**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ**

**КОЛЛЕДЖ ПРЕДПРИНИМАТЕЛЬСТВА №11**

**КУРСОВАЯ РАБОТА**

**На тему: Автоматизация учета кадров на предприятии**

по дисциплине «Внедрение информационных систем»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группы ИСиП-23  Пан Филипп Юльевич  (ФИО)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | **Преподаватель:**  Преподаватель ЦИКТ ГАПОУ КП№11  Макиевский Станислав Евгеньевич  (ФИО)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Москва

2021

Оглавление

[**Введение** 2](#_Toc65571819)

[**1.** **Теоретическая часть** 3](#_Toc65571820)

[**1.1.** **Анализ предметной области** 3](#_Toc65571821)

[1. Анализ предметной области; 3](#_Toc65571822)

[2. Определение технологии разработки приложения; 3](#_Toc65571823)

[3. Создание глоссарий; 3](#_Toc65571824)

[4. Построение функциональной модели предметной области; 3](#_Toc65571825)

[5. Построение модели данных, 3](#_Toc65571826)

[6. Проектирование базы данных, 3](#_Toc65571827)

[7. Проектирование словаря данных; 3](#_Toc65571828)

[8. Проектирование пользовательского интерфейса информационной системы. 3](#_Toc65571829)

[• Построение наиболее эффективной системы, определение критериев и требований для предприятия. 3](#_Toc65571830)

[• Разработка и внедрение технической базы. 3](#_Toc65571831)

[• Введение в систему минимума данных, необходимых для функционирования информационных баз. Начало работы. 3](#_Toc65571832)

[• Планомерное расширение системы автоматизации учета кадров, введение новых личных дел в электронном виде и интеграция данных, уже имеющихся на бумажных носителях. 3](#_Toc65571833)

[• Оперативный контроль и учёт кадров на предприятии; 3](#_Toc65571834)

[• Анализ и обработка входной информации о персонале компании; 3](#_Toc65571835)

[**1.2.** **Выбор технологий и средств разработки** 3](#_Toc65571836)

[**2.** **Практическая часть** 6](#_Toc65571837)

[**2.1.** **Выбор программных средств для реализации** 6](#_Toc65571838)

[**2.2.** **Функциональные возможности приложения** 8](#_Toc65571839)

[**2.3.** **Требования к дизайну** 8](#_Toc65571840)

[**Заключение** 9](#_Toc65571841)

[**Список использованной литературы** 10](#_Toc65571842)

# **Введение**

Ежегодно в системе кадрового делопроизводства происходят изменения, которые требуют все более сложных алгоритмов управления. Ужесточается трудовое законодательство, компании расширяются, возникает необходимость в трудоустройстве все большего количества специалистов. В итоге собственник принимает решение об автоматизации учета кадров, как единственно верное и способное привести деятельность предприятия в соответствие с действующими стандартами, а также снизить количество ошибок и, в конечном итоге, привести к росту доходности бизнеса.

Отдел кадров существует в любой компании, и его работа становится всё сложнее по мере роста предприятия и появления новых подразделений и должностей.

Отдел кадров занимается решением следующих вопросов:

* Работа с персоналом: поиск, наём, увольнение, переводы внутри компании.
* Работа с внутренними документами: отчёты, учёт рабочего времени, составление личных дел сотрудников, выдача справок и больничных листов.
* Расчёт и начисление заработной платы.
* Работа с внешними документами (налоговая документация).
* Приём и обработка заявлений.

При количестве сотрудников до 10 человек учет работников можно осуществлять вручную, однако на предприятиях, где работают 10-100 сотрудников, это будет уже сложно, а если в компании больше 100 человек, то автоматизация учета кадров совершенно необходима.

