

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

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

Кафедра математического и компьютерного моделирования

**Разработка веб-приложения для**

**информационного обеспечения новых сотрудников в организации**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы

направление 38.03.05 — Бизнес-информатика

механико-математического факультета

Гуськова Ильи Александровича

Научный руководитель

к.ф. - м.н, доцент

А.А. Орел

Зав. кафедрой

зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2019

**Введение.** На сегодняшний день информационные технологии развиваются быстрым темпом. Все большее количество людей получают доступ в сеть Интернет. Все больше предприятий используют информационные системы для организации и управления внутренними процессами.

Информационные системы (ИС) - это комбинации аппаратных, программных и телекоммуникационных сетей, которые люди создают и используют для сбора, создания и распространения полезной информации, обычно в условиях организации. В ИС выполняются следующие операции:

- ввод информации из внешних или внутренних источников;
- обработка входящей информации;
- хранение информации для дальнейшего ее использования;
- вывод информации в удобном для пользователя виде.

Основа ИС, объект ее обработки - база данных (БД). База данных - это организованная совокупность данных о какой-либо предметной области. Таким образом БД выполняет функцию хранения информации в ИС.

ИС является средой, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, БД, люди, различного рода технические и программные средства связи и т.д. Хотя сама идея ИС и некоторые принципы их организации возникли задолго до появления компьютеров, однако компьютеризация в десятки и сотни раз повысила эффективность ИС и расширила сферы их применения.

Внедрение информационных технологий в предприятие позволяет значительно облегчить организацию множества внутренних процессов, сбор информации о соискателях, сотрудниках и другом персонале, систематизацию и анализ результатов обучения новых сотрудников. Также внедрение ИТ позволяет реализовывать многие новые формы, например дистанционное обучение. При

этом само понятие дистанционного обучения содержит в себе различные методы, например обучение посредством удаленного тестирования или прочтения электронной литературы.

С течением времени у любого сколько-нибудь большого предприятия по мере его развития возникает потребность в автоматизации, и, как следствие, упрощении, процесса обучения и адаптации новых сотрудников. Специалисты отделов кадров тратят огромное количество времени и сил на внедрение новых сотрудников в рабочий процесс. Соискателям, в свою очередь, необходимо прочесть внушающее количество различного рода литературы, будь то правила пожарной безопасности или основы работы с каким-либо инструментом, а затем расписаться за каждый отдельный документ.

Актуальность данной работы заключается в том, что в настоящее время информационные технологии стремительно развиваются, и на рынке все больше набирают популярность именно веб-приложения. Все больше крупных компаний хотят заменить свои старые сайты современными веб-приложениями. Веб-приложения позволяют автоматизировать многие вещи, упрощая бизнес-процессы компании.

Целями данной работы являются: создание клиентской части веб-приложения, создание серверной части веб-приложения, создание базы данных, хранящей всю информацию, содержащуюся в веб-приложении.

**Задачей** данной бакалаврской работы стало создание информационной системы (веб-приложения) для предприятий, автоматизирующей процесс обучения и интеграции новых сотрудников.

**Основное содержание работы содержит 5 разделов:**

1. Технологии моделирования ИС с использованием UML диаграмм
2. Технологии разработки базы данных.
3. Язык программирования C# и его фреймворк ASP.NET;

4. Язык программирования JavaScript и его фреймворк Angular.
5. Разработка веб приложения для обучения и адаптации новых сотрудников в организациях.

В первом разделе приводится краткое описание основных понятий UML и типы UML диаграмм.

UML (англ. *Unified Modeling Language* — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. Он основан на схематическом представлении компонентов программного обеспечения. Использование визуальных представлений позволяет лучше понять возможные недостатки или ошибки в программном обеспечении или бизнес-процессах.

Типы диаграмм UML:

1. диаграмма вариантов использования;
2. диаграмма классов;
3. диаграмма состояний;
4. диаграмма последовательности;
5. диаграмма коопераций;
6. диаграмма компонентов;
7. диаграмма развертывания.

