

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН  
ИННОВАЦИОННЫЙ ЕВРАЗИЙСКИЙ УНИВЕРСИТЕТ

МАГИСТРАТУРА

Кафедра «Автоматизированные системы обработки информации и  
управления»

Магистерская диссертация

**ИССЛЕДОВАНИЕ ДОКУМЕНТООБОРОТА ДЕКАНАТА ДЛЯ  
РАЗРАБОТКИ КОМПЛЕКСНОЙ АВТОМАТИЗАЦИИ ИИЕУ  
6N0704 «Вычислительная техника и программное обеспечение»**

Исполнитель \_\_\_\_\_ А.В. Чебатко  
(подпись, дата)

Научный руководитель  
Проф., к.т.н. \_\_\_\_\_ Р.А. Шагиева  
(подпись, дата)

Нормоконтроллер \_\_\_\_\_ Т.М. Салий  
(подпись, дата)

Допущена к защите:

Зав.кафедрой «Автоматизированные системы обработки информации и  
управления»  
Профессор, к.т.н. \_\_\_\_\_ В.В.Наумов  
(подпись, дата)

Павлодар, 2010

**РЕФЕРАТ**

## РЕФЕРАТ

Данная диссертационная работа посвящена анализу работы деканатов и разработке автоматизированной системы, предназначенной для создания, хранения и обработки информации о студентах деканатов. Работа представлена на 68 страницах с использованием 9 рисунков.

**Перечень ключевые слов:** электронный документооборот, система, веб-приложение, сервер, клиент, каркас приложения, модель-представление-контроллер, документ, приказ.

**Актуальность** данной работы обусловлена, важностью повышения эффективности работы деканатов в структуре университета.

**Новизна** исследования заключается в применении каркаса приложений Yii и архитектуры Модель-Представление-Контроллер для создания системы автоматизации деканата.

**Цель** исследования - выработка и методов создания, хранения и обработки информации о студентах деканатов ИнЕУ, а так же методов подготовки и обработки документации деканатов.

В качестве **объекта** исследования выступает организация эффективной работы с данными студентов и быстрой подготовки документации.

**Предметом** является система автоматизации обработки и хранения информации о студентах и создания документации для деканатов.

**Полученные результаты.** Разработана система которая позволяет повысить эффективность работы секретарей деканатов путем автоматизации работы с документами и ускорения доступа к информации о студентах.

## ABSTRACT

This dissertation analyzes the work of deans' offices and the development of an automated system for creating, storing and processing information about the student dean. The work presented in 68 pages with 9 figures.

**List of key words:** electronic document management system, web-based application server, client, application framework, the model-view-controller, the document order.

**The relevance** of this work is due, the importance of enhancing the effectiveness of the deans' offices in the university structure.

**The novelty** of the research is to use a skeleton application Yii and architecture Model-View-Controller to establish a system of automation of the deanery.

**Objective** - To elaborate and techniques for creating, storing and processing information about the student dean's InEU, as well as methods for preparing and processing documentation dean.

**As the object** of research is the organization work effectively with these students and rapid preparation of documentation.

**The subject** is the automation system for processing and storing information about students and the creation of documentation for the dean's office.

**Results** A system that enhances the efficiency of the secretaries deans' offices by automating the work with documents and accelerate access to information about students.

## ОПРЕДЕЛЕНИЯ

В настоящей диссертации применяются следующие термины с соответствующими определениями:

**Документ** – материальный носитель, содержащий, выражающий информацию в зафиксированном виде и специально предназначенный для её передачи во времени и пространстве.

**Электронный документооборот** – единый механизм по работе с документами, представленными в электронном виде, с реализацией концепции «безбумажного делопроизводства».

**Атрибут** – необходимое, существенное, неотъемлемое свойство объекта.

**Объектно-ориентированное программирование** – парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

**Каркас приложения** – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. В его состав могут входить вспомогательные программы, библиотеки кода и прочее.

**Веб-браузер** – программное обеспечение для просмотра веб-сайтов, то есть для запроса веб-страниц (преимущественно из Сети), их обработки, вывода и перехода от одной страницы к другой.

**Веб-сервер** – это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей диссертации применяются следующие обозначения и сокращения:

ИнЕУ – Инновационный Евразийский Университет.

ИТ – информационные технологии.

СУД – система управления документами.

СУЭД – система управления электронными документами.

EDMS – electronic document management system, см. СУЭД.

OCR – optical character recognition (оптическое распознавание символов) – автоматическое распознавание с помощью специальных программ графических изображений символов печатного текста (например, введённого в компьютер с помощью сканера) и преобразование их в формат, пригодный для обработки текстовыми процессорами, редакторами текстов и т. д.

ICR – intelligent character recognition (интеллектуальное распознавание символов) – технология, обеспечивающая распознавание изображений текстов из символов, вписываемых в специальные формы.

БД – база данных.

СУБД – система управления базами данных.

ПО – программное обеспечение.

PDM – product data management – управление производственной документацией.

КИС – корпоративная информационная система.

MVC – модель-представление-контроллер. Архитектура программного обеспечения, в которой модель данных, бизнес-логика и пользовательский интерфейс разделены на три отдельных компонента.

## СОДЕРЖАНИЕ

<b>Введение</b>	8
<b>1 Обзор и анализ литературы по проблеме электронного документооборота</b>	11
1.1 Понятие системы электронного документооборота	11
1.2 Виды систем электронного документооборота	13
1.3 Функции и задачи систем управления документами	23
<b>2 Исследование работы деканата</b>	30
<b>3 Разработка методов создания, хранения и обработки информации о студентах и методов создания документации</b>	33
3.1 Выбор средств автоматизации	39
3.2 Инструменты разработки	40
3.3 Описание системы	60
<b>Заключение</b>	67
<b>Список использованных источников</b>	68

## ВВЕДЕНИЕ

В настоящее время многие предприятия уделяют все больше внимания управлению производственными процессами и повышению эффективности процессов производства и обслуживания. Такая же тенденция наблюдается и в высших учебных заведениях Республики Казахстан. Эффективное решение проблемы документооборота является одной из приоритетных целей высших учебных заведений.

Высшее учебное заведение является организацией, которая осуществляет образовательный процесс, научную и инновационную деятельность, управляющие процессы и процессы обеспечения (бухгалтерия, кадровый учет, делопроизводство). В качестве «продукции» университета можно выделить:

- услуги по реализации закона об образовании;
- услуги по подготовке специалистов;
- выпуск учебно-методической продукции (учебные курсы, книги);
- выпуск научно-технической продукции (реализация инновационных проектов, проведение НИОКР и т.п.).

На любом из предприятий главным способом представления информации являются документы. В современных рыночных условиях от предприятий требуется способность гибко реагировать на внешние изменения и эффективность движения документов внутри предприятия является одним из критичных факторов, влияющих на общую эффективность предприятия. Малоэффективный документооборот означает несвоевременное исполнение управляющих функций, задержку в доставке важной информации, упущение бизнес возможностей. Результатом может явиться упущенная прибыль, снижение эффективности и даже крах всего бизнеса. Особенно остро проблема эффективного документооборота стоит на крупных предприятиях. Ввиду всего перечисленного, на любом предприятии, активно работающем с документами, со временем требуется введение системы электронного документооборота.

Для ИНЕУ, как и для любого крупного учебного учреждения, уже давно остро встал вопрос автоматизации хранения, поиска и обработки больших массивов информации, а также обеспечения безопасности её хранения, передачи и возможности совместного использования, что особенно актуально, учитывая территориально – распределённую структуру университета со своими корпусами.

На основе результатов исследования работы деканатов выявлен перечень недостатков:

- Низкая оперативность получения и передачи информации. Неудобный и долгий поиск необходимой информации;
- Некачественное исполнение функциональных обязанностей из-за перегрузки процедуральной работой.
- Дублирование отчетной и командой информации. Независимый сбор и обработка аналогичной информации разными отделами;
- Отсутствие автоматизированного сбора и обработки информации.

На основании вышеизложенного можно сделать вывод о необходимости снижения временных затрат на получение и обработку информации деканатов Инновационного Евразийского Университета.

**Целью** данного исследования является выработка и методов создания, хранения и обработки информации о студентах деканатов ИнЕУ, а так же методов подготовки и обработки документации деканатов.

Для достижения поставленной цели необходимо решить следующие **основные задачи**:

- изучить существующие разработки по предметной области исследования;
- изучить методику предпроектного обследования предприятия;
- разработать альтернативный метод создания, хранения и обработки информации о студентах и документации.

**Направление научного исследования** – исследование деятельности деканатов в процессе их работы. Рассмотрение документации, создаваемой и используемой в процессе работы деканатов. На основе полученных данных создается метод обработки информации о студентах и документации деканата.

**Теоретическая значимость.** Положения, разработанные в диссертации, могут служить основой для дальнейших исследований по проблеме автоматизации документооборота Инновационного Евразийского Университета.

**Практическая значимость** исследования заключается в возможности применения результатов исследования в вузах Республики Казахстан для автоматизации деятельности подразделений с похожими задачами.

**Новизна** исследования состоит в разработке методов создания, хранения и обработки информации о студентах и обработку документации с разработкой информационной системы на основе веб-технологий с использованием каркаса разработки Yii.

**Ожидаемые результаты.** Применение разработанных методов создания, обработки и хранения информации о студентах позволит достигнуть следующих результатов:

- облегчить работу секретарей деканатов и улучшить эффективность их работы;
- обеспечить быстрый поиск необходимой информации;
- облегчить создание необходимой документации
- исключить дублирование информации
- ввести единую информационную базу

# 1 ОБЗОР И АНАЛИЗ ЛИТЕРАТУРЫ ПО ПРОБЛЕМЕ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

## 1.1 Понятие системы электронного документооборота

В основе успешного бизнеса любой фирмы лежит правильность и своевременность принятия решений руководством фирмы, гибкое реагирование на изменения ситуации на рынке или на предприятии. Постоянный рост объемов информации, необходимой для принятия управленческих решений, приводит к тому, что приходится получать, обрабатывать и хранить документы в большем количестве, чем раньше. Традиционные методы работы с документами становятся при этом малоэффективными.

Следует отметить, что потоки корпоративной бизнес-информации чрезвычайно разнообразны по источникам и формам ее представления. Однако их можно условно классифицировать по форме хранения: на электронные документы и бумажные. Существует мнение, что в настоящее время только около 30% всей корпоративной информации хранится в электронном виде (как в структурированном в базах данных, так и в неструктурированном). Вся остальная информация (около 70%) хранится на бумаге, создавая немалые трудности при ее поиске.

Кроме того, в современных непростых экономических условиях может теряться гибкость в реагировании на изменения рынка, вследствие чего дальнейший рост компании останавливается. Правильно организованный документооборот позволяет избежать таких «болезней роста». Решению задачи оптимизации документооборота активно способствует динамичное развитие современных компьютерных и сетевых технологий. По экспертным оценкам применение электронного документооборота способствует росту производительности труда сотрудников на 25-50%, а время обработки одного документа сокращается более чем на 75%. Поэтому неocenимую роль в деятельности любого предприятия играет эффективная система управления электронным документооборотом (СЭД), обеспечивающая бесперебойную циркуляцию его бизнес-потоков.

Концепция безбумажной технологии имеет за плечами не одно десятилетие. В это же время сегодня можно говорить лишь о частичной реализации этой идеи. Здесь ряд причин:

- Можно указать на психологический аспект проблемы. Получение информации в электронном виде требует привычки;
- Безбумажные технологии требуют основательной технической поддержки: соответствующей производительной вычислительной

техники, высокой пропускной способности коммуникационных линий, технологичности алгоритмов;

- Безбумажные технологии требуют основательной юридической поддержки.

Согласно определению компании IDC системы "Системы электронного управления документами (ЭУД) обеспечивают процесс создания, управления доступом и распространения больших объемов документов в компьютерных сетях, а также обеспечивают контроль над потоками документов в организации. Часто эти документы хранятся в специальных хранилищах или в иерархии файловой системы. Типы файлов, которые, как правило, поддерживают системы ЭУД включают текстовые документы, образы, электронные таблицы, аудио-, видео- данные, и документы Web. Общими возможностями систем ЭУД являются создание документов, управление доступом, преобразование и безопасность"

В настоящее время на большинстве предприятий корпоративная информация (договоры, прайс-листы, приказы, деловая переписка, финансовая документация и др.) хранится как в бумажном, так и в электронном виде (часто неупорядоченно на файл-сервере). При этом практически неизбежно возникает дублирование информации, а также не обеспечивается поддержка разных версий документов и безопасность их хранения. То есть, и при преобладании электронной формы хранения документов, они не всегда используются эффективно. Потребность в эффективном управлении электронными документами и привела к созданию систем электронного документооборота (первые из которых появились еще в 1980-е гг.). Эти системы еще называют Electronic Document Management Systems (EDMS).

EDMS-системы являются неотъемлемой частью мирового рынка ПО управления электронным документооборотом и содержимым информационных систем, так называемого DCT (document and content technologies) рынка. Аналитики IDC выделяют следующие сегменты DCT-рынка: системы электронного документооборота; системы управления содержимым на порталах предприятий и системы управления содержимым для электронной коммерции. Рост потребности предприятий в повышении эффективности групповой работы с корпоративными документами неизбежно ведет к дальнейшему развитию данного рынка. Развитию рынка способствуют также распространение электронной коммерции и рост потребности предприятий в Web-совместимых интегрированных инструментальных средствах доступа к информации. При этом особенно быстро растет спрос потребителей на инструментальные средства сбора, поиска и анализа информации, с помощью которых возможно

более оперативно обрабатывать разнородные собрания текстовых файлов, графических файлов, видео- и аудиофайлов.

Тем не менее, по оценкам аналитиков, важность внедрения эффективных систем электронного документооборота для успешного ведения бизнеса сохранилась, и в обозримом будущем их значимость будет только усиливаться.

Под управлением электронным документооборотом в общем случае принято понимать организацию движения документов между подразделениями предприятия, группами пользователей или пользователями. При этом под движением документов понимается не их физическое перемещение (т.к. они чаще всего остаются на сервере), а передачу прав на их использование с уведомлением конкретных пользователей и контролем за их исполнением.