# **Теоретическая часть**

## **Анализ предметной области**

**Цель курсовой работы:** Разработка системы для учета кадров на предприятии.

**Основные задачи проектирования:**

## 1. Анализ предметной области;

## 2. Определение технологии разработки приложения;

## 3. Создание глоссарий;

## 4. Построение функциональной модели предметной области;

## 5. Построение модели данных,

## 6. Проектирование базы данных,

## 7. Проектирование словаря данных;

## 8. Проектирование пользовательского интерфейса информационной системы.

**Этапы автоматизации учета кадров:**

## Построение наиболее эффективной системы, определение критериев и требований для предприятия.

## Разработка и внедрение технической базы.

## Введение в систему минимума данных, необходимых для функционирования информационных баз. Начало работы.

## Планомерное расширение системы автоматизации учета кадров, введение новых личных дел в электронном виде и интеграция данных, уже имеющихся на бумажных носителях.

**Функции выбранной предметной области ограничены следующими локальными задачами;**

## Оперативный контроль и учёт кадров на предприятии;

## Анализ и обработка входной информации о персонале компании;

## **Выбор технологий и средств разработки**

Средства, которые будут использоваться во время проектирования приложения:

* **Visual Studio**

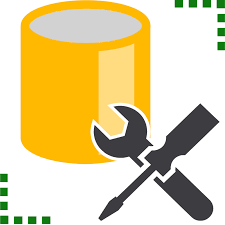


Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки. **Visual Studio** включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Интегрированная среда разработки, ИСP (англ. Integrated development environment — IDE), также единая среда разработки, ЕСР — комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Среда разработки включает в себя:

* текстовый редактор,
* компилятор и/или интерпретатор,
* средства автоматизации сборки,
* отладчик.
* **SQL Server Management Studio**



**SQL Server Management Studio (SSMS) —** это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server, Базы данных SQL Azure и Azure Synapse Analytics, а также управления ими. Среда SSMS предоставляет единую полнофункциональную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом отличных редакторов сценариев для доступа к службе SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней.

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

Плюсы **Microsoft SQL Server:**

* СУБД масштабируется, поэтому работать с ней можно на портативных ПК или мощной мультипроцессорной технике. Процессор может одновременно обрабатывать большой объем запросов.
* Размер страниц – до 8 кб, поэтому данные извлекаются быстро, подробную и сложную информацию хранить удобнее. Система позволяет обрабатывать транзакции в интерактивном режиме, есть динамическая блокировка.
* Рутинные административные задачи автоматизированы: это управление блокировками, памятью, редактура размеров файлов. У системы продуманы настройки, можно создать профили пользователей.
* Реализован поиск по фразам, тексту, словам, можно создавать ключевые индексы.
* В SQL Server есть репликации через интернет, предусмотрена синхронизация. Есть полноценный веб-ассистент для форматирования страниц.
* В систему интегрирован сервер интерактивного анализа для принятия решений, создания корпоративных отчетов. Есть службы преобразования информации.
* Запросы можно формулировать на английском языке, без программирования.

СУБД поддерживает работу с другими продуктами Microsoft: Access, MS Excel.

* **AdobeXD**

Adobe Experience Design (Adobe XD) — программа для разработки интерфейсов от Adobe Systems. Поддерживает векторную графику и веб-верстку и создает небольшие активные прототипы.

* **Microsoft Visio**



Microsoft Visio – программа, с помощью которой появляется возможность для составления графиков, чертежей, диаграмм, блок-схем. Приложение помогает представить графическую информацию в простом и доступном виде.

Это отличный графический редактор, позволяющий работать с диаграммами и схемами. Программа содержит мощный набор инструментов, который будет полезен для работы. Приложение может применяться в разных сферах. Его используют IT-специалисты, менеджеры, аналитики. Программа входит в пакет Office, а также может использоваться в виде отдельного компонента.

Возможности Visio:

**Графическое оформление схем**. С помощью средств Visio можно задать различные эффекты для фигур на схемах процессов, выбрать темы оформления схем, изменять фигуры, сохраняя макеты схем и метаданные фигур.