Далеко не все типы диаграмм используются на регулярной основе при проектировании ИС. В отношении UML диаграмм применим принцип Парето: 20% диаграмм используются разработчиками в 80% случаев. Наиболее часто используемые диаграммы в разработке программного обеспечения: диаграммы вариантов использования, диаграммы классов и диаграммы последовательностей.

Диаграмма вариантов использования является одной из трех самых часто используемых UML диаграмм, в которой проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью прецедентов. Актером (действующим лицом) называется любая сущность взаимодействующая с системой извне. Иначе говоря, каждый вариант использования определяет набор действий, совершаемых системой при взаимодействии с актером.

Еще одной популярной UML диаграммой является диаграмма классов.

Диаграмма класса UML является наиболее распространенным типом диаграммы для документации программного обеспечения. Поскольку большая часть программного обеспечения, создаваемого в настоящее время, все еще основана на парадигме объектно-ориентированного программирования, использование диаграмм классов для документирования программного обеспечения является весьма логичным.

Диаграммы классов содержат классы, их атрибуты и их поведение (методы класса).

Диаграммы классов могут применяться и при прямом проектировании, то есть в процессе разработки новой системы, и при *обратном проектировании* - описании существующих и используемых систем. Информация с диаграммы классов напрямую отображается в исходный код приложения. Таким образом, диаграмма классов - конечный результат проектирования и отправная точка процесса разработки.

Диаграммы последовательности UML - это диаграммы взаимодействия, которые подробно описывают, как выполняются операции. Они фиксируют взаимодействие между объектами в контексте совместной работы. Диаграммы последовательности фокусируются на времени и визуально показывают порядок взаимодействия, используя вертикальную ось диаграммы для представления

времени. Диаграммы последовательностей обычно содержат *объекты*, которые взаимодействуют в рамках сценария, *сообщения*, которыми они обмениваются, и *возвращаемые результаты*, связанные с сообщениями. Впрочем, часто возвращаемые результаты обозначают лишь в том случае, если это не очевидно из контекста.

Во втором разделе приводится краткое описание основных понятий баз данных.

База данных (БД) представляет собой электронную систему, которая позволяет легко получать доступ к данным, манипулировать ими и обновлять их. Другими словами, база данных используется организацией как метод хранения, управления и извлечения информации.

Существует большое количество разновидностей баз данных, отличающихся по различным свойствам и критериям. К основным типам баз данных относятся:

- Иерархическая;
- Сетевая;
- Объектно-ориентированная;
- Реляционная.

Самыми распространенными являются реляционные базы данных. Реляционная база данных состоит из таблиц, которые в свою очередь состоят из строк и столбцов. Содержащиеся в таблицах данные связываются между собой по ключевым значениям.

Для работы с базами данных используются специальные программные средства — системы управления базами данных (СУБД).

Система управления базами данных (СУБД) — это комплекс программных средств, необходимых для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации. Наиболее

распространенными СУБД являются MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

Для оперирования данными в реляционной базе данных с помощью СУБД используется специальный язык SQL.

SQL (structured query language) — в переводе с английского языка структурированных запросов, применяющийся для создания, изменения и удаления данных.

Реляционные базы данных характеризуются наличием некоторых типов таблиц и ключей, позволяющих определить отношения между таблицами. Для того чтобы понять принципы разработки реляционных баз данных, требуется дать определения различных типов реляционных ключей и таблиц:

- Базовая таблица. В реляционной базе данных базовой таблицей называется таблица, которая включает один или несколько столбцов свойств объекта и содержит первичный ключ, который однозначно определяет этот объект. Более того, базовая таблица должна содержать первичный ключ. Базовые таблицы часто называют первичными, поскольку они имеют первичный ключ.
- Промежуточная таблица. Таблица, не являющаяся базовой (т. к. она не объединяет свойства объекта или не содержит поле первичного ключа), которая используется для обеспечения связей между другими таблицами, называется таблицей отношений. Ключевые поля в таблицах отношений должны быть внешними ключами, связанными с первичными ключами базовой таблицы. Проще говоря, таблица отношений состоит только из внешних ключей и не содержит независимых элементов данных.