Главное назначение систем электронного документооборота это организация хранения электронных документов, а также работы с ними (в частности, их поиска как по атрибутам, так и по содержимому). В системах электронного документооборота также реализован санкционированный доступ к документам, отслеживаются произведенные в них изменения и контролируются все их версии и подверсии.

## **1.2 Виды систем электронного документооборота**

С целью выяснения сущностных особенностей опишем возможные подходы к классификации систем электронного документооборота [1]. С точки зрения технологий работы с документами прежде всего выделяются т.н. электронные архивы (ЭА). Среди них, в свою очередь, можно выделить системы с развитыми средствами полнотекстового поиска, которые обеспечивают нечеткий поиск (по размытым критериям), смысловой поиск и т. д., то есть, поисково-ориентированные системы. И второй подкласс ЭА - системы, ориентированные на эффективную организацию хранения электронных документов, реализующие т.н. HSM (High Storage Management) подход. Эти системы обеспечивают поддержку широкого диапазона оборудования для хранения информации и т. п. устройств. К такого рода системам относится, например, система IC Архив.

Свои специфические особенности имеют т.н. системы workflow (WF), которые рассчитаны на обеспечение движения работ (объектов) по заранее задаваемым маршрутам (так называемая "жесткая маршрутизация"). На каждом этапе объект может меняться (например, на документе появляется резолюция начальника), поэтому его называют общим словом "работа" (work). Эти системы называют системами workflow - "поток работ". К работам могут быть привязаны документы, но не документы в принципе являются базовым

объектом этих систем. С помощью таких систем можно организовать определенные работы, для которых заранее известны и могут быть формализованы и прописаны все этапы (например, алгоритм приёма сотрудника на работу и прохождения им вступительного адаптационного курса и испытательного срока в компании).

В отдельный класс можно выделить системы, ориентированные на поддержку совместной работы сотрудников (collaboration). Это новое веяние, связанное с пониманием изменчивости рыночных условий в современном мире и с необходимостью иметь для быстрого движения "только самое нужное", без лишнего, очень полезного, но тяжелого балласта. Такие системы, ориентированные на т.н. западный стиль организации работы с документами, нацелены на использование горизонтальных связей и не включают понятия иерархии в организации, не заботятся о какой-либо формализации потока работ. Данные системы создаются с целью обеспечения совместной работы сотрудников, находящихся в различных офисах компании, и сохранения создаваемых ими документов. Популярной стала реализация данных систем в концепции т.н. "порталов" - образовательного портала фирмы и пр. Подобного рода системы предоставляют также сервисы хранения и публикации документов во внутренней информационной среде компании, поиска информации, обсуждения, средства назначения встреч и проведения служебных совещаний (как реальных, так и виртуальных). В практике организации электронного делопроизводства встречаются, как правило, две конкурирующие платформы - Lotes Notes и Microsoft Exchange. Возможно также создание систем, в какой-то степени реализующих все три вышеописанных подхода - т.н. комбинированных систем.

На основе систем электронного документооборота и с их использованием строятся системы управления бизнес-процессами. Кратко перечислим основные их типы. Исторически первыми возникли т.н. системы MRP - Material Requirements Planning - системы учёта поставки материалов и комплектующих в производственный процесс. После дополнения денежной (стоимостной) составляющей данные системы трансформировались в системы более общего уровня - т.н. ERP - Enterprise Requirements Planning - системы планирования коммерческой деятельности предприятия. На основе данного класса систем строится т.н. управленческий учёт, то есть рассчитываются реальные показатели деятельности фирмы - операционный результат, прибыль и т.п. К специфическому классу систем, решающих т.н. транспортные задачи относятся SCM - Supply Shain Management - или системы управления логистикой поставок.

При организации обслуживания клиентов в последнее время всё чаще используются CRM - Custom Relationship Management - системы управления отношениями с клиентами. Оптимизировать и стимулировать работу сотрудников и подразделений компании призваны системы BSC - Balanced Scorecard - системы сбалансированных показателей.

Каждая из указанных систем использует весьма специфические виды информации - к примеру, в CRM системах документируется, архивируется и хранится вся история взаимоотношений с клиентом, а также вся возможная информация о нём, включая личные данные на него и на родственников. В зависимости от задач система в состоянии генерировать в бумажной форме массивы писем, осуществляя т.н. direct-mail рассылку с предложениями новых услуг, товаров, поздравлениями с личными праздниками и т.д.

Данные системы в чистом виде представляют из себя скорее своеобразный конструктор, содержащий инструментарий для создания документов, определения маршрутов их движения, операций над ними и автоматизированных рабочих мест отдельных сотрудников в соответствии со спецификой решаемых ими задач. Для специалиста, занимающегося внедрением такого рода систем, очень важно знать их возможности и особенности и уметь формализовать специфику документооборота в конкретной организации, т.е. грамотно поставить задачи разработчику.

Таким образом, за последние несколько лет инструментарий специалиста в области документационного обеспечения существенно расширился и усложнился и далеко не ограничивается пакетом Microsoft Office. Современные системы электронного документооборота не только полностью выполняют его, документооборота, традиционные функции, существенно сберегая время сотрудников и ресурсы организации, но и реализуют новые, ранее недоступные возможности. Например, контекстный поиск в генеральной совокупности документов, либо отслеживание версий документа с сохранением всех их в одном источнике. Наряду с традиционными определениями документами, вводятся в оборот новые понятия - например, поток работ.

При возможной передаче на архивное хранение встаёт проблема обеспечения хранения не только собственно документов, но и описания маршрутов их движения, специфических атрибутов (например, ЭЦП). Необходимо сохранение конфигурации систем, возможно даже версий программного обеспечения, на котором данные системы создавались. И, в целом, овладение данным инструментарием - задача непростая, однако она позволяет существенно повысить рыночную стоимость выпускаемых специалистов, связать обучение технологиям документооборота с реальными потребностями и запросами организаций.

Системы ЭУД, ориентированные на бизнес-процессы, как правило, предназначены для специфических вертикальных и горизонтальных приложений, иногда ориентированные на использование в определенной индустрии. Эти решения, как правило, обеспечивают полный жизненный цикл работы с документами, включая технологии работы с образами, управления записями и потоками работ, управление контентом и т.д.

Корпоративные системы ЭУД обеспечивают корпоративную инфраструктуру для создания, совместной работы над документами и их публикации, доступную, как правило, всем пользователям в организации. Основные возможности этих систем аналогичны системам, ориентированным на бизнес-процессы. Однако, их отличительной особенностью является способ использования и распространения. Аналогично таким средствам как текстовые редакторы и электронные таблицы, корпоративные системы ЭУД являются стандартным, "приложением по умолчанию" для создания и публикации документов в организации. Как правило, эти средства не ориентированы на использование только в какой-то определенной индустрии или для узко определенной задачи. Они предлагаются и внедряются как общекорпоративные технологии, доступные практически любой категории пользователей.

Системы управления контентом обеспечивают процесс отслеживания создания, доступа, контроля и доставки информации вплоть до уровня разделов документов и объектов для их последующего повторного использования и компиляции. Потенциально доступность информации не в виде документов, а в меньших объектах облегчает процесс обмена информацией между приложениями.

Системы управления информацией, называемые также порталами, обеспечивают агрегирование, управление и доставку информации через сети Internet, intranet и extranet. Эти технологии обеспечивают фундамент создания информационных порталов. Системы управления информацией дают возможность организациям накапливать и использовать экспертизу в распределенной корпоративной среде на основе использования бизнес-правил, контекста и метаданных. Хотя большинство доступных сегодня технологий обеспечивают, в основном, статические публикации, обеспечение большей интерактивности и средств совместной работы – дело ближайшего будущего. Системы управления образами преобразуют информацию с бумажных носителей в цифровой формат, как правило, это TIFF (Tagged Image File Format), после чего документ может быть использован в работе уже в электронной форме.

Системы управления потоками работ (workflow) обеспечивают систематическую маршрутизацию работ любого типа в рамках

структурированных и неструктурированных бизнес-процессов. Они используются в целях ускорения бизнес-процессов, увеличения эффективности и степени контролируемости процессов в организации.

Следует сказать, что в настоящее время существует немало подходов к решению проблем управления электронным документооборотом. Кроме так называемых универсальных систем электронного документооборота (EDMS) решить задачу управления документами можно при помощи следующих основных видов программных средств:

- средств групповой работы (типа ПО Lotus Notes, Novell GroupWise и MS Exchange);
- систем управления особыми видами документов (в частности, PDM-систем);
- специальных модулей управления документооборотом в составе корпоративных информационных систем для предприятий (SAP R/3, Baan и др.).

Чем же универсальные системы электронного документооборота отличаются от них? И в чем преимущества их использования по сравнению с другими подходами?

Пожалуй, в каждом конкретном случае нужно исходить из особенностей целей и задач, которые ставит предприятие при внедрении системы управления документооборотом. Например, для одних задач наилучшим решением будет ПО Novell GroupWise, для других же оптимальным будет внедрение именно EDMS-системы. Важным свойством универсальных систем электронного документооборота является то, что они предоставляют возможность решения большого числа задач управления документами. Тем не менее, следует отметить, что полностью все задачи управления документооборотом они все равно не смогут решить (из-за их динамически меняющегося и, практически, неисчерпаемого многообразия). Кроме того, внедрение специализированного ПО может оказаться дешевле и эффективней.

И все-таки у EDMS-систем есть ряд серьезных достоинств. В частности, по сравнению с ПО поддержки групповой работы, большинство наиболее распространенных в мире EDMS-систем обладает следующими достоинствами:

- практическая неограниченность размера электронного архива;
- встроенная поддержка полнотекстовой индексации;
- поддержка версий и подверсий документов;
- возможность хранения документов в архиве в исходном формате;
- возможность ведения журналов действий пользователей.

Возникает резонный вопрос: можно ли на основе ПО поддержки групповой работы создать полноценную систему управления документооборотом предприятия? Следует сказать, что это вполне возможно, т.к. в некоторых программных средствах поддержки групповой работы есть развитые среды разработки приложений (например, в Lotus Notes). Более того, на рынке уже существуют приложения (разработанные, например, на базе того же ПО Lotus Notes), в которых вышеперечисленные возможности EDMS-систем также реализованы. Однако надо заметить, что эти возможности реализованы в EDMS-системах уже изначально. Кроме того, при работе с групповым ПО в ряде случаев возникают вопросы снижения быстродействия приложений при работе с большими и сверхбольшими базами данных. Также групповое ПО ориентировано, прежде всего, на решение задач организации взаимодействия между пользователями. По своей сути, оно является расширением программ электронной почты и не всегда подходит для работы с крупными архивами документов. На рынке сформировалось мнение, что применение группового ПО (в качестве основы для создания систем электронного документооборота) наиболее уместно на малых и средних предприятиях. Следует также отметить, что EDMS-системы в большинстве случаев имеют шлюзы в групповое ПО и, при необходимости, могут использовать имеющиеся в нем почтовые программы.

В свою очередь, специализированные системы управления документами можно условно разделить на 2 категории [2]. В первую категорию входят программные средства (достаточно давно разработанные), изначально ориентированные на узкопрофильное применение и не имеющие средств интеграции с другими информационными системами. Подобные системы еще есть на предприятиях, и, в основном, они представлены местными разработчиками. Помимо управления документооборотом, такие программные системы предназначены для решения конкретных финансовых задач, управления персоналом, ведения складского учета и т.д. Отметим, что у большинства таких систем есть существенные ограничения при их применении на предприятиях. В частности, для каждого из приложений в такой системе необходимо поддерживать свою базу данных (нередко эти базы данных никак не связаны друг с другом). Затраты на поддержание инфраструктуры таких систем настолько велики, что они постепенно изживают себя. Во вторую категорию входят специализированные PDM-системы, предназначенные для управления производственной информацией и имеющие средства интеграции с другими программными системами. PDM-системы работают уже и с электронными образцами сложнейших объектов (например, кораблей и машин).

Что касается корпоративных информационных систем (КИС), то в большинстве из них встроены модули поддержки делопроизводства. Однако возможности этих модулей достаточно ограничены, т.к. практически невозможно создать универсальную и полнофункциональную КИС. Кроме того, цена западных КИС настолько велика (многие сотни тысяч и миллионы долларов), что не всякое предприятие сможет позволить себе ее внедрение. Следует все же отметить, что в большинстве распространенных EDMS-систем реализована интеграция с наиболее известными КИС (в частности, с SAP R/3, Oracle Applications и др.). Именно возможность интеграции с различными приложениями является одним из характерных свойств EDMS-систем. Благодаря наличию этого свойства, EDMS-системы могут выступать в качестве связующего звена между различными системами, функционирующими на предприятии, создавая, тем самым, основу для организации всего его делопроизводства.

У EDMS-систем есть и другие свойства, благодаря наличию которых применение универсальных систем электронного документооборота позволяет существенно оптимизировать управление современным предприятием.

Опишем основные свойства EDMS-систем.

Открытость. Все EDMS-системы построены по модульному принципу, а их API-интерфейсы являются открытыми. Это позволяет добавить к EDMS-системам (при необходимости) новые функции или расширять уже имеющиеся. В настоящее время разработка приложений, хорошо интегрируемых с EDMS-системами, стала отдельным видом бизнеса в отрасли промышленного производства ПО, и множество третьих фирм готовы предложить здесь свои услуги. Возможность относительно простого добавления к EDMS-системам множества модулей разработки третьих фирм значительно расширяет их функциональные возможности. Например, для EDMS-систем разработаны модули ввода документов со сканера, связи с электронной почтой, с программами пересылки факсов и др.

Высокая степень интеграции с прикладным ПО. Важной особенностью EDMS-систем является также высокая степень их интеграции с прикладными программами за счет использования технологий OLE Automation, DDE, ActiveX, ODMA, MAPI и др. А непосредственно при работе с документами вообще нет необходимости пользоваться утилитами EDMS-системы. Пользователи имеют дело только с обычными прикладными программами: в момент инсталляции клиентской части EDMS-системы прикладные программы дополняются новыми функциями и элементами меню. Например, пользователь текстового процессора MS Word, открывая файл, сразу видит библиотеки и папки с документами EDMS-системы (откуда он и выбирает необходимый ему документ). При

сохранении документ автоматически помещается в базу данных EDMS-системы. То же относится и к другим офисным и специализированным программам.