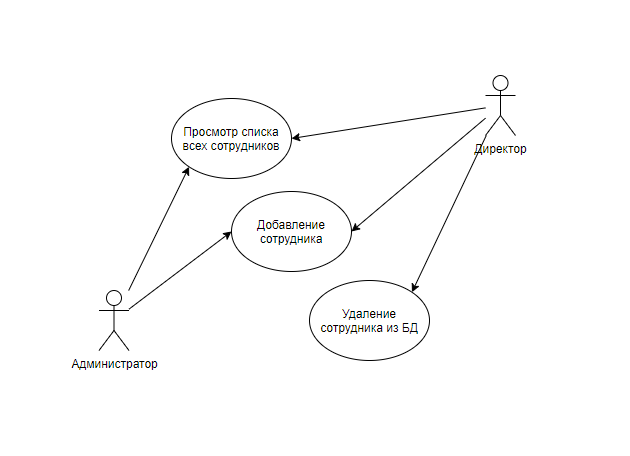
**Совместная работа над схемами**. Используя web браузер можно организовать общий доступ к просмотру схем. При дополнительной установке SharePoint Server и Microsoft Lync 2013 у пользователей появляется возможность комментировать схемы, осуществлять совместную работу с ними и обмениваться сообщения.

**Взаимосвязь схем с наборами данных**. Каждую фигуру из схемы можно связать с набором данных из Excel, SharePoint, службы SharePoint Business Connectivity Services и SQL Server. Для наглядного представления данных можно использовать большое количество графиков и цветовых схем.

**Создание схем с помощью стандартных нотаций**. Для проверки корректности создаваемых схем в Visio встроены правила, позволяющие контролировать правильность применения элементов. Эти правила заданы для стандартных нотаций, таких как BPMN . При необходимости, такие правила можно задавать самостоятельно

## **Проектирование приложения**

**Диаграмма вариантов использования (**[**англ.**](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) **use case diagram)** в [UML](https://ru.wikipedia.org/wiki/UML) — диаграмма, отражающая отношения между [актерами](https://ru.wikipedia.org/wiki/%D0%90%D0%BA%D1%82%D0%BE%D1%80_(UML)) и [прецедентами](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82_(UML)) и являющаяся составной частью **модели прецедентов**, позволяющей описать систему на концептуальном уровне.