- Первичный ключ. Первичный ключ состоит из набора значений, которые однозначно определяют запись базовой таблицы. Любому значению первичного ключа должна соответствовать одна и только одна строка таблицы. Первичный ключ включает одно поле только в том случае, если это поле не содержит повторяющихся значений.
- Составные ключи. Если для выполнения условий, накладываемых на значения первичного ключа, заданный ключ включает несколько полей таблицы, то тогда он называется составным.
- Внешние ключи. Внешний ключ — это столбец, значения которого соответствуют значениям первичного ключа другой связанной таблицы.

В третьем разделе содержится краткая информация о языке программирования C# и фреймворке ASP.NET.

C# — это строго типизированный, объектно-ориентированный, простой и в то же время мощный язык программирования, который позволяет разработчикам создавать многофункциональные приложения. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET (программной платформы от компании Microsoft, предназначенной для создания обычных программ и веб-приложений).

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. C# — это фактически гибрид разных языков. Переняв многое от своих предшественников — языков C++, Java, Delphi, Модула и Smalltalk - и опираясь на практику их использования, C# синтаксически не менее чист, чем Java, так же прост, как Visual Basic, и обладает практически той же мощностью и гибкостью, что и C++.



ASP.NET Core является кроссплатформенной, высокопроизводительной средой с открытым исходным кодом для создания современных облачных приложений, подключенных к Интернету.

ASP.NET Core предоставляет следующие преимущества:

- Единое решение для создания пользовательского веб-интерфейса и веб-API.
- Разработано для тестируемости.
- Razor Pages делает создание кодов сценариев для страниц проще и эффективнее.
- Возможность разработки и запуска в ОС Windows, macOS и Linux.
- Открытый исходный код и ориентация на сообщество.
- Интеграция современных клиентских платформ и рабочих процессов разработки.
- Облачная система конфигурации на основе среды.
- Встроенное введение зависимостей.
- Упрощенный высокопроизводительный модульный конвейер HTTP-запросов.
- Возможность размещения в IIS, Nginx, Apache, Docker или в собственном процессе.
- Параллельное управление версиями приложения, ориентированное на .NET Core.
- Инструментарий, упрощающий процесс современной веб-разработки.

Четвертый раздел включает в себя основные понятия о языке программирования JavaScript а также фреймворке Angular.

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

JavaScript — язык сценариев, или скриптов. Скрипт представляет собой программный код — набор инструкций, который не требует предварительной обработки (например, компиляции) перед запуском. Код JavaScript интерпретируется движком браузера во время загрузки веб-страницы. Интерпретатор браузера выполняет построчный анализ, обработку и выполнение исходной программы или запроса.

Также, JavaScript обладает прототипным наследованием. Он поддерживает несколько встроенных объектов, а также позволяет создавать или удалять свои собственные (пользовательские) объекты. Объекты могут наследовать свойства непосредственно друг от друга, образуя цепочку объект-прототип.

Angular представляет собой фреймворк от компании Google для создания клиентских приложений. Прежде всего он нацелен на разработку SPA-решений (Single Page Application), то есть одностраничных приложений. В этом плане Angular является наследником другого фреймворка AngularJS. В то же время Angular это не новая версия AngularJS, а принципиально новый фреймворк.

Angular предоставляет такую функциональность, как двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом, шаблоны, маршрутизация и так далее.

Одной из ключевых особенностей Angular является то, что он использует в качестве языка программирования TypeScript. Но при желании можем писать приложения на Angular с помощью таких языков как Dart или JavaScript. Однако

TypeScript все таки является основным языком для Angular.

TypeScript — язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript.

TypeScript является обратно совместимым с JavaScript и компилируется в последний. Фактически, после компиляции программу на TypeScript можно выполнять в любом современном браузере.