Особенности хранения документов. EDMS-системы работают, преимущественно, на распределенных архитектурах и используют разнообразные комбинации технологий сбора, индексирования, хранения, поиска и просмотра электронных документов. Во всех EDMS-системах реализована иерархическая система хранения документов. Можно сказать, что применяется принцип шкаф/полка/папка. Каждый документ помещается в папку, которая, в свою очередь, находится на полке и т.д. Количество уровней вложения при хранении документов не ограничено. Один и тот же документ может входить в состав нескольких папок и полок за счет применения механизма ссылок (исходный документ, в этом случае, остается неизменным и хранится в определенном администратором EDMS-системы месте). В ряде EDMS-систем реализованы еще более мощные возможности хранения за счет установления связей между документами (так называемые связанные документы). Причем эти связи можно устанавливать и редактировать в графическом виде.

Любому документу в EDMS-системе присущ определенный набор атрибутов (например, его название, автор документа, время его создания и др.). Набор атрибутов может меняться от одного типа документа к другому (в пределах одного типа документов он сохраняется неизменным). В EDMS-системах атрибуты документа хранятся в реляционной базе данных. Для каждого типа документов с помощью визуальных средств создается шаблон карточки, где в понятном графическом виде представлены наименования атрибутов документа. При введении документа в EDMS-систему берется необходимый шаблон и заполняется карточка (заносятся значения атрибутов). После заполнения карточка оказывается связанной с самим документом.

В большинстве случаев серверная часть EDMS-системы состоит из следующих логических компонентов (которые могут располагаться как на одном, так и на нескольких серверах):

- хранилища атрибутов документов (карточек);
- хранилища документов;
- сервисы полнотекстовой индексации.

Под хранилищем документов обычно понимается хранилище содержимого документов. Хранилище атрибутов и хранилище документов часто объединяют под общим названием архив документов. Для хранения

атрибутов в большинстве EDMS-систем используются СУБД Oracle, Sybase, MS SQL Server и Informix, обеспечивающие поиск документов по атрибутам.

Для хранения непосредственно содержимого документов в большинстве EDMS-систем применяются файл-серверы MS Windows NT, Novell NetWare, UNIX и др. В этом случае могут быть реализованы и гетерогенные комбинации сетевых сред. Например, база данных с атрибутами документов может работать под управлением ОС UNIX в сети TCP/IP, а сами документы могут храниться под ОС Novell NetWare в сети IPX/SPX. Следует отметить, что большим преимуществом EDMS-систем является хранение документов в исходном формате (и автоматическое распознавание EDMS-системами множества форматов файлов).

В последнее время все большую популярность приобретает хранение документов вместе с атрибутами в базе данных. Такой подход имеет свои преимущества и недостатки. Преимуществом является значительное повышение безопасности доступа к документам, а основным недостатком низкая эффективность работы с документами при большом объеме хранимой информации. При данном подходе также требуется использование мощных серверов с большими объемами оперативной памяти и жестких дисков. Кроме того, в случае сбоя базы данных восстановить хранившиеся в ней документы будет очень непросто. Необходимо также строго привязываться к конкретной СУБД.

Сервис полнотекстовой индексации обычно является опциональным. Полнотекстовая индексация предназначена для индексации содержимого документов, чтобы, при необходимости, пользователь мог найти документы с указанными в строке поиска словами. Данный сервис должен работать с документами в их исходных форматах, т.е. должен понимать все поддерживаемые в EDMS-системе форматы файлов. Он также должен поддерживать язык, с которым работает пользователь.

Модули EDMS-систем, отвечающие за документооборот, принято называть модулями маршрутизации документов. В общем случае используется свободная и жесткая маршрутизация документов. При свободной маршрутизации любой участвующий в документообороте пользователь может по своему усмотрению изменить существующий (или задать новый) маршрут прохождения документов. При жесткой маршрутизации маршруты прохождения документов строго регламентированы, и пользователи не вправе их менять. Однако при жесткой маршрутизации может выполняться обработка логических операций, когда маршрут изменяется при выполнении каких-либо заранее заданных условий (например, отправке документа руководству при превышении конкретным пользователем своих должностных полномочий, предположим,

финансовых). В большинстве EDMS-систем модуль маршрутизации входит в комплект поставки, в некоторых его необходимо приобретать отдельно. В частности, полнофункциональные модули маршрутизации разрабатывают и поставляют третьи фирмы.

В EDMS-системах реализованы надежные средства разграничения полномочий и контроля за доступом к документам. В большинстве случаев с их помощью определяются следующие виды доступа (набор задаваемых полномочий зависит от конкретной EDMS-системы):

- полный контроль над документом;
- право редактировать, но не уничтожать документ;
- право создавать новые версии документа, но не редактировать его;
- право аннотировать документ, но не редактировать его и не создавать новые версии;
- право читать документ, но не редактировать его;
- право доступа к карточке, но не к содержимому документа;
- полное отсутствие прав доступа к документу.

Во время работы с EDMS-системой каждое действие пользователя протоколируется, и, таким образом, вся история его работы с документами может быть легко проконтролирована.

При одновременной работе с документом сразу нескольких пользователей (особенно, когда его необходимо согласовывать в различных инстанциях) очень удобным свойством EDMS-систем является использование версий и подверсий документа. Предположим, исполнитель создал первую версию документа и передал ее на рассмотрение следующему пользователю. Второй пользователь изменил документ и создал на его основе уже новую версию. Затем он передал свою версию документа в следующую инстанцию третьему пользователю, который создал уже третью версию. Спустя определенное время, ознакомившись с замечаниями и исправлениями, первый исполнитель документа решает доработать исходную версию и на ее основе создает подверсию первой версии документа. Достоинством EDMS-систем является реализованная в них возможность автоматического отслеживания версий и подверсий документов (пользователи всегда могут определить, какая именно версия/подверсия документа является наиболее актуальной по порядку или времени их создания).

В состав большинства EDMS-систем входят утилиты для просмотра документов (так называемые viewers), понимающие многие десятки форматов файлов. С их помощью очень удобно работать, в частности, с графическими файлами (например, с файлами чертежей в САПР). Помимо базового комплекта

утилит просмотра (входящего в каждую EDMS-систему), у третьих фирм можно приобрести дополнительные утилиты, хорошо интегрируемые с EDMS-системами.

При организации групповой работы над документами обычно весьма полезна возможность их аннотирования. Так как в некоторых случаях пользователи лишены прав на внесение каких-либо изменений в документ в процессе его согласования, то они могут воспользоваться возможностью его аннотирования. В большинстве EDMS-систем аннотирование реализуется за счет включения в карточку документа атрибута для аннотации и передачи пользователям прав на редактирование такого поля карточки. Но такое решение не всегда приемлемо (особенно при аннотировании графического документа). В связи с этим в некоторых EDMS-системах реализована так называемая функция красного карандаша, с помощью которой можно графически указать недостатки на самом изображении. Программные средства, в которых реализована функция красного карандаша, широко предлагаются третьими фирмами.

Клиентами большинства EDMS-систем могут быть ПК с ОС MS Windows 95, Windows NT. В некоторых EDMS-системах допускается также использование платформ UNIX и Macintosh. Кроме того, все современные EDMS-системы позволяют работать с документами через стандартные Web-браузеры. Так как Web-браузеры могут быть размещены на разнообразных клиентских платформах, то это облегчает решение проблемы обеспечения работы EDMS-систем в гетерогенных сетевых средах. При использовании Интернет-технологий у EDMS-систем появляется еще один серверный компонент, отвечающий за доступ к документам через Web-браузер.

Последние несколько лет отмечены постоянным ростом внедряемого в практику делопроизводства качественно нового инструментария работы с документами в организациях - систем электронного документооборота (в литературе встречаются и другие названия для данного класса систем). Изучение их возможностей и особенностей является важной задачей не только с точки зрения внедрения в учебный процесс Историко-архивного института, но с позиций изучения источниковедческих особенностей документов, создаваемых и функционирующих внутри подобных систем. Отдельной практически значимой исследовательской проблемой является также изучение проблем организации приёма комплексов документов из этих систем, либо систем в целом, на архивное хранение.

### **1.3 Функции и задачи систем управления документами.**

Определим тот необходимый уровень функциональных возможностей, которому должна соответствовать современная система автоматизации делопроизводства. Любая система должна поддерживать полный жизненный цикл документа в организации - от его первичной регистрации до списания в дело.

Жизненный цикл документа состоит из двух основных стадий:

Стадия разработки документа, которая может включать:

- собственно разработку содержания документа;
- оформление документа;
- утверждение документа.

В том случае если документ находится на стадии разработки, то он считается неопубликованным, и права на документ определяются правами доступа конкретного пользователя.

Стадия опубликованного документа, которая может включать:

- активный доступ;
- архивный документ: краткосрочного хранения; долгосрочного хранения;
- уничтожение документа.

Когда документ переходит на вторую стадию, он считается опубликованным, тогда права на документ остаются только одни - доступ на чтение. В качестве примера опубликованного документа можно привести шаблон стандартного бланка предприятия.

Организация работы современного офиса предполагает оперативное информационное обслуживание. Сроки подбора всех документов по запрошенному руководством вопросу не должны превышать нескольких часов. В то же время многие документы (докладные, аналитические записки, отчёты, стенограммы заседаний и т.п.) содержат информацию о множестве вопросов, не отражённых в заголовках. Поэтому оперативную и максимально полную подборку можно получить, только используя компьютерный поиск по текстам документов в электронной форме. Для этого в СУД предусмотрена организация электронного архива, т.е., простыми словами, не надо идти в хранилище, искать дело, заполнять лист-заместитель, при необходимости – изготавливать копию документов, при возврате дела – осуществлять полистный просмотр, раскладку на место и т.п. – все процедуры автоматизируются.

Электронный архив предприятия - это комплекс программного и аппаратного обеспечения, предназначенный для решения следующих задач:

Организация хранения электронных документов. Необходимо обеспечивать хранение произвольного количества электронных документов на

разнообразных носителях. Носители электронных документов характеризуются двумя основными параметрами, стоимостью хранения одного мегабайта информации и скоростью доступа к информации. Причем эти два параметра обратно пропорциональны друг другу - в зависимости от решаемых задач, необходимо находить оптимальное соответствие и выбирать определенный носитель информации. Также на выбор носителя влияет срок хранения информации. Иногда для ряда задач необходимо создавать систему хранения, которая состоит из разнотипных носителей. Например, для оперативного доступа требуется применение высокоскоростных жестких дисков, а для архивного хранения достаточно роботизированных библиотек оптических дисков. Соответственно для таких систем хранения необходимо решать задачи не только совместной работы данных носителей информации, но и обеспечивать миграцию документов между носителями. Эта миграция может осуществляться либо путем настройки системы администратором (например, после истечения 90 дней со дня создания документа он должен автоматически переместиться на более медленный и дешевый носитель информации), либо автоматически в зависимости от частоты обращения пользователей к тому или иному документу.

Организация учета бумажных и микрографических документов. Архивная система кроме работы с электронными документами должна учитывать бумажные и микрографические документы - система должна хранить только электронную карточку на документ данного типа и поддерживать контроль стандартных архивных операций: выдача документа, возврат документа и т. п.

Организация поиска документов. Наряду с организацией хранения документов необходимо их быстро и эффективно искать. Если со скоростью поиска все понятно, то для пояснения понятия эффективности поиска коротко рассмотрим модели поиска. Существует два подхода к поиску документов. Первый подход состоит в том, что в процессе поиска ищется документ, который точно существует в системе, и задача состоит в том, чтобы процесс поиска свелся к нахождению требуемого документа или документов. Этот подход применяется в 90% всех случаев поиска. Второй подход состоит в том, что ищутся все документы по интересующему вопросу. Для этого подхода присущи такие понятия, как полнота - соответствие между найденными документами по данному запросу и действительному списку документов и шум - соответствие документов, отвечающих запросу и документов, нерелевантных ему. Поддержка защиты документов от несанкционированного доступа и аудита работы. Архивная система должна иметь защиту на уровне документа - каждый документ должен иметь ассоциированный с ним список пользователей, имеющих право совершать с ним определенные операции.

Поддержка просмотра документов без загрузки приложений, которые порождают документ. Архивная система должна поддерживать специальные программы просмотра, позволяющие получить доступ к документам разнообразных форматов без загрузки ресурсоёмких приложений.

Поддержка аннотирования документа. Иногда для обеспечения коллективной работы с документом необходимо позволять наносить комментарии на документ, не изменяя его основного содержания. Комментарии (знаки, текст, цветные пометки) хранятся в слоях, которые могут быть привязаны к имени автора, создавшего эти комментарии.

Поддержка коллективной работы с документом. Выражается в поддержке целостности документов, для этого должны быть реализованы библиотечные функции выписки и возврата документов на/с редактирования, что предотвращает одновременное редактирование одного документа несколькими пользователями, устраняя тем самым возможные конфликты. Для того чтобы позволить в рамках одного документа, работать, одновременно, нескольким пользователям, вводится понятие версии и подверсии документа – один документ может содержать несколько версий, а каждая версия – несколько подверсий. Кроме того, в рамках одного документа и одной версии (подверсии) документа может существовать несколько представлений документов (в разных форматах).

Поддержка составных документов. Каждый документ может представлять собой совокупность других документов. Такой документ носит название составного или контейнера, а в делопроизводстве такой документ носит название "дело", по своим характеристикам он аналогичен простому документу документы могут быть объединены в составной документ с помощью нескольких типов связей. Эти связи определяют, какие версии объединяются в контейнер, например, последняя по дате, последняя отредактированная, старшая версия и т. п. Эти связи определяют, как будет осуществляться сборка документа в составной документ. Для составных документов должно существовать приложение, которое будет осуществлять окончательную сборку такого документа.

Поддержка распространения опубликованных документов. Как только документ публикуется его иногда необходимо распространять. Это происходит двумя основными путями, либо через систему электронной почты, путем рассылки, либо через Internet, путем публикации на Web-сервере. Кроме базовых архивных задач существует ряд пограничных функций, связанных с документами разных типов (электронных, бумажных, микрографических) и переводом их из одного типа в другой. Данная проблема подразделяется на два основных класса.

Персональный ввод бумажных документов. Ввод небольшого количества разнотипных бумажных документов осуществляется с помощью планшетных или персональных сканеров. После операции сканирования документ вручную индексируется путем заполнения карточки документа.