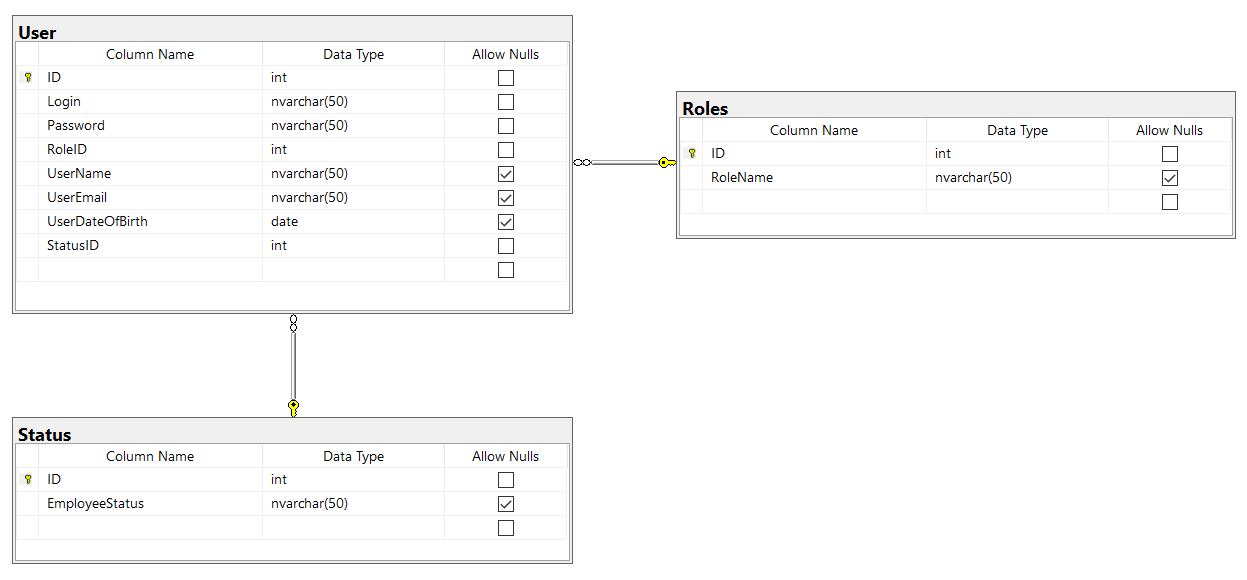


Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

* чётко отделить систему от её окружения;
* определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;
* определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть прецедентов).

Работа над диаграммой может начаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ).



ER-модель (от англ. entity-relationship model, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь» (англ. entity-relationship diagram, ERD, ER-диаграмма).

Понятия «ER-модель» и «ER-диаграмма» часто не различают, хотя для визуализации ER-моделей могут быть использованы и другие графические нотации, либо визуализация может вообще не применяться (например, использоваться текстовое описание).

Таблица User отвечает за наличие пользователей, в ней есть такие поля, как:

* ID – идентификатор сотрудника;
* Login – логин сотрудника;
* Password – пароль сотрудника;
* RoleID – идентификатор роли сотрудника;
* UserName – имя сотрудника;
* UserEmail – фамилия сотрудника;
* UserDateOfBirth – данные о почте сотрудника;
* StatusID – идентификатор статуса сотрудника;

Таблица Roles отвечает за то, под какой ролью будет авторизироваться сотрудника в системе, если речь идет об администраторе или директоре. В таблице описаны все должности. В ней есть следующие поля:

* ID – идентификатор роли;
* RoleName – название роли.

Таблица Status отвечает за то, в каком статусе находится сотрудник на предприятии: на полную ставку, стажер или уволен. В таблице следующие поля:

* ID – идентификатор статуса;
* EmployeeStatus – название статуса.

# **Практическая часть**

## **Выбор программных средств для реализации**

**XCode**

XCode — интегрированная среда разработки (IDE) программного обеспечения для платформ macOS, iOS, watchOS и tvOS, разработанная корпорацией Apple. Первая версия выпущена в 2003 году. Стабильные версии распространяются бесплатно через Mac App Store. Зарегистрированные разработчики также имеют доступ к бета-сборкам через сайт Apple Developer. Xcode включает в себя большую часть документации разработчика от Apple и Interface Builder — приложение, использующееся для создания графических интерфейсов.

Пакет Xcode включает в себя изменённую версию свободного набора компиляторов GNU Compiler Collection и поддерживает языки C, C++, Objective-C, Objective-C++ (англ.) русск., Swift, Java, AppleScript, Python и Ruby с различными моделями программирования. Пакет Xcode использует GDB в качестве back-end’а для своего отладчика.

**Язык программирования Swift**

**Причины использования именно этого языка**

**Причина #1. Гибкость.** Swift отлично работает на операционных системах Apple — iOS и macOS, но не только. Он подходит и для разработки под «операционку» с открытым кодом Linux, а некоторые умельцы даже приспособили его к Android.

**Причина #2. Открытость.** Вы сами можете участвовать в развитии этого языка: в прошлом году Apple открыла его исходный код. И благодаря непосредственному участию сообщества программистов, он каждый день становится функциональнее и стабильнее.

**Причина #3. Простота.** Программный код, написанный на Swift, выглядит очень просто, язык интуитивно понятен каждому программисту, — его доступность и гибкость были ключевыми целями Apple. Кроме того, в сети есть множество курсов для самообучения программированию на Swift: их предлагают Стэнфордский университет, Coursera, Treehouse и другие.

