных про­дук­тов, но в веб-раз­ра­ботке всё же встре­ча­ется отно­си­тельно ред­ко, как и дру­гие про­грамм­ные про­дукты кор­по­ра­ции Microsoft. Поэтому найти раз­ра­бот­чи­ков для веб-про­екта бывает доста­точно непро­сто. Исполь­зо­ва­ние. NET «тянет» за собой покупку и иного ПО от кор­по­ра­ции Microsoft (сер­вер­ной ОС, СУБД и т.п.). Тех­но­ло­гия доста­точно доро­гая в раз­ра­ботке и сопро­вож­де­нии: кроме затрат на покупку лицен­зий на необ­хо­ди­мое ПО суще­ствен­ный вклад в бюд­жет про­ек­тов вно­сят высо­кие зар­платы раз­ра­бот­чи­ков [11].

# Патентный поиск и аналоги программного обеспечения

Изобретение – это основной объект промышленной собственности.

Патент на изобретение – это документ, выдаваемый компетентным государственным органом и удостоверяющий: приоритет изобретения, авторство и исключительное право на изобретение. Действует в пределах территории того государства, ведомство которого его выдало.

Под патентоспособностью понимают юридическое свойство объекта промышленной собственности, определяющее его способность охраняться документом исключительного права (патентом) на территории конкретной страны в течение срока действия патента.

Решение является патентоспособным если:

* это способ или вещество;
* обладает новизной;
* имеет изобретательский уровень;
* является промышленно применяемым.

Патентный поиск – это процесс отбора соответствующих запросу документов или сведений по одному или нескольким признакам из массива патентных документов или данных, при этом осуществляется процесс поиска из множества документов и текстов только тех, которые соответствуют теме или предмету запроса.

Существует 3 вида патентного поиска:

* тематический поиск. В данном виде поиска объектом поиска выступает вид изобретения;
* именной поиск. Этот вид поиска производится по названию фирмы-патентообладателя, по фамилии заявителя либо по фамилии автора изобретения;
* нумерационный поиск. Поиск производится по номеру документа либо по нумерационным указателям.

В соответствии с темой дипломного проекта был проведен патентный поиск в области программных систем для расчета дорожных одежд нежесткого типа.

В качестве аналогов можно выделить: 1) программа для учета ставок, размещенных в букмекерской конторе, её пунктах приема ставок; 2) программа для учета ставок, размещенных в букмекерской конторе, её пунктах приема ставок - модуль расчета НДФЛ; 3)управление функционированием процессингового центра автономной букмекерской конторы системы; 4) управление функционированием процессингового центра автономной букмекерской конторы системы;

3.1 Программное обеспечение мониторинга заданий резервного копирования (Simple Backup Monitor IT)

Язык программирования**:**Java.

Номер регистрации (свидетельства): 2013616479.

Дата регистрации: 09.07.2013.

Реферат:

Программа предназначена для: предоставления средств оперативного визуального контроля процессов создания резервных копий ОС, БД и данных приложений серверов на основе информации о событиях резервного копирования , регистрируемых во встроенной БД IBM TivoliStorageManager; сохранения и просмотра информации о ходе выполнения операций резервного копирования за заданный период (ретроспективный контроль). Входные данные: информация о событиях резервного копирования в формате actlog; параметры настройки программного обеспечения. Выходные данные: отчет о результатах выполнения резервного копирования всех серверов; детальная информация о выполнении каждого задания резервного копирования .

# 3.2 Программа для разработки и администрирования баз данных InterBase/Firebird "EMS SQL ManagerforInterBase/Firebird

Язык программирования:Delphi.

Номер регистрации (свидетельства): 2017613511.

Дата регистрации: 21.03.2017.

Реферат: Программа работает с последними версиями СУБД для IB и FB и поддерживает все новейшие функциональные возможности этих серверов, включая триггеры БД ,журналирование, инкрементальное резервное копирование и другие. Программа включает в себя такие инструменты как: визуальный конструктор баз данных, который позволяет быстро создавать базы данных InterBase/Firebird; визуальный конструктор запросов и отладчик хранимых процедур для разработки SQL сценариев InterBase/Firebird, а также другие инструменты для администрирования серверов InterBase и Firebird.

3.3 Программное средство «Система управления базами данных Синтез»

Язык программирования:С.

Номер регистрации (свидетельства): 2017613104.

Дата регистрации: 10.03.2017.

Реферат: Программа предназначена для обеспечения функционирования баз данных автоматизированных систем. Область применения: автоматизированные системы ЦОД (центр обработки данных). Функциональные возможности: создание новых баз данных; создание объектов баз данных, в т.ч. таблиц, представлений данных (views), функций; поддержка резервного копирования и восстановления БД без остановки сервера; дискреционное разграничение доступа к защищаемым объектам (таблицы, представления, схемы, базы данных) внутренними средствами СУБД; разграничение доступа при помощи политик SELinux, установленных на сервере СУБД. Для использования программы требуется программный комплекс «Клиентская операционная система Синтез».

# Заключение

В ходе написания первой главы был проведен соответствующий аналитический обзор литературы, в котором были описаны цели и задачи дипломного проекта. Задача состоит в том, чтобы разработать программный продукт, автоматизирующий процесс документооборота в деканате высшего учебного заведения. Также был проведен патентный поиск аналогов программного обеспечения для автоматизации документооборота. Каждый аналог был подробно описан, и было показано, для чего предназначено то или иное программное обеспечение.

Список используемых источников

1. Кафедра Интеллектуальных Информационных Технологий ИнФО УрФУ [Электронный ресурс] – Режим доступа: <http://lecturesdb.readthedocs.io/databases/dbms.html>. Дата доступа: 15.04.2017. 2. Частное Боровское исследовательское учреждение по внедрению новых технологий [Электронный ресурс] – Режим доступа: <http://bourabai.ru/dbt/servers/oracle2.htm>. Дата доступа: 15.04.2017.

3. Абитуриент БГТУ [Электронный ресурс] – Режим доступа: <https://abiturient.belstu.by/foreign-abiturient.html>. Дата доступа: 15.04.2017.

4. Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>. Дата доступа: 15.04.2017.

5. Mozilla Developer Network [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/Что_такое_веб_сервер>. Дата доступа: 15.04.2017.

6. Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Microsoft_SQL_Server>. Дата доступа: 15.04.2017.

7. HyperHost хостинг [Электронный ресурс] – Режим доступа: <https://hyperhost.ua/info/?p=3007>. Дата доступа: 15.04.2017.

8. Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Сервер_приложений>. Дата доступа: 15.04.2017.

9. developerWorks [Электронный ресурс] – Режим доступа: <https://www.ibm.com/developerworks/ru/library/wa-mean1/>. Дата доступа: 15.04.2017.

10. Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Spring_Framework>. Дата доступа: 15.04.2017.

11. Web Creator [Электронный ресурс] – Режим доступа: <https://web-creator.ru/articles/dot_net_and_asp>. Дата доступа: 15.04.2017.

2. Логистическая компания в Минске[Электронный ресурс] – Режим доступа: https://http://plogistics.by. Дата доступа: 11.04.2017.

3. Федеральный Институт Промышленной собственности [Электронный ресурс] – Режим доступа: http://www1.fips.ru/wps/wcm/connect/content\_ru/. Дата доступа:11.04.2017.