Массовый (поточный) ввод бумажных документов. Основное отличие от предыдущего состоит в том, что обрабатывается большое количество однотипных документов. В качестве примеров приложений данной технологии в конкретных предметных областях можно привести: систему ввода и хранения платежных поручений в банке, систему обработки анкет опроса населения, систему обработки результатов голосования.

При реализации технологии массового ввода документов можно рассматривать два основных класса задач.

Задача извлечения данных из бумажных документов. Например, имеется форма с результатами опроса населения. Необходимо ввести большое количество анкет, извлечь из них данные и загрузить в некоторую базу. В этом случае нас интересуют только извлеченные структурированные данные, а не сами изображения документов.

Задача извлечения данных из бумажных документов с сохранением изображения документа. Если вы обрабатываете не форму с опросами населения, а платежное поручение клиента, то имеет смысл после извлечения данных сохранить изображение документа для того случая, когда потребуется анализ исходного документа.

При организации систем документооборота одной из основных составляющих являются системы маршрутизации и контроля исполнения, которые оперируют документами, хранящимися в архиве.

Свободная маршрутизация. Выделяется два основных типа маршрутов документов. Последовательная маршрутизация - документ последовательно проходит одного исполнителя за другим. Передача документа от одного пользователя к другому может происходить по истечении контрольного времени, либо после завершения работы одним из них. Параллельная маршрутизация - документ одновременно поступает всем исполнителям, а завершение маршрута происходит, когда один либо все пользователи завершат работу с документом.

Системы электронной почты. Минимальной достаточной системой, обеспечивающей маршрутизацию документов является система электронной почты, которая осуществляет параллельное распространение документов (маршрутизация отличается от распространения или рассылки тем, что маршрутизируемый документ возвращается в начало маршрута, например к инициатору, а рассылаемый документ уходит к исполнителю без контроля факта

возврата). С помощью дополнительных приложений система электронной почты может обеспечивать последовательную маршрутизацию документов.

Свободная маршрутизация документов с контролем исполнения. Под контролем исполнения понимается следующая функциональность:

- Контроль доставки задания - инициатору выдается информация о том, что его задание достигло места назначения (исполнителя).
- Контроль прочтения задания - инициатору выдается информация о том, что с его заданием ознакомились сотрудники, для которых это задание было предназначено.
- Контроль выполнения - инициатору выдается информация о том, что задание выполнено.
- Мониторинг задания - инициатор всегда может посмотреть, кто и что сейчас делает с его заданием.
- Извещение о нарушении сроков исполнения - система документооборота может известить инициатора о том, что посланное им задание просрочено конкретным сотрудником.
- История выполнения заданий.
- Контроль качества исполнения - означает, что, если пользователь говорит о том, что задание исполнено, это еще не означает, что оно действительно исполнено, инициатор должен проверить качество исполнения, подтвердить или нет исполнение.

Информация может выдаваться в виде изменения статуса задания в окнах входящих и исходящих заданий или в виде нового задания сформированного системой инициатору либо с помощью сообщения по электронной почте.

Маршрутизация документов по заранее определенным маршрутам с контролем исполнения (жесткая маршрутизация). Маршруты могут быть более сложными, чем простые последовательные или параллельные:

- комбинированные из последовательных и параллельных элементов;
- условные, с переходами в зависимости от состояния тех или иных переменных маршрутов.

Такие маршруты становятся сложными для их задания "на лету", поэтому в этом случае используется специализированный графический редактор, позволяющий создать маршрут. Инициатор вызывает созданный и именованный маршрут и прикрепляет к нему документы - иницирует его. Система маршрутизации должна быть интегрирована с архивной системой, и реальные приложения для работы с документами не могут быть основаны только на файловой системе. И вот почему. Любой процесс маршрутизации документов -

это движение одного документа, а не множества его копий, как это происходит в системах электронной почты. Посылать один документ необходимо не только по соображениям экономии пространства, но и в основном для поддержания его целостности - в процессе маршрутизации многие пользователи пытаются вносить изменения в документ.

Рассмотренные возможности и функции обеспечивают построение любой частной системы документооборота на любом предприятии в любой предметной области. Естественно, для построения частного решения можно ограничивать функционал системы в зависимости от предъявляемых заказчиком требований.

## 2. ИССЛЕДОВАНИЕ РАБОТЫ ДЕКАНАТА

Инновационный Евразийский Университет – один из ведущих вузов Республики Казахстан, крупный центр образования, науки и культуры, обеспечивающий многоуровневую подготовку специалистов с 1991 года.

Основной целью Евразийского Инновационного университета является предоставление качественных образовательных услуг (профессиональных, общественных) на рынке региона через создание модели исследовательского университета, ведущими функциями которого являются: обучение студентов, научные исследования, разработка и внедрение результатов исследований в производство, функция обслуживания общества.

Для достижения поставленной цели необходимо правильно организовать учебный процесс и оперативно и грамотно управлять им. Эти функции возложены на определённые подразделения ИнЕУ.

Факультет является учебным, научным и административным структурным подразделением университета, осуществляющим подготовку специалистов по одной или нескольким родственным специальностям, повышение квалификации руководящих, инженерно-технических и педагогических кадров, а также руководство научно-исследовательской работой преподавателей и студентов.

Основная задача факультета - обеспечение координации деятельности структурных подразделений факультета:

- по профориентации, организации набора, формированию и сохранению контингента студентов;
- по организации учебного процесса и всех видов практики студентов; по развитию методической, учебно-лабораторной, производственной и материальной базы структурных подразделений факультета; по организации научно-исследовательской и международной деятельности;
- по подготовке научно-педагогических кадров и повышению их квалификации;
- по организации воспитательной и культурно-массовой работы со студентами, в том числе проживающими в общежитии и на квартирах; по организации делового сотрудничества с заказчиками и работодателями и трудоустройству выпускников;
- по организации работы совещательных органов (совет факультета, научно-методический совет, совет кураторов, старостат, стипендиальная комиссия);
- по организации делопроизводства на факультете и ведению учебной документации студентов.

Для выполнения поставленных задач факультет выполняет функциональные обязанности:

Профориентация, организация набора, формирование и сохранение контингента студентов:

- участие в работе приемной комиссии университета, организация работы профориентационных групп факультета дуального обучения;
- подготовка рекламно-информационных материалов по специальностям факультета дуального обучения;
- установление постоянного делового сотрудничества со школами, колледжами, органами образования и общественностью;
- привлечение выпускников школ и колледжей для участия в культурно-массовых и спортивных мероприятиях университета;
- обеспечение набора студентов, формирование академических групп и обеспечение сохранности контингента.

Организация учебного процесса и всех видов практики студентов:

- обеспечение соответствия рабочих учебных планов и программ образовательным стандартам и контроль за их выполнением;
- контроль за организацией учебного процесса и состоянием учебной дисциплины;
- анализ успеваемости и посещаемости студентов, организация системы контроля знаний;
- работа с заказчиками и родителями студентов по вопросам обучения и учебной дисциплины;
- контроль за организацией всех видов практики и самостоятельной работы студентов, обеспечение отчетности по итогам практики;
- организация дипломного проектирования и работы государственных аттестационных комиссий (формирование составов, подготовка балльных ведомостей, анализ результатов защиты, обеспечение отчетности по итогам работы ГАК);
- организация экзаменационных сессий (разработка расписания, подготовка зачетно- экзаменационных ведомостей, контроль за ходом сессии, анализ и отчетность по результатам);
- контроль за предоставлением дополнительных образовательных и консультационных услуг на факультете дуального обучения.

Развитие методической, учебно-лабораторной, производственной и материальной базы:

- руководство научно-методическим советом, формирование тематического плана научно-методической работы;
- руководство разработкой рабочих учебных планов и программ высшего образования по всем направлениям;
- стимулирование методической работы преподавателей (создание новых учебников и учебных пособий, новых технологий обучения);
- совершенствование учебно-лабораторной, производственной и материальной базы кафедр и других структур факультета;
- обеспечение сохранности и поддержание санитарного и эстетического состояния закрепленных учебно-лабораторных площадей, учебной мебели и оборудования.

Организация научно-исследовательской и международной деятельности:

- обеспечение организационного единства учебного процесса и выполняемых на факультете научных исследований, улучшение их качества, увеличение объема;
- создание необходимых условий и стимулирование студентов и сотрудников к участию в научной работе и международной деятельности факультета;
- обеспечение участия подразделений факультета в подготовке научных проектов для получения международных грантов на проведение научных исследований;
- организация сотрудничества с зарубежными вузами по вопросам обучения и совместных научно-исследовательских работ.

Подготовка научно-педагогических кадров и повышение их квалификации:

- создание необходимых условий для обучения магистрантов, аспирантов и соискателей кафедры;
- оказание помощи в выпуске наукоемкой продукции и научных трудов ППС и сотрудников кафедр;
- контроль за своевременным повышением квалификации ППС и обеспечение на кафедре контрактной системы трудоустройства на конкурсной основе;
- стимулирование преподавателей и сотрудников к повышению их научно-педагогической квалификации;

Организация воспитательной и культурно-массовой работы со студентами, в т.ч. проживающими в общежитии и на квартирах:

- обеспечение участия студентов факультета в общеуниверситетских мероприятиях;
- организация мероприятий по профилактике правонарушений (лекции, встречи ,тд);
- обеспечение санитарного и правового порядка в общежитии;
- организация мероприятий по формированию здорового образа жизни студентов и сотрудников факультета дуального обучения;
- формирование у сотрудников и преподавателей ответственного отношения к выполнению своих должностных обязанностей;
- формирование у сотрудников, ППС и студентов бережного отношения к имуществу университета, формирование положительного имиджа факультета и университета в целом;

Организация делового сотрудничества с заказчиками и работодателями, трудоустройство выпускников:

- организация и проведение презентации выпускников с приглашением заказчиков и работодателей;
- формирование банка данных на выпускников и фотоматериалов для музея факультета;
- установление связей с предприятиями, организациями и физическими лицами по вопросам трудоустройства выпускников.

Организация работы совещательных органов:

- планирование и руководство работой Совета факультета;
- планирование и организация работы Научно-методического совета;
- организация работы Совета кураторов, старостата и стипендиальной комиссии.

Организация делопроизводства на факультете и ведение документации студентов:

- подготовка приказов по движению контингента студентов (о формировании академических групп, присвоении шифров, переводе, восстановлении и отчислении и т.д.);
- подготовка распоряжений по факультету дуального обучения;
- оформление учебных карточек, зачетных книжек и студенческих билетов;
- подготовка балльных и зачетно-экзаменационных ведомостей, регистрация, выдача и учет экзаменационных листов (направлений), академических справок, характеристик, справок с места учебы и т.д.;

- ведение журналов успеваемости и посещаемости студентов.

Деканат — организационный центр по управлению работой факультета, возглавляемый деканом. Деканат выполняет функции координации и административного обеспечения учебного процесса, ведения делопроизводства.

В настоящее время в университете имеются следующие факультеты:

- факультет очного обучения инженерной академии
- факультет заочного обучения инженерной академии
- факультет очного обучения академии образования
- факультет заочного обучения академии образования
- факультет очного обучения академии управления
- факультет заочного обучения академии управления
- факультет магистерского и второго высшего образования
- факультет дистанционного обучения

Кроме работы со студентами, большую часть работы секретарей составляет работа с документами. Ниже приведен список приказов и распоряжений, с которыми работает деканат:

- Приказ об отчислении в связи с переводом
- Приказ об отчислении в связи с нарушением условий договора при платном обучении
- Приказ об отчислении как не вышедшего из академического отпуска.
- Приказ об отчислении в связи с успешной защитой дипломных работ..
- Приказ об отчислении за академическую неуспеваемость
- Приказ об отчислении по собственному желанию
- Приказ об отчислении в связи с призывом на воинскую службу
- Приказ об отчислении как не преступившего на повторный курс обучения
- Приказ о восстановлении (ранее отчисленного)
- Приказ о зачислении по переводу
- Приказ о зачислении на основании свидетельств на государственные образовательные гранты
- Приказ о зачислении на договорной основе.
- Приказ о переводе внутри вуза
- Приказ о переводе на заочный факультет.
- Приказ о переводе с курса на курс.

- Приказ о предоставлении академического отпуска
- Приказ о выходе из академического отпуска
- Приказ о предоставлении академического отпуска по беременности и родам
- Приказ об изменении фамилии
- Приказ о формировании академических групп
- Приказ о назначении старост.
- Приказ о назначении кураторов групп
- Распоряжение о допуске к учебным занятиям
- Приказ о допуске к государственному экзамену
- Распоряжение о защите дипломного проекта.
- Приказ о допуске к дипломному проектированию
- Приказ о направлении на практику
- Приказ о повторном обучении на курсе.

Помимо работы с перечисленными документами, деканат также:

- выдает направления и справки
- ведет карточки студентов
- отправляет ежемесячный отчет в учебный отдел
- выдает студенческие билеты и зачетные книжки
- ежегодно заполняет форму со статистическими данными

Все документы стандартизированны и заполняются по образцу, который можно найти на сайте университета. Для каждого документа определен порядок визирования, который так же находится на сайте. Заполненные документы отправляются через курьерскую почту университета к адресатам. При обнаружении ошибки в документе или возникшем вопросе на каком-либо этапе, документ возвращается назад и, после исправления, проходит весь цикл заново. Кроме того, возникают задержки на каждом из этапов визирования. Из-за этого, в некоторых случаях прохождение документа по всей цепочке занимает длительное время.

В настоящее время в качестве автоматизированных рабочих мест деканата используется программа «Деканат». Она работает в связке с программой «Абитуриент». Рассмотрим данные программы.

Программа «Абитуриент» используется для создания базы данных абитуриентов, подающих документы во время работы приемной комиссии.