**Причина #4. Функциональность.** Наследуя важные возможности предыдущих «яблочных» языков — C и Objective-C, Swift несёт и много нового функционала. Он быстр, безопасен, гибок и способен помочь умелому разработчику решить практически любую задачу.

**Причина #5. Удобство.** Swift совместим с большим количеством инструментов и систем: если не нравится среда разработки Xcode, можно использовать AppCode или даже любимый текстовый редактор. Кроме того, для Swift сформирована интегрированная экосистема, обеспечивающая совместимость кода, написанного на Objective-C и на Swift, а также тестирование результатов с помощью платформ, подобных Quick. Поэтому разработчик может проверить работоспособность программного кода сразу после его размещения на GitHub.

**Операционная система**

iOS — операционная система корпорации Apple, устанавливаемая ею на мобильные телефоны iPhone 3-13, планшеты iPad, а также на музыкальный плеер iPod.

**Плюсы операционной системы Apple iOS**

**Удобство и стиль интерфейса.**

Разработчиками из Apple был создан действительно простой и легкий в освоении интерфейс. Даже тот человек, который впервые взял “яблочное” устройство в руки, сразу же разберется во всех тонкостях и настройках.

Что касается графической части, то тут все тоже на уровне – присутствуют разнообразные графические эффекты, анимации и прочие украшения. И они, к слову, совсем не нагружают систему. Даже на старых смартфонах (2g, 3g) все работает как часы.

**Отсутствие проблем совместимости.**

Apple iOS является узкоспециализированной операционной системой, функционирующей только на портативных устройствах одноименной компании. Это позволяет разработчикам тщательно проработать и отшлифовать весь функционал, убрав различные проблемы и баги (ошибки). Поэтому, если в списке поддерживаемых тем или иных приложений системы значится определенная версия iOS, это говорит о том, что проблем с запуском у Вас никаких не будет.

**Собственный магазин приложений.**

Множество различных наименований на любой вкус. Игры, текстовые редакторы, напоминалки – список можно продолжать и продолжать. Часть из них являются универсальными. То есть, если есть несколько устройств от Apple, например, iPhone и iPad, то достаточно единожды купить приложение, чтобы пользоваться им с нескольких гаджетов. Разрешение экрана, графические настройки и прочее – все это автоматически подстроится под параметры устройства.

**Высокая автономность.**

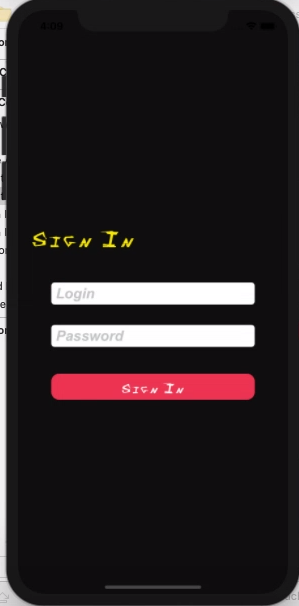
Apple iOS грамотно распоряжается ресурсами батареи, позволяя обеспечить продолжительное время работы без необходимости подзарядки.

**Преимущества для разработчиков:**

* строго ограниченное количество версий смартфона, размеров экрана и самой операционной системы;
* все изменения в ОС прозрачны: Apple ежегодно выпускает гайдлайны по разработке приложений для iOS и объясняет, как заменить элементы кода на более производительные;
* приложений и разработчиков меньше, конкуренция ниже.