TypeScript отличается от JavaScript возможностью явного статического назначения типов, поддержкой использования полноценных классов (как в традиционных объектно-ориентированных языках), а также поддержкой подключения модулей, что призвано повысить скорость разработки, облегчить читаемость, рефакторинг и повторное использование кода, помочь осуществлять поиск ошибок на этапе разработки и компиляции, и, возможно, ускорить выполнение программ.

Плюсы использования Angular:

- Поддержка Google, Microsoft;
- Инструменты разработчика (CLI);
- Единая структура проекта;
- TypeScript;
- Реактивное программирование с RxJS;
- Единственный фреймворк с Dependency Injection;
- Шаблоны, основанные на расширении HTML;
- Кроссбраузерный Shadow DOM (либо его эмуляция);
- Кроссбраузерная поддержка HTTP, WebSockets, Service Workers;
- Не нужно ничего дополнительно настраивать. Больше никаких оберток;
- Более современный фреймворк, чем AngularJS (на уровне React, Vue);
- Большое сообщество

Пятый раздел посвящен разработке веб-приложения по обучению и адаптации новых сотрудников в организации.

В любой организации при найме нового сотрудника возникает необходимость в максимально быстром обучении и интеграции этого сотрудника в рабочий процесс. Назначать ментора (а это должен быть уже опытный и давно работающий человек) не является практичным, ведь тогда упадет его производительность, что может сказаться на производительности всего отдела. Вследствие этого возникает необходимость в автоматизированной системе, которая позволит новому сотруднику самостоятельно прочитать необходимую литературу и пройти по ней тестирование для контроля за его обучением.

Изначально, новый сотрудник должен зарегистрироваться на платформе, но так-как она не предназначена для открытого доступа в сети Интернет, был настроен SMTP сервер, позволяющий отправлять соискателями письмо на электронную почту, содержащее уникальную ссылку для доступа к

веб-приложению, а также была реализована система личных кабинетов, с помощью которых можно отслеживать прогресс обучения каждого отдельно взятого соискателя.

Также были реализованы модули библиотеки и аттестации. В библиотеке соискатель может найти необходимый материал при помощи поиска по названию или посредством выбора необходимой ему категории, прочитать выбранный материал и отметить его как прочитанный, для отслеживания прогресса обучения. В аттестации же была реализована система автоматизированных тестов, в которой любой сотрудник с соответствующим уровнем прав доступа может создать новый тест и он будет доступен для всех пользователей платформы. После прохождения теста сразу выводится результат и оценка.

Также для ознакомления соискателя с организацией в целом был реализован модуль главной страницы, где присутствует основная информация об организации, новости организации и основные контакты, которые будут динамически генерироваться для каждого нового сотрудника в зависимости от его должности и отдела.

По мере роста функциональности платформы добавлялись новые сущности в базу данных. При помощи фреймворка Entity для языка программирования C# можно добавлять новые сущности и поля в базу данных посредством миграций, что позволяет очень легко масштабировать базу в зависимости от потребностей.

**Заключение.** Веб-приложение - это клиент-сервер программа, клиентская часть которой (включая пользовательский интерфейс и логику) выполняется в веб-браузере.

Основным преимуществом веб-приложений является их мультиплатформенность. Работать с такими приложениями можно абсолютно с любой операционной системы и из любой точки мира (при наличии доступа в Интернет).

Еще одним преимуществом является то, что веб-приложения работают на удаленном сервере и не занимают места на физическом носителе у клиента, а также не требуют больших мощностей устройства для функционирования.

Именно по этой причине веб-приложения являются такими востребованными в настоящее время.

При выполнении данной работы в теоретической части были рассмотрены основные технологии, методы, особенности и составляющие части веб-приложений, язык моделирования бизнес-процессов UML и его диаграммы. Также были рассмотрены клиентский язык программирования JavaScript и серверный язык программирования C#.

В практической части были реализованы следующие аспекты веб-приложения:

- разработана база данных;
- настроен SMTP сервер для отправки электронной почты;
- разработана клиентская часть веб-приложения (пользовательский интерфейс) на языке программирования JavaScript;
- разработана серверная часть веб-приложения на языке программирования C#