Она позволяет хранить следующие данные об абитуриентах:

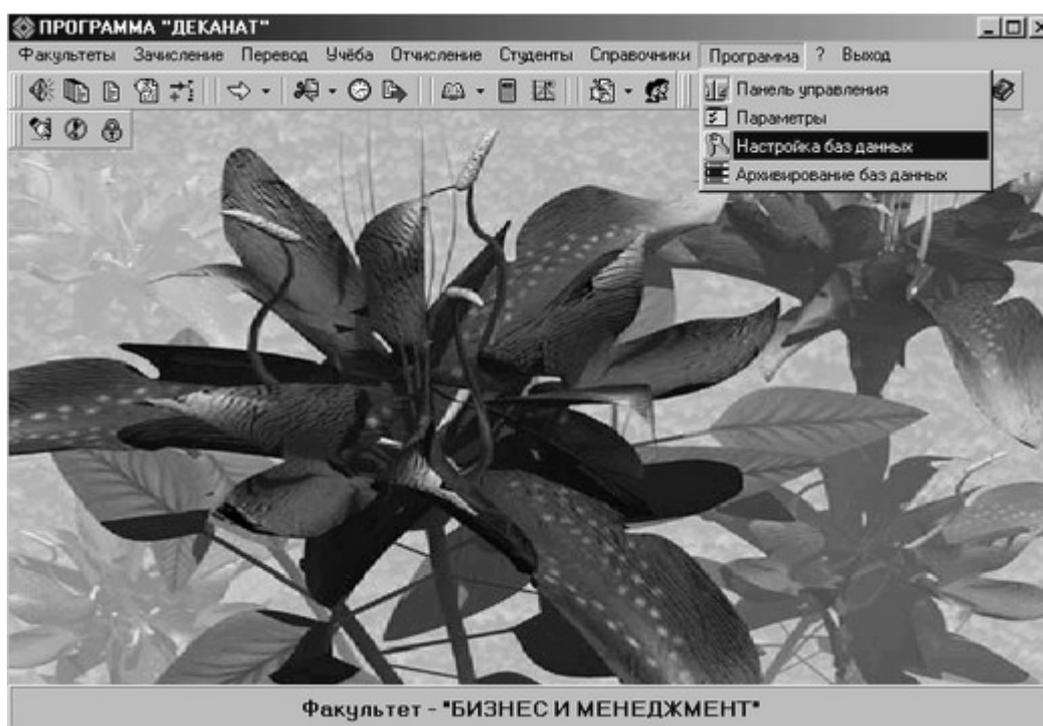
- ФИО
- Дата рождения

- Национальность
- Пол
- Адрес проживания
- Какое учебное заведение закончил
- Средний балл аттестата
- Баллы ЕНТ
- Данные о родителях
- Язык обучения
- Изучаемый иностранный язык

Программа позволяет производить поиск и сортировку по различным критериям. В программе реализованы списки стран, населенных пунктов, национальностей, учебных заведений, специальностей, что облегчает ввод однотипных данных и позволяет сократить количество ошибок. Для обеспечения безопасности в программе реализовано разграничение прав доступа. Внесение данных в программу осуществляют операторы во время работы приемной комиссии.

Программа «Деканат» предназначена для автоматизации учебной работы деканатов университета. Она решает определенный круг задач по статистической обработке информации и подготовке на печать необходимой документации.

Внешний вид программы представлен на рисунке 1



## Рисунок 1 - Программа «Деканат»

В программе реализованы следующие возможности:

- Зачисление абитуриентов в студенты (перенос данных из программы «Абитуриент», добавление новых студентов в самой программе).
- Формирование академических групп, назначение кураторов и старост, стипендии.
- Учебный процесс (составление списка сдаваемых студентами дисциплин, формирование ведомостей, проставление оценок, практика, сводные ведомости, статистика по итогам сессии, дипломное проектирование, приказ по закреплению дисциплин и недельные сетки учебных занятий).
- Переводы студентов (на следующий курс, в другую группу, на другой факультет).
- Перерывы в учебе (академические отпуска и командировки), оставление на повторный курс обучения.
- Отчисление студентов по различным причинам (за неуплату, после окончания учебы, по собственному желанию и др.).
- Восстановление отчисленных за неуплату студентов.
- Обработка информации о студентах (различные справки и приказы на студентов, личные карточки и академические справки, изменение фамилии и др.).
- Гибкий справочник факультетов, позволяющий создавать, изменять, делить факультеты с различными направлениями (эти функции доступны только администратору).
- Защита программы и баз данных паролями, каждый пользователь имеет ограниченный доступ только к информации деканата, в котором работает, а администратор – ко всем.
- Программа снабжена всеми необходимыми справочниками (специальности, студенческие группы, дисциплины, кадровый состав, населенные пункты, национальности).
- Автоматическая настройка конфигурации программы (параметры интерфейса и псевдонимы баз данных).

Связь между программой «Абитуриент» и программой «Деканат» осуществляется при помощи специальной программы импорта данных.

Из недостатков данных программ можно выделить:

- Программы «Деканат» и «Абитуриент» используют различные базы данных и различные структуры БД, в результате чего необходима специальная

программа импорта данных, кроме того необходимо следить за тем, чтобы информация в соответствующих справочниках обеих программ совпадала.

- Программа «Деканат» на каждом рабочем месте использует локальную базу данных, в результате также становится трудно поддерживать согласованность информации. Происходит дублирование информации.

- Из-за раздробленности данных, трудно получить статистику по всем студентам одновременно. Для получения статистики используется специальная программа, которая объединяет базы данных в одну.

- Не смотря на то, что в программе реализован перевод с факультета на факультет, объединение и дробление факультетов, эти операции трудновыполнимы из-за раздробленности базы данных. Для проведения этих операций необходимо слияние баз данных и снова их разделение, после проведения операций.

- Программа использует устаревшие формы приказов для печати. Кроме того, было выявлено, что в последнее время данные программы не используются и вся документация заполняется вручную.

Для решения описанных проблем возможно разработать систему автоматизации, которая будет учитывать все описанные недостатки.

### **3. РАЗРАБОТКА МЕТОДОВ СОЗДАНИЯ, ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ О СТУДЕНТАХ И МЕТОДОВ СОЗДАНИЯ ДОКУМЕНТАЦИИ.**

#### **3.1 Выбор средств автоматизации**

После принятия решения о необходимости создания системы автоматизации возникает вопрос, возможно ли использовать существующую СЭД или следует разработать систему.

Современные системы СЭД представляют большой набор возможностей, однако для данной задачи они не будут использоваться, кроме того, существующие СЭД обладают довольно высокой ценой.

Альтернативным вариантом является использование свободно распространяемых бесплатных технологий разработки. Поскольку от разрабатываемого программного обеспечения требуется поддержка клиент-серверной архитектуры, то возможно разработать веб-портал, который будет отвечать заданным требованиям. В этом случае в качестве клиента выступает браузер на машине пользователя, а в качестве сервера любой бесплатный веб-сервер. Такой подход обладает определенными преимуществами:

Клиентская и серверная кроссплатформенность. Веб браузеры входят в поставку всех современных операционных систем, следовательно клиентская часть не будет зависеть от используемой операционной системы. Программы веб-серверы так же доступны на всех распространенных операционных системах;

Удобство для пользователя. Поскольку приложение представляет собой веб-портал и запускается при помощи браузера, пользователям которые уже умеют работать с браузером нет необходимости изучать новые способы работы с программой, так как используются стандартные элементы управления такие как ссылки, веб формы и стандартная навигация браузера;

Отсутствует необходимость настройки клиентских машин. Пользователю не требуется устанавливать каких-либо дополнительных программ, а достаточно лишь зайти на портал через сайт;

Отсутствие повышенных требований к производительности. Веб приложения не предъявляют больших требований к ресурсам системы, таким образом возможность работы сохраняется даже на старых компьютерах. Это особенно актуально в условиях университета, поскольку во многих деканатах компьютеры очень старые.

## 3.2 Инструменты разработки

Рассмотрим основные инструменты, используемые при разработке веб-порталов.

При запросе браузером страницы сайта, и отправке ее сервером обратно используется протокол HTTP. HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных в первую очередь в виде текстовых сообщений. Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (англ. Uniform Resource Identifier) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы, но ими могут быть логические объекты или что-то абстрактное. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и так далее. Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

Достоинства протокола HTTP:

- простота. Протокол настолько прост в реализации, что позволяет с лёгкостью создавать не только клиентские приложения, но и примитивные серверы.
- расширяемость. Вы можете легко расширять возможности протокола благодаря внедрению своих собственных заголовков, сохраняя совместимость с другими клиентами и серверами. Они будут игнорировать неизвестные им заголовки, но при этом вы можете получить необходимый вам функционал при решении специфической задачи.
- распространённость. При выборе протокола HTTP для решения конкретных задач немаловажным фактором является его распространённость. Как следствие, это обилие различной документации по протоколу на многих языках мира, включение удобных в использовании средств разработки в популярные IDE, поддержка протокола в качестве клиента многими

программами и обширный выбор среди хостинговых компаний с серверами HTTP.

Недостатки протокола HTTP:

- большой размер пакетов. Использование текстового формата в протоколе порождает соответствующий недостаток: большой размер сообщений по сравнению с передачей двоичных данных. Из-за этого возрастает нагрузка на оборудование при формировании, обработке и передаче сообщений. Для решения данной проблемы в протокол встроены средства для обеспечения кэширования на стороне клиента. Нормативными документами по протоколу предусмотрено наличие прокси-серверов, которые позволяют получить клиенту документ с наиболее близкого к нему сервера. Так же в протокол было внедрено дельта-кодирование, чтобы клиенту передавался не весь документ, а только его изменённая часть;

- отсутствие «навигации». Хотя протокол разрабатывался как средство работы с ресурсами сервера, у него отсутствуют в явном виде средства навигации среди этих ресурсов. Например, клиент не может явным образом запросить список доступных файлов как в протоколе FTP. Предполагалось что конечный пользователь уже знает URI необходимого ему документа, закачав который он будет производить навигацию благодаря гиперссылкам. Это вполне нормально и удобно для человека, но затруднительно когда стоят задачи автоматической обработки и анализа всех ресурсов сервера без участия человека. Решение этой проблемы лежит полностью на плечах разработчиков приложений, использующих данный протокол;

- нет поддержки распределённости. Протокол HTTP разрабатывался для решения типичных бытовых задач где само по себе время обработки запроса должно занимать незначительное время или вообще не приниматься в расчёт. Но в промышленном использовании с применением распределённых вычислений при высоких нагрузках на сервер протокол HTTP оказывается беспомощен.

В 1999 году была выпущена спецификация протокола HTTP/1.1, который используется и в настоящее время.

Для того, чтобы браузер мог получить данные с сервера, на сервере должна стоять специальная программа, которая работает с протоколом HTTP. Такая программа называется веб-сервером или HTTP-сервером.

Основная задача HTTP сервера - это ожидание запросов от клиентов и отправка им ответов. Клиент (обычно веб-браузер) запрашивает ресурс (обычно HTML файл или графический файл). Сервер связывает запрос с файлом или направляет запрос программе, которая генерирует необходимые данные. После

этого сервер отсылает ответ обратно клиенту. Это можно представить в виде рисунка 2.

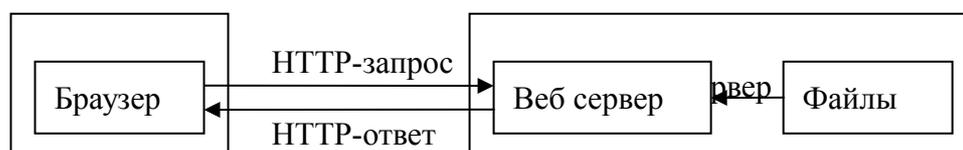


Рисунок 2 - Схема взаимодействия клиента с веб-сервером

Веб-сервер Apache удовлетворяет практически всем потребностям современных веб-разработок, но в то же время он достаточно прост, чтобы его устанавливали программисты для отладки своих программ.

Apache - это мощный, удобный, расширяемый и функциональный веб-сервер, полностью поддерживающий протокол HTTP/1.1 и распространяющийся с открытым исходным кодом. Он имеет отличные рабочие характеристики и поэтому используется более широко, чем все остальные Web-серверы вместе взятые. С апреля 1996 это самый популярный HTTP-сервер в Интернете; в августе 2007 года он работал на 51 % всех веб-серверов, в мае 2009 года — на 46 %. Сервер может работать практически на всех распространенных платформах. Существуют версии для Windows, Linux и BSD систем.

Первая публичная версия Apache появилась в апреле 1995 года, а в декабре вышел релиз 1.0. На протяжении всех этих лет и по сей день Apache остается совершенно бесплатным.

На данный момент разработка ведется в ветке 2.2, а в версиях 1.3 и 2.0 производится лишь исправление ошибок безопасности. Вторая версия претерпела значительное количество изменений, которые в первую очередь коснулись ядра программы и некоторых важных модулей.

Apache настраивается с помощью текстовых конфигурационных файлов. Основные параметры уже настроены «по умолчанию» и будут работать в большинстве случаев. Немаловажным преимуществом является то, что создатели активно общаются с пользователями и реагируют на все сообщения об ошибках.

Самая простая функция, которую может выполнять Apache – стоять на сервере и обслуживать обычный HTML-сайт. При получении запроса на определенную страницу сервер отправляет в ее ответ браузеру.

С помощью сервера Apache можно производить простую аутентификацию. Функция посложнее, которая заложена в протоколе HTTP/1.1 – аутентификация пользователей. С помощью штатных средств сервера Apache вы можете разграничить доступ к определенным страницам сайта для разных пользователей. Это нужно, например, для того чтобы сделать администраторский интерфейс к сайту. Для этого используются файлы `.htaccess` и `.htpasswd`, а также модули `mod_auth` и `mod_access`. Пользователи могут быть разбиты на группы, и для каждой из них можно назначить свои права доступа.

Сервер Apache поддерживает работу с технологией SSI. Для разделения дизайна и функциональной части сайта, а также для упрощения изменения статических объектов существует технология SSI. Она позволяет вам поместить всю повторяющуюся информацию в один файл, а затем вставлять в страницы ссылку на нее. Затем, если понадобится изменить несколько строк в этой информации, то придется поменять их только в одном файле. Сервер Apache поддерживает эту технологию и позволяет использовать серверные включения в полном объеме.