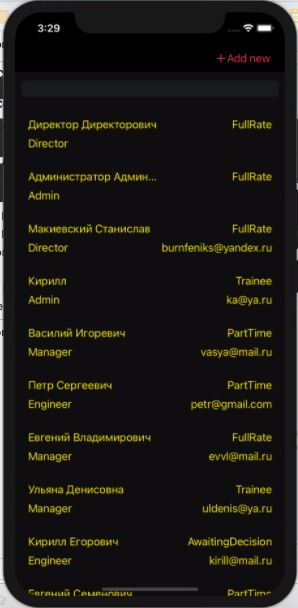
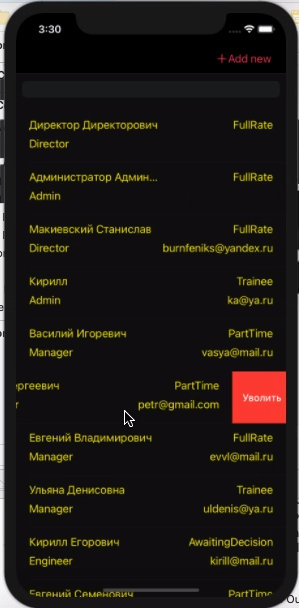
## **Функциональные возможности приложения**

При запуске программы появляется окно авторизации:



1. Страница авторизации

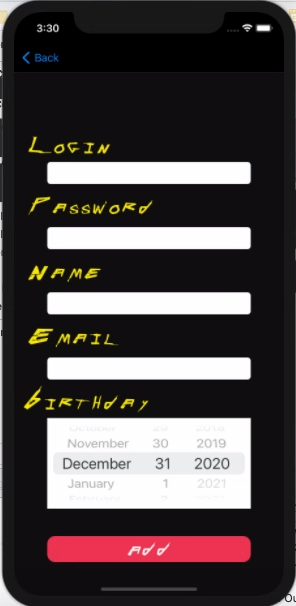
Система разработана для двух основных пользователей – администратора и директора.

2. Страница Администратора и Директора (удаление)

Администратор – сотрудник предприятия, которому доступен список всех остальных сотрудников. Ему доступна возможность добавления новых сотрудников.

Директор – руководитель предприятия, которому помимо просмотра списка сотрудников и добавления доступна возможность удалять данные о сотрудниках из базы данных.



3. Страница добавления нового сотрудника

Ниже представлен программный код окна авторизации и добавления данных в систему. Для реализации представленных методов используются библиотеки: Alamofire и SwiftyJSON.

Alamofire — это HTTP сетевая библиотека на Swift для iOS и Mac OS X. Она обеспечивает элегантный интерфейс сетевого стека Foundation от Apple, упрощающего ряд общих сетевых задач.

Alamofire предоставляет цепочку методов ответ/запрос, параметр JSON и ответ сериализации, аутентификацию и многие другие функции

Alamofire элегантен, потому что написан с нуля на Swift и не наследуюет ничего от AFNetworking, его аналога в Objective-C.

**HTTP** является протоколом приложения, или набором правил, которые используют веб-сайты для передачи данных с веб-сервера на ваш экран. Вы встречали HTTP (или HTTPS) в передней части каждого URL, который вы вводите в веб-браузере. Возможно, вы слышали о других протоколах приложений, таких как FTP, Telnet и SSH. HTTP определяет несколько методов запроса или глаголы, которые клиент (ваш веб-браузер или приложение) использует, для того чтобы указать нужное действие:

***GET***: используется для извлечения данных, таких как веб-страницы, но не изменяет какие-либо данные на сервере.

***HEAD***: идентично ***GET***, но только обратно отправляет заголовки, а не фактическую информацию.

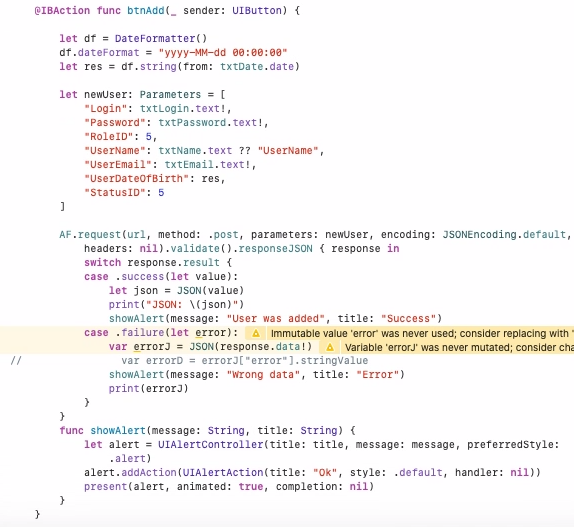
***POST***: используется для отправки данных на сервер, обычно при заполнении формы и нажатии кнопки **submit**(передачи данных).