На сервере Apache можно создавать пользовательские директории для веб-сайтов. Если на одном сервере с установленной операционной системой семейства Unix и сервером Apache заведено несколько пользователей, то каждому из них можно создать отдельную директорию. Точнее, она будет создаваться автоматически вместе с псевдонимом. Это делается с помощью модуля `mod_userdir` и директивы `UserDir`. Так, например, можно папке `public_html` в домашней папке пользователя сопоставить адрес `www.site.ru/~user`. В общем-то, так и делается на серверах большинства сайтов, предоставляющих бесплатный хостинг. Администратор сервера может разрешить или запретить определенным пользователям создавать домашние страницы, использовать SSI и другие функции сервера. Полноценный же хостинг обычно предусматривает создание отдельного виртуального сервера для каждого пользователя.

Вы можете настроить виртуальные хосты. Благодаря им на одном физическом сервере можно содержать несколько виртуальных. Сервер Apache был одним из первых серверов, которые начали поддерживать виртуальные сервера (хосты). Эта возможность позволяет размещать на одном физическом сервере несколько полноценных сайтов. У каждого из них может быть свой домен, администратор, IP-адрес и так далее.

Apache поддерживает работу со скриптами PHP, CGI и другими

В данный момент большинство интернет-страниц являются динамическими. Это значит, что их внешний вид и наполнение формируется с помощью программного скрипта, написанного на одном из «языков» (их нельзя

в полной мере назвать языками, определение достаточно условно). Модуль `mod_cgi` позволяет вам размещать на сервере CGI-скрипты. Что касается PHP, то возможность интеграции его в Apache предусмотрена разработчиками самого PHP. Apache же выполняет только функции посредника между скриптом и компилятором.

В последней версии Apache существует модуль для работы с сильной криптозащитой SSL/TLS. На основе сервера Apache можно создавать не только простые любительские сайты, но и ресурсы, требующие серьезной криптографической защиты передаваемых данных. Специально для этого был разработан протокол SSL/TLS, а его поддержка была встроена в Apache 2.0. С помощью специального модуля можно осуществлять аутентификацию на основе именных сертификатов, что позволяет практически наверняка гарантировать подлинность пользователя.

Apache ведет подробные протоколы всего происходящего на сервере

Ну и, разумеется, сервер Apache может вести протокол всех действий, совершаемых с ним. Причем администратор может сам выбрать степень подробности протокола. Протоколы ведутся отдельно для ошибок, для успешных операций и для каждого виртуального хоста.

Очень часто статические HTML страницы не могут обеспечить нужной функциональности. В качестве примера можно привести разбор данных HTML-форм, поиск по сайту или регистрацию пользователей. В настоящее время для создания HTML страниц на сервере чаще всего используются CGI, либо PHP скрипты.

CGI (от англ. Common Gateway Interface — «общий интерфейс шлюза») — стандарт интерфейса, используемого для связи внешней программы с веб-сервером. Сам интерфейс разработан таким образом, чтобы можно было использовать любой язык программирования, который может работать со стандартными устройствами ввода/вывода. Такими возможностями обладают даже скрипты для встроенных командных интерпретаторов операционных систем, поэтому в тех случаях, когда нет нужды в сложной функциональности, могут использоваться даже такие простые командные скрипты. Поскольку CGI скрипт написан на одном из языков программирования, то его возможности ограничены только возможностями этого языка. CGI скрипты решают такие задачи, как работа с файлами, работа с базами данных, обработка ввода пользователей, регистрация и аутентификация пользователей.

PHP - это скриптовый язык, который может быть встроен в HTML, исполняемый на стороне сервера. Синтаксис языка в большинстве заимствован из C, Java и Perl с добавлением нескольких специфичных для PHP особенностей. Цель языка - позволить веб-разработчикам легко создавать

динамически генерируемые страницы. Отличием PHP от CGI является, то, что код PHP можно непосредственно встраивать в код HTML, что значительно облегчает разработку.

Главным образом, область применения PHP сфокусирована на написание скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять всё то, что выполняет любая другая программа CGI, например, обрабатывать данных форм, генерировать динамические страницы или отсылать и принимать cookies. Но PHP способен выполнять и множество других задач.

Существуют три основных области, где используется PHP.

- создание скриптов для выполнения на стороне сервера. PHP наиболее широко используется именно таким образом. Все, что вам понадобится, это парсер PHP (в виде программы CGI или серверного модуля), веб-сервер и браузер. Чтобы вы могли просматривать результаты выполнения PHP-скриптов в браузере, вам нужен работающий веб-сервер и установленный PHP. В случае, если вы просто экспериментируете, вы вполне можете использовать свой домашний компьютер вместо сервера;

- создание скриптов для выполнения в командной строке. Вы можете создать PHP-скрипт, способный запускаться вне зависимости от вебсервера и браузера. Все, что вам потребуется - парсер PHP. Такой способ использования PHP идеально подходит для скриптов, которые должны выполняться регулярно, например, с помощью cron (на платформах \*nix или Linux) или с помощью планировщика задач (Task Scheduler) на платформах Windows. Эти скрипты также могут быть использованы в задачах простой обработки текстов. За дополнительной информацией обращайтесь к главе Использование PHP в среде командной строки;

- создание оконных приложений, выполняющихся на стороне клиента. Возможно, PHP является не самым лучшим языком для создания подобных приложений, но, если вы очень хорошо знаете PHP и хотели бы использовать некоторые его возможности в своих клиент-приложениях, вы можете использовать PHP-GTK для создания таких приложений. Подобным образом вы можете создавать и кросс-платформенные приложения.

PHP доступен для большинства операционных систем, включая Linux, многие модификации Unix (такие, как HP-UX, Solaris и OpenBSD), Microsoft Windows, Mac OS X, RISC OS, и многих других. Также в PHP включена поддержка большинства современных вебсерверов, таких, как Apache, Microsoft Internet Information Server, Personal Web Server, серверов Netscape и iPlanet, сервера Oreilly Website Pro, Caudium, Xitami, OmniHTTPd и многих других. Для

большинства серверов PHP поставляется в качестве модуля, для других, поддерживающих стандарт CGI, PHP может функционировать в качестве процессора CGI.

PHP способен не только выдавать HTML. Возможности PHP включают формирование изображений, файлов PDF и даже роликов Flash (с использованием libswf и Ming), создаваемых "на лету". PHP также способен выдавать любые текстовые данные, такие, как XHTML и другие XML-файлы. PHP способен осуществлять автоматическую генерацию таких файлов и сохранять их в файловой системе вашего сервера вместо того, чтобы отдавать клиенту, организуя, таким образом, кеш динамического содержания, расположенный на стороне сервера.

Одним из значительных преимуществ PHP является поддержка широкого круга баз данных. Создание скрипта, использующего базы данных, - невероятно просто.

Также в PHP включена поддержка DBX для работы на абстрактном уровне, так что вы можете работать с любой базой данных, использующих DBX. Кроме того, PHP поддерживает ODBC (Open Database Connection standard), таким образом, вы можете работать с любой базой данных, поддерживающей этот всемирно признанный стандарт.

PHP также поддерживает "общение" с другими сервисами с использованием таких протоколов, как LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) и многих других. Кроме того, вы получаете возможность работать с сетевыми сокетами "напрямую". PHP поддерживает стандарт обмена сложными структурами данных WDDX. Обращая внимание на взаимодействие между различными языками, следует упомянуть о поддержке объектов Java и возможности их использования в качестве объектов PHP. Для доступа к удаленным объектам вы можете использовать расширение CORBA.

PHP включает средства обработки текстовой информации, начиная с регулярных выражений Perl или POSIX Extended и заканчивая парсером документов XML. Для парсинга XML в PHP 4 используются стандарты SAX и DOM. Для преобразования документов XML вы можете использовать расширение XSLT. В PHP 5 обработка XML-документов стандартизирована и происходит на базе библиотеки libxml2. Также добавлены два новые расширения для работы с XML: SimpleXML и XMLReader.

В настоящее время PHP поставляется в двух вариантах. Для многих серверов PHP может быть установлен как модуль сервера. Это возможно для таких серверов как Apache, Microsoft Internet Information Server, Netscape и iPlanet. Если PHP не поддерживает интерфейс для вашего сервера, вы всегда можете использовать его как обработчик CGI.

Зачастую генерация динамических страниц, это еще не все, что требуется от веб сервера. Очень часто возникает необходимость сохранять, редактировать и удалять всевозможные данные. В случае, если сложность структуры и объем этих данных невысоки, можно воспользоваться обычными файлами. Однако, если требуется работать с тысячами однотипных данных, то лучше воспользоваться базой данных.

Языки программирования баз данных отличаются от алгоритмических языков, прежде всего решаемыми задачами. База данных – это файл (или групп файлов), представляющий собой упорядоченный набор записей, имеющих единообразную структуру и организованных по единому шаблону (как правило, в табличном виде). База данных может состоять из нескольких таблиц. Удобно хранить в базах данных различные сведения из справочников, картотек, журналов бухгалтерского учета и так далее.

При работе с базами данных чаще всего требуется выполнять следующие операции:

- создание/ модификация свойств/ удаление таблиц в базе данных;
- поиск, отбор, сортировка информации по запросам пользователей;
- добавление новых записей;
- модификация существующих записей;
- удаление существующих записей.

Первые базы данных появились очень давно, как только появилась потребность в обработке больших массивов информации и выборки групп записей по определенным признакам. Для этого был создан структурированный язык запросов SQL (Structured Query Language). Он основан на мощной математической теории и позволяет выполнять эффективную обработку баз данных, манипулируя не отдельными записями, а группами записей.

Для управления БД используется комплекс программ, называемых СУБД, которая обеспечивает создание и хранение БД, управляет доступом к информации, обеспечивает ее пользование (получения требуемых данных) и ведение. Она позволяет создавать таблицы, определять связи между ними, формировать запросы к БД, индексировать поля и т. п. СУБД может также включать средства создания интерфейса БД и генератора отчетов, представляет возможности ведения БД - добавления, модификации и удаления информации.

Свойства полей баз данных СУБД:

- имя поля - определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц);

- тип поля - определяет тип данных, которые могут содержаться в данном поле;
- размер поля - определяет предельную длину (например для символьных данных – в символах) данных, которые могут размещаться в данном поле;
- формат поля - определяет способ форматирования данных в ячейках, принадлежащих полю;
- маска ввода - определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных);
- подпись - определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство "Имя поля");
- значение по умолчанию - то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных);
- условие на значение - ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты);
- сообщение об ошибке - текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности выполняется автоматически, если задано свойство "Условие на значение");
- обязательное поле - свойство, определяющее обязательность заполнения данного поля при наполнении базы;
- пустые строки - свойство, разрешающее ввод пустых строковых данных (от свойства “Обязательное поле” отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например к текстовым).

Базы данных – это тоже файлы, но работа с ними отличается от работы с файлами других типов, создаваемых прочими приложениями.

При работе с обычными приложениями для сохранения данных мы выдаем соответствующую команду, задаем имя файла и доверяемся операционной системе. Если мы закроем файл, не сохранив его, то вся работа по созданию или редактированию файла пропадет безвозвратно.

Базы данных – это особые структуры. Информация, которая в них содержится, очень часто имеет общественную ценность. Нередко с одной и той же базой работают тысячи людей по всей стране. От информации, которая содержится в некоторых базах, может зависеть благополучие множества людей.

Поэтому целостность содержимого не может и не должна зависеть ни от конкретных действий некоего пользователя, забывшего сохранить файл перед выключением компьютера, ни от перебоев в электросети.

Проблема безопасности баз данных решается тем, что в СУБД для сохранения информации используется двойной подход. В части операций, как обычно, участвует операционная система компьютера, но некоторые операции сохранения происходят в обход операционной системы.

Операции изменения структуры базы данных, создания новых таблиц или иных объектов происходят при сохранении файла базы данных. Об этих операциях СУБД предупреждает пользователя. Это, так сказать, глобальные операции. Их никогда не проводят с базой данных, находящейся в коммерческой эксплуатации, - только с ее копией. В этом случае любые сбои в работе вычислительных систем не страшны.

С другой стороны, операции по изменению содержания данных, не затрагивающие структуру базы, максимально автоматизированы и выполняются без предупреждения. Если работая с таблицей данных, мы что-то в ней меняем в составе данных, то изменения сохраняются немедленно и автоматически.

Обычно, решив отказаться от изменений в документе, его просто закрывают без сохранения и вновь открывают предыдущую копию. Этот прием работает почти во всех приложениях, но только не в СУБД. Все изменения, вносимые в таблицы базы, сохраняются на диске без нашего ведома, поэтому попытка закрыть базу “без сохранения” ничего не даст, так как все уже сохранено. Таким образом, редактируя таблицы базы данных, создавая новые записи и удаляя старые, мы как бы работаем с жестким диском напрямую, минуя операционную систему.

Основные объекты базы данных:

- таблицы. Это основные объекты любой базы данных. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы хранят и структуру базы (поля, их типы и свойства);
- запросы. Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразования данных по заданному алгоритму, создавая новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое. Из соображений безопасности, чем меньше доступа к базовым таблицам имеют конечные пользователи, тем лучше. Во-первых, снижается риск того, что неумелыми

действиями они повредят данные в таблицах. Во-вторых, предоставив разным пользователям разные запросы, можно эффективно разграничить их доступ к данным в строгом соответствии с кругом персональных обязанностей.

Термин первичный ключ обозначает поле (столбец) или группу полей таблицы базы данных, значение которого (или комбинация значений которых) используется в качестве уникального идентификатора записи (строки) этой таблицы.

Первичный ключ в таблице является базовым уникальным идентификатором для записей. Значение первичного ключа используется везде, где нужно указать на конкретную запись. На использовании первичных ключей основана организация связей между таблицами реляционной БД. Чтобы организовать между двумя таблицами связь типа «один к одному» или «один ко многим» в одну из связываемых таблиц добавляют поле (поля), содержащее(ие) значение первичного ключа записи в связанной таблице (такое поле называют внешним ключом). Для организации связи типа «многие ко многим» создают отдельную таблицу (так называемую «таблицу связи» или «таблицу ассоциации»), каждая запись которой содержит первичные ключи двух связанных записей в разных таблицах.

Первичный ключ может состоять из единственного поля таблицы, значения которого уникальны для каждой записи. Такой первичный ключ называют простым ключом.

Если таблица не имеет единственного уникального поля, первичный ключ может быть составлен из нескольких полей, совокупность значений которых гарантирует уникальность. Такой первичный ключ называют составным ключом.

Внешним ключом называется поле таблицы, предназначенное для хранения значения первичного ключа другой таблицы с целью организации связи между этими таблицами. Связь между таблицами позволяет:

- либо исключить возможность удаления или изменения данных в ключевом поле главной таблицы, если с этим полем связаны какие-либо поля других таблиц;
- либо сделать так, что при удалении (или изменении) данных в ключевом поле главной таблицы автоматически (и абсолютно корректно) произойдет удаление или изменение соответствующих данных в полях связанных таблиц.

Как правило, в базу данных входит несколько таблиц, которые являются связанными, т. е. каждой записи в одной таблице может соответствовать одна или несколько записей в другой.

Индекс - объект базы данных, создаваемый с целью повышения производительности выполнения запросов. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному значению путем последовательного просмотра таблицы строка за строкой может занимать много времени. Индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и, таким образом, позволяет находить нужную строку по заданному значению. Ускорение работы с использованием индексов достигается в первую очередь за счёт того, что индекс имеет структуру, оптимизированную под поиск - например, балансированного дерева. Кроме того, индексы могут быть объявлены как уникальные и как неуникальные. Уникальный индекс реализует ограничение целостности на таблице, исключая возможность вставки повторяющихся значений.

Существует два типа индексов: кластерные и некластерные. У каждой таблицы может быть только один кластерный индекс и множество некластерных. При наличии кластерного индекса строки таблицы физически хранятся в заданном порядке и непосредственно связаны с элементами индекса, благодаря чему значительно ускоряется доступ к данным при использовании запросов, использующих данный индекс. Если в таблице нет кластерного индекса, таблица является неупорядоченной. Некластерный индекс, созданный для такой таблицы, содержит только указатели на записи таблицы, в связи с чем при выборке необходимо по крайней мере ещё одно обращение к диску для получения собственно записи.

Помимо веб-сервера Apache и PHP при разработке веб сайтов наиболее часто используется СУБД MySQL.

MySQL - это система управления реляционными базами данных. MySQL является очень быстрой, надежной, легкой в использовании и при этом свободно распространяемой (бесплатной) системой управления базами данных. Использование MySQL совместно с Apache и PHP стало особенно популярным по вполне объективным причинам.

MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API).

После того, как браузер запросил у сервера статическую или динамическую страницу, веб-сервер возвращает ему результат в виде HTML страницы.

HTML— это стандартный язык разметки документов во Всемирной паутине. Все веб-страницы создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузером и отображается в виде документа, удобном для человека.

Язык HTML позволяет размечать текст. В том числе вы можете:

- сделать текст жирным, курсивным или подчёркнутым;
- вставить специальные символы (выходящие за рамки ASCII символы пунктуации, математические символы, греческие и готические буквы, стрелки и т. п.);
- поменять гарнитуру, кегль, начертание, цвет шрифта;
- выровнять текст по центру, левому/правому краю, по ширине;
- оформить текст как гиперссылку на другую страницу или файл;
- нарисовать таблицу.

Позже, когда появилась необходимость интерактивности веб-страниц, в HTML появились

- формы для введения пользователем данных, которые позднее подвергаются обработке. Формы и другую информацию можно обрабатывать с помощью специальных серверных программ (например, на языках PHP или Perl);
- открытие мультимедийных файлов, выводимых как непосредственно браузером (например, изображения в форматах JPEG, GIF или PNG; аудиофайлы MIDI и др.), так и внешними приложениями, «встраиваемыми» в окно браузера (Flash-анимация, Java-апплеты и прочее).

HTML — это теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — тегами. Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных (например, тег перевода строки `<br />`). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь атрибуты, определяющие какие-либо их свойства (например, размер шрифта для элемента `font`). Атрибуты указываются в открывающем теге.

В середине 1990-х годов возникло следующее явление. Основные производители браузеров — компании Netscape и Microsoft — начали внедрять собственные наборы элементов в HTML-разметку. Создалась путаница из различных конструкций для работы во Всемирной паутине, доступных для просмотра то в одном, то в другом браузере. Особенно большие трудности были при создании кросс-браузерных программ на языке JavaScript. Веб-мастерам

приходилось создавать несколько вариантов страниц или прибегать к другим ухищрениям.

Сейчас Консорциумом Всемирной паутины (W3C) разрабатывается пятую версию языка HTML5. Черновой вариант спецификации языка появился в Интернете 20 ноября 2007.

Для отделения содержимого документа от его оформления используются каскадные таблицы стилей (CSS). CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления документа. Основной целью разработки CSS являлось разделение содержимого (написанного на HTML или другом языке разметки) и представления документа (написанного на CSS). Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печать, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими Шрифт Брайля.

До появления CSS оформление веб-страниц осуществлялось непосредственно внутри содержимого документа. Однако с появлением CSS стало возможным принципиальное разделение содержания и представления документа. За счёт этого нововведения стало возможным лёгкое применение единого стиля оформления для массы схожих документов, а также быстрое изменение этого оформления.

Преимуществами CSS вёрстки являются:

- несколько дизайнов страницы для разных устройств просмотра. Например, на экране дизайн будет рассчитан на большую ширину, во время печати меню не будет выводиться, а на КПК и сотовом телефоне меню будет следовать за содержимым;
- уменьшение времени загрузки страниц сайта за счет переноса правил представления данных в отдельный CSS-файл. В этом случае браузер загружает только структуру документа и данные, хранимые на странице, а представление этих данных загружается браузером только один раз и кэшируется;
- простота последующего изменения дизайна. Не нужно править каждую страницу, а лишь изменить CSS-файл;
- дополнительные возможности оформления. Например, с помощью CSS-вёрстки можно сделать блок текста, который остальной текст будет

обтекать (например для меню) или сделать так, чтобы меню было всегда видно при скроллинге страницы.

Несмотря на то, что можно сформировать динамический документ на сервере, на стороне клиента возможности менять документ средствами (X)HTML нет. Для изменения документа на стороне клиента используются браузерные языки, самым распространенным из которых является JavaScript. Программа JavaScript выполняется браузером. Для безопасности браузер может запрещать выполнение JavaScript.

Разработанный компанией Netscape, язык был включён в браузер Netscape Navigator начиная со второй версии и первоначально назывался LiveScript. Синтаксис языка брал начало от языков Си и Java, и, поскольку технология Java была в то время очень модной, LiveScript переименовали в JavaScript, получив соответствующую лицензию у Sun. Компания Microsoft, увидев успех JavaScript, создала свою версию этого языка под названием JScript. Прочие производители браузеров также создали свои версии этого языка, что делает задачу написания сложного универсального (совместимого с любым браузером) скрипта довольно трудной.

JavaScript обладает рядом свойств объектно-ориентированного языка, но благодаря прототипированию поддержка объектов в нём отличается от традиционных ОО языков. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам, что придаёт языку дополнительную гибкость.

JavaScript имеет C-подобный синтаксис, но по сравнению с языком Си имеет некоторые:

При помощи JavaScript можно изменять параметры отдельных элементов страницы, менять расположение отдельных элементов относительно друг друга или создавать новые веб-страницы в отдельных окнах и многое другое.

Благодаря легкости в изучении, php является выбором многих начинающих программистов. Однако эта же легкость таит в себе много подводных камней. Многие программы на php представляют собой смесь неструктурированного php кода и html тегов. С увеличением объема кода, такой код становится трудно сопровождать. Разные части системы становятся зависимыми друг от друга и, при необходимости исправлений в одной из частей, приходится переделывать значительную часть кода. Также становится проблематично разделить разработку системы между несколькими программистами.

Для решения вышеописанных проблем используется архитектура Модель-Представление-Контроллер (Model-View-Controller - MVC). Данный подход позволяет разделить работу с объектом на три отдельных компонента: контроллер, модель и вид.

Опишем основные компоненты такой системы:

- Контроллер принимает данные, введенные пользователем, и решает, какие действия необходимо произвести. При необходимости он может передать входные данные модели и инициализировать представление для отображения результата пользователю.

- Модель используется для непосредственного доступа к данным, расчетов и т.д. Например, в качестве модели может выступать класс, который будет выполнять операции создания, редактирования и удаления записей в одной таблице базы данных.

- Представление служит для вывода графического интерфейса приложения пользователю.

Рассмотрим пример работы подобной системы (рисунок 3):

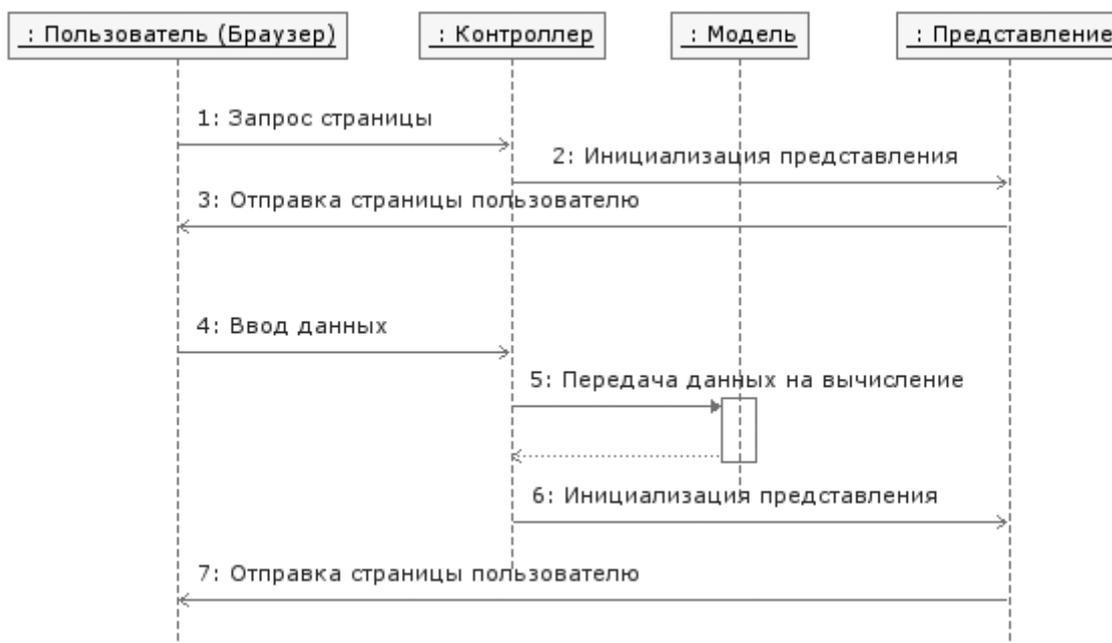


Рисунок 3 – Схема работы MVC-системы

- Пользователь заходит на страницу через браузер. Никакие другие параметры не передаются.

- Контроллер приложения определяет представление по-умолчанию, инициализирует его и передает ему управление.

- Представление по-умолчанию формирует страницу на основе шаблонов и отправляет её пользователю.

- Пользователь посылает данные приложению через форму.

- Контроллер инициализирует модель и передает ей входные данные для осуществления вычислений. Модель осуществляет вычисления и передает результат или код ошибки обратно контроллеру.
- Контроллер на основе входных данных определяет, какое представление необходимо отобразить, инициализирует его и передает ему управление вместе с результатами работы модели.
- Представление формирует страницу на основе шаблонов и поступивших данных и отправляет ее пользователю.

Главным преимуществом подобного подхода является возможность изменения одних частей системы без необходимости изменения других частей или с минимальными изменениями в них. Например, представим, что данные приложения хранились в текстовых файлах и возникла необходимость использовать базу данных. В этом случае достаточно лишь изменить модель данных для работы с базой данных, не трогая логику работы приложения и пользовательский интерфейс. Точно так же изменяя представление можно изменить формат вывода интерфейса с чистого HTML на XHTML или XML, использовать шаблонизаторы или PHP код с HTML для вывода шаблонов.

Другим преимуществом является возможность разделения работы между несколькими людьми. Один программист может работать над моделью, второй – над контроллером, третий – над представлением. А при использовании в представлении шаблонизаторов (например Smarty), можно передать работу над ним верстальщику, который не имеет опыта программирования на PHP.

Из недостатков можно выделить:

- Увеличение объема кода
- Необходимо придерживаться заданного интерфейса
- Для разработки необходимы более квалифицированные специалисты

При разработке веб-приложений разработчики сталкиваются с тем, что от них требуется программирование однотипных действий: разбора входных параметров, создание, редактирование и удаление объектов, работа с базой данных, формирование HTML кода из шаблонов. Для автоматизации подобных рутинных действий используются каркасы приложений или фреймворки.

RНР фреймворки упрощают разработку веб-приложений, написанных на RНР, предоставляя фундамент для построения веб-приложений. Другими словами, RНР фреймворки реализуют концепцию быстрой разработки приложений (Rapid Application Development RAD), что позволяет разработчикам более выгодно использовать свое время, разрабатывать более стабильные приложения и сократить количество повторяющегося кода. Начинаям программистам фреймворки помогут разрабатывать более

стабильные приложения, путем обеспечения корректного взаимодействия с базой данных.

Основной идеей РНР фреймворка является архитектура Модель-Представление-Контроллер. Этот архитектурный паттерн позволяет разделить бизнес-логику, модель представления данных и пользовательский интерфейс на независимые компоненты. Таким образом, становится возможным изменять одни компоненты системы, без изменения других, в результате чего программирование на РНР становится более быстрым и менее сложным.

Главным преимуществом использования РНР фреймворка становится ускорение разработки. Повторное использование кода для похожих процессов помогает сохранить разработчику большое количество времени и усилий. Фреймворки предлагают готовые модули для стандартных задач, так что разработчик может заняться непосредственно разработкой приложения, а не написанием одного и того же кода для каждого нового проекта.

Следующим преимуществом является стабильность приложений, разработанных с помощью фреймворков. Поскольку РНР не сильно ограничивает свободу программиста, становится возможным легко написать плохой код и даже не заметить этого. Такое приложение часто будет работать, но так же будет содержать в себе большую дыру в безопасности. РНР фреймворки содержат методы, которые позволяют значительно сократить количество подобных ошибок.

Существует большое количество различных РНР фреймворков, которые предлагают различные возможности. При выборе фреймворка важно учитывать, кто будет использовать и модифицировать приложение в дальнейшем. При использовании приложения множеством людей, следует выбирать популярные фреймворки, известные большому количеству разработчиков. Если же планируется использовать приложение для личных целей, то разумней будет выбрать фреймворк, который больше подходит требованиям разработчика.