***PUT***: используется для отправки данных в определенное придусмотренное место.

***DELETE***: удаляет данные из указанного придусмотренного места.



4. Авторизация: код



5. Добавление: код

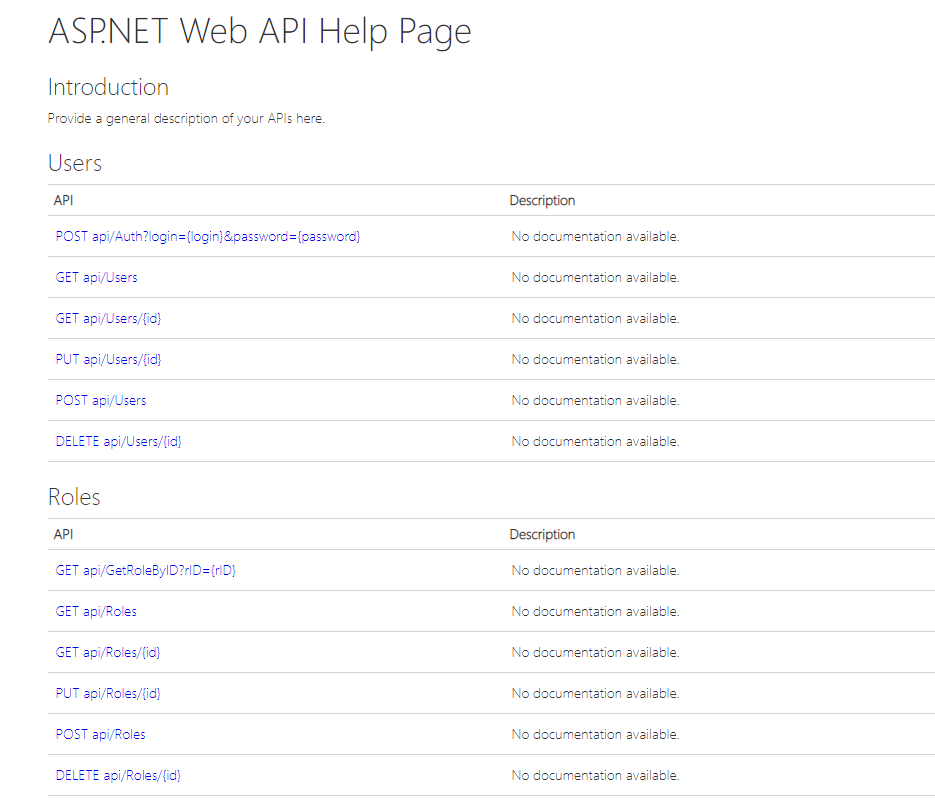
Взаимодействие с базой данных происходит через Web API ASP.NET.

ASP.NET — бесплатная интернет-платформа для создания замечательных веб-сайтов и веб-приложений с помощью HTML, CSS и JavaScript. Также можно создавать веб-API и использовать технологии реального времени, такие как веб-сокеты.