При выборе фреймворка следует обратить внимание на несколько факторов: легкость использования, скорость разработки и работы, популярность среди других разработчиков, наличие документации и поддержки пользователей. Все фреймворки немного отличаются друг от друга и имеют как плюсы, так и минусы.

Рассмотрим проблемы, которые могут возникнуть при использовании фреймворков.

Фреймворки позволяют сократить количество ошибок при разработке программ, так как они предоставляют стандартные функции, которые уже были протестированы многими разработчиками. Однако и при использовании фреймворков не исключены ошибки. Существует множество фреймворков, с

плохой или отсутствующей поддержкой, либо фреймворки, созданные программистами с плохим знанием PHP. Такие фреймворки могут содержать в себе множество ошибок и являться серьезной проблемой безопасности.

Другой из возможных проблем при использовании фреймворков является несовместимость фреймворка с версией PHP или версией баз данных. В случае несоответствия требований фреймворк может функционировать неправильно или не функционировать вовсе.

Следующей проблемой является не следование инструкциям по установке. При неверной установке фреймворк может неверно функционировать, либо может возникнуть проблема в безопасности веб-сервера.

Фреймворк Yii является высокопроизводительным фреймворком, предназначенным для разработки крупных веб-приложений.

Yii фреймворк поставляется в комплекте с полным набором модулей, необходимых для разработки современного веб приложения. Рассмотрим возможности yii фреймворка:

- Архитектура Модель-Представление-Контроллер. Фреймворк использует эту проверенную архитектуру для лучшего разделения бизнес-логики, модели данных и пользовательского интерфейса.

- Объекты доступа к данным (Database Access Objects, DAO). DAO предоставляют общий интерфейс для доступа к базам данных. Таким образом становится возможным смена используемой СУБД, без необходимости изменения кода для доступа к СУБД. Однако DAO обеспечивает совместимость SQL запросов между разными СУБД.

- Шаблоны проектирования Active Record и Relational Active Record. Паттерн Active Record предоставляет удобный объектно-ориентированный интерфейс для работы с реляционными базами данных. В результате программист не пишет напрямую SQL запросы для доступа к БД, а использует методы объекта, реализующего данный шаблон. Основным неудобством при работе с Active Record является то, что данный шаблон представляет только одну таблицу базы данных. Для работы со связанными таблицами в Yii используется Relational Active Record.

- Интеграция с популярным фреймворком jQuery. JQuery в настоящее время является одним из самых популярных фреймворков для программирования на стороне клиента. Он поддерживает простой доступ к элементам страницы, разнообразные манипуляции с элементами DOM, содержит функции для осуществления AJAX запросов, для него существует большое количество расширений.

- Автоматическая проверка ввода пользователя. Проверка пользовательского ввода является одной из самой утомительной частей веб-разработки. Зачастую от разработчика требуется написать большое количество кода только для того, чтобы проверить, является ли введенная пользователем информация верной. В случае ошибки необходимо вернуть пользователя к предыдущей странице, вывести сообщение об ошибке, выделить элемент, в котором введена неверная информация и при этом требуется сохранить всю введенную пользователем информацию в остальных элементах формы. В случае предварительной проверки на клиентской стороне (в браузере), объем кода возрастает в два раза. Yii содержит готовый набор инструментов для автоматизации данного процесса. В нем представлены классы-валидаторы, для проверки наиболее распространенных типов пользовательских данных, дополнительно у программиста есть возможность расширить стандартный набор валидаторов.

- Веб 2.0 виджеты. Yii содержит большой набор элементов пользовательского интерфейса, используемых при проектировании современных Web 2.0 сайтов, таких как календарь, автодополняемые поля ввода, древовидные списки и др.

- Аутентификация и авторизация. Фреймворк имеет встроенную поддержку аутентификации.

- Поддержка тем позволяет мгновенно менять внешний вид приложения.

- Интернационализация (I18N) и локализация (L10N). Поддерживается перевод сообщений, форматирование даты и времени, форматирование чисел и локализация интерфейса.

- Многоуровневое кеширование. В Yii активно используется кеширование для улучшения производительности. Поддерживается кеширование данных, кеширование страниц, кеширование фрагментов и динамическое содержимое. Механизм хранения кеша может быть легко изменен без необходимости изменения кода.

- Обработка ошибок и ведение журналов – Yii обеспечивает вывод подробной информации о произошедших ошибках. Степень подробности можно легко изменить в файле конфигурации. Сообщения в журналах могут быть категоризированы и отфильтрованы.

- Средства обеспечения безопасности. Yii содержит большое количество защитных механизмов, которые помогут предотвратить разнообразные атаки на веб приложение, включая XSS-атаки, CSRF-атаки, подмена cookies и другие.

- Совместимость с XHTML – весь генерируемый код соответствует стандарту XHTML.
- Автоматическая генерация кода. Фреймворк может автоматически генерировать скелет приложения, создавать модели на основе существующей базы данных, создавать контроллеры и представления, поддерживающие стандартные операции создания, редактирования и удаления объектов (CRUD).
- Полная объектно-ориентированность. Код фреймворка разработан с полным соответствием ООП парадигме. Он не содержит глобальных переменных и функций.
- Возможность работы со сторонним кодом. Yii поддерживает работу с кодом третьих лиц. Можно использовать код других фреймворков или дополнительные библиотеки для разработки проекта.
- Детальная документация. Все классы и функции подробно документированы, для обучения новичков предлагается несколько обучающих примеров, кроме того, существует русскоязычное сообщество, которое занимается переводом всей документации на русский язык.
- Библиотека расширений. Фреймворк предлагает библиотеку расширений, состоящую из компонент, написанных пользователями.

### 3.3 Описание системы

Система поддерживает разграничение прав на административные и пользовательские. Доступ к системе осуществляется по имени пользователя и паролю. Каждому факультету присваивается свой пароль. Секретарям деканата доступны функции непосредственно связанные с работой деканата. Администратору же доступны функции общего управления системой.

Рассмотрим список функций, доступных различным видам пользователей.

Пользовательские функции доступные секретарям деканата:

- Ввод данных о людях
- Запись людей в абитуриенты
- Зачисление абитуриентов в студенты
- Приказы на отчисление
- Приказы на повторный курс
- Приказы на перевод
- Приказы на восстановление
- Сохранение приказов в формате rtf

- Просмотр РУПов и создание РУПов
- Просмотр карточек студентов
- Просмотр списков студентов
- Проставление оценок студентам
- Просмотр списков групп
- Создание ведомостей и сводных ведомостей

Администраторские функции для управления системой:

- Управление пользователями
- Редактирование справочников
- Создание типов приказов
- Редактирование шаблонов приказов
- Создание действий приказов

В верхней части страницы располагается меню, для доступа к функциям системы.

### **Меню «Абитуриенты»**

#### 1. Список абитуриентов

На экран выводится таблица с абитуриентами с разбивкой по страницам. Пример списка абитуриентов представлен на рисунке 4. Для каждого абитуриента указывается его ФИО, специальность, на которую он поступает и другие данные. Рядом с именем абитуриента выводятся две кнопки: для редактирования и удаления абитуриента. Для пакетного удаления абитуриентов требуется выделить удаляемых абитуриентов и нажать на кнопку «удалить выделенных». Можно добавить нового абитуриента, нажав на кнопку «Добавить абитуриента».

2	Иванов	Иван	Иванович
3	Петров	Петр	Петрович

Рисунок 4 - Список абитуриентов

#### 2. Добавление абитуриента.

Система предлагает два варианта добавления абитуриента: если добавляемый человек уже обучался в университете, можно выбрать его из

списка людей, если же человек поступает впервые, то нужно ввести информацию об абитуриенте:

- ФИО
- Дату рождения и место рождения
- Пол, национальность, гражданство
- РНН, информацию о документе, удостоверяющем личность
- Контактные данные: адрес проживания, телефон, email
- Место работы, должность
- Социальный статус, семейное положение
- Данные о родителях: ФИО, телефон и место работы
- Фото

Форма ввода информации об абитуриенте представлена на рисунке 5.

Фамилия \*

Имя \*

Отчество

Дата рождения \*

Person Birth Country Id

Казахстан ▼

Место рождения

Павлодар ▼

Пол

Рисунок 5 - Ввод информации об абитуриенте

После заполнения данных об абитуриенте или выбора абитуриента из списка нужно указать данные о будущей учебе. Указывается:

- Специальность, на которую поступает абитуриент
- Форма обучения (полная, сокращенная, второе высшее)
- Тип обучения (очное, заочное, дистанционное)
- Год поступления
- Язык обучения

Далее для каждой из форм обучения указываются дополнительные поля.

Для полной формы обучения:

- Какую школу закончил абитуриент
- Средний балл аттестата
- Номер аттестата
- Год выпуска
- Название школы
- Что сдавал ЕНТ или КТ
- Баллы за ЕНТ или КТ

Для сокращенной формы обучения:

- Какой колледж, техникум (и т.д.) закончил абитуриент
- Средний балл диплома
- Номер диплома
- Год выпуска

Для второго высшего:

- Какой ВУЗ закончил абитуриент
- Средний балл диплома
- Номер диплома
- Год выпуска

### **Меню «Студенты»**

На данной странице выводится список студентов с разбивкой на страницы. Для удобной навигации список можно сортировать по различным полям. Рядом с каждой записью студента находятся ссылки для просмотра карточки студента, для печати карточки студента, для просмотра и редактирования оценок студента, а так же ссылка на просмотр приказов, связанных со студентом. На рисунке 6 показан пример карточки студента



## ИННОВАЦИОННЫЙ ЕВРАЗИЙСКИЙ УНИВЕРСИТЕТ

Факультет ДО

Специальность 050301 Юриспруденция

### УЧЕБНАЯ КАРТОЧКА

1. Фамилия, имя, отчество Абдрахманова Айдана Кабдуллаевна
2. Год и месяц рождения 9.5.1991
3. Место рождения Казахстан, Павлодар
4. Национальность казашка 5. Соц.положение \_\_\_\_\_
6. Что окончил ЭКИИЕУ  
год окончания 2009
7. Изучаемый в школе иностранный язык Английский язык
8. Семейное положение \_\_\_\_\_
9. Адрес местожительства п. Солнечный Ауезова 6 кв. 96  
e-mail \_\_\_\_\_  
Телефон 87052693313
10. Статус студента (с/ст): \_\_\_\_\_

Рисунок 6 – Карточка студента

### Меню «Приказы»

В данном меню отображается список всех типов приказов, доступных для создания. При выборе типа приказа отображается страница со списком созданных приказов. Список разбит на две группы: утвержденные приказы и черновики приказов. Черновики приказов возможно редактировать, утвержденные же приказы редактированию не подлежат. Для каждого из приказов доступна опция экспорта в формат RTF. Так же на странице располагается ссылка для создания нового приказа.

Опишем создание нового приказа на примере приказа «Отчисление в связи с переводом».

На странице создания приказа выводится список полей, необходимых для приказа. Для приказа «Отчисление в связи с переводом» необходимо указать переводимого студента, для этого служит выпадающий список со списком студентов (рисунок 7).

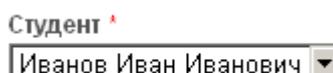


Рисунок 7 – Выбор студента для приказа

Далее указывается наименование учреждения, в которое переводится студент (рисунок 8).

Название учреждения ^

Рисунок 8 – ввод информации об учреждении

И дата заявления студента (рисунок 9).

Дата заявления студента ^

February 2010						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

Рисунок 9 – ввод информации о дате заявления студента

После сохранения приказ попадает в черновики, откуда его можно экспортировать в формат odf. Если приказ был утвержден, то необходимо отметить этот факт, нажав на соответствующую кнопку. В результате выполняются все действия, ассоциированные с приказом, а сам приказ перемещается в группу утвержденных приказов.

### **Меню «Группы»**

#### 1. Список групп.

На странице групп выводится список групп и опции для сортировки и фильтрации. Рядом с каждой записью группы выводятся ссылки для просмотра списка студентов группы, учебного плана группы.

#### 2. Просмотр группы.

При просмотре группы выводится список студентов группы, для каждого из студентов присутствует ссылка на просмотр карточки студента и оценок студента. Так же на этой странице выводятся имя куратора группы, старосты группы и ссылка на просмотр учебных планов группы.

Далее опишем администраторские опции.

Управление пользователями. Для каждого пользователя указывается имя пользователя и пароль, а так же факультет, информацию о котором может редактировать пользователь.

Редактирование справочников. Для администратора доступны несколько справочников, на основе которых заполняется информация о студентах, приказах и т.д. В качестве справочников представлены:

- Страны
- Места жительства
- Факультеты
- Кафедры
- Специальности
- Дисциплины
- Национальности

Создание типов приказов. Данная функция предназначена для автоматизации создания различных видов приказов. Она позволяет редактировать структуру приказа, задавать набор полей, которые требуются для создания приказа. Для облегчения проверки корректности ввода все поля приказа являются типизированными. В качестве типа поля можно указать «один студент» или «список студентов». В результате при создании приказа будет создан список, в котором будет возможно выбрать одного или несколько студентов соответственно.

Для печати приказов используются шаблоны приказов. Шаблоны представляют собой текстовые документ формата RTF. Данный формат может открываться на большинстве современных операционных систем и удобен для редактирования текстовой информации.

Данная система была протестирована в деканате факультета дуального обучения. С помощью нее были распечатаны приказы на зачисление студентов первых курсов, а так же карточки студентов первых курсов. В результате было выявлено, что автоматическая печать документов позволяет значительно повысить эффективность работы деканата.

## **ЗАКЛЮЧЕНИЕ**

В результате написания диссертационной работы была разработана система автоматизации создания, хранения и обработки информации о студентах деканатов и создания приказов из шаблонов на базе открытых технологий разработки. Данная система позволяет секретарям деканата повысить эффективность своей работы. В результате тестирования системы в деканате факультета дуального обучения система показала значительные преимущества подобного подхода по сравнению с ручным созданием и заполнением документации, что доказывает ее эффективность.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Интернет источник: <http://ftad.ru/library/ftad0/37.shtml>;
2. Интернет источник: <http://www.dvgu.ru/meteo/PC/Documents.htm>;
- 3.