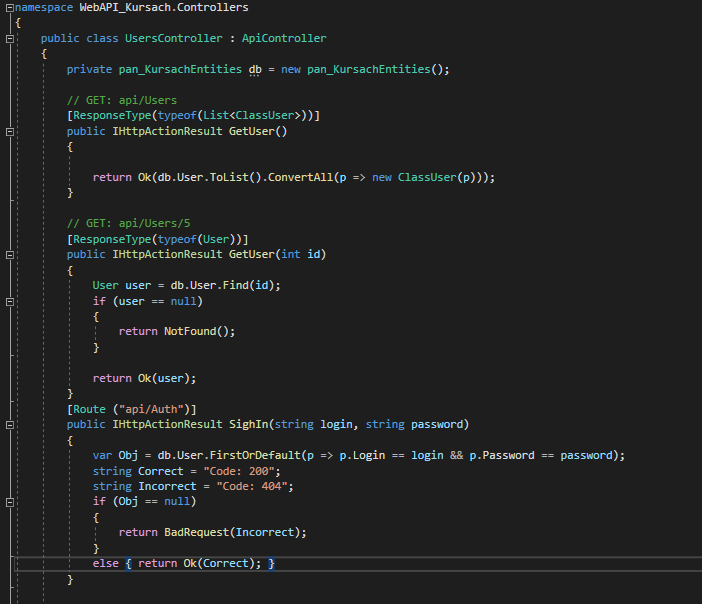
ASP.NET использует схему MVC (Модель-Вид-Контроллер). Каждая часть отвечает за определенные действия. К примеру, пользователь что-то делает на сайте и отправляет запрос на сервер. Контроллер интерпретирует действия пользователя, оповещая модель о необходимости изменений. В свою очередь модель предоставляет данные и реагирует на команды контроллера, изменяя своё состояние. И наконец вид отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Так, в ASP.NET мы могли создавать ASMX-веб-службы. Кроме того, мы можем использовать более функциональную в этом плане технологию WCF для создания веб-служб.

Но с последним релизом MVC 4 и вообще всей платформы .NET 4.5 в нашем распоряжении оказался еще один инструмент для создания веб-служб - Web API. Концепция Web API (которую кстати можно использовать не только в MVC, но и в веб-формах) новый подход к реализации веб-приложений. Интерфейс прикладного программирования (Application Programming Interfaces, APIs) - это готовые конструкции языка программирования, позволяющие разработчику строить сложный функционал с меньшими усилиями. Они "скрывают" более сложный код от программиста, обеспечивая простоту использования.



Программный код Web API для данного проекта был написан на языке программирования C#. Ниже представлены примеры кода основных используемых методов API – GET и POST.



6. Авторизация в API: Код



7. Добавление данных в API: Код

## **Требования к дизайну**

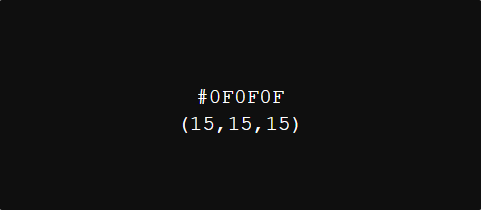
Стилистический справочник, или руководство по стилю, — справочное издание с набором стандартов и обязательных для соблюдения требований при написании и оформлении статей в конкретном издании или при составлении документов в той или иной организации. Служит для поддержания стилистического и оформительского однообразия как для отдельного текстового документа, так и для множества документов.

Применяемые цвета:

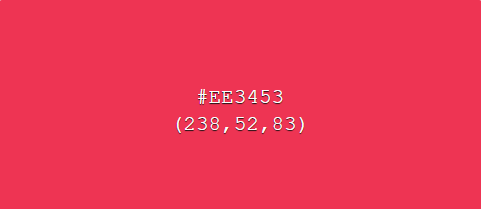
#FFFFFF



#0F0F0F



#EE3453



Дополнительные цвета:

#FAE800



# **Заключение**

В ходе выполнения курсовой работы был выполнен анализ кадрового учета, в результате которого была установлена необходимость разработки автоматизированной системы учета кадров на предприятии.

Результатом курсового проекта стала спроектированная, реализованная и исправно работающая информационная система, которая предоставляет все необходимые данные о составе кадров на текущий момент.

# **Список использованной литературы**

1. <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio#Visual_Studio_2019>, Москва 05.12.2019
2. <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio#Visual_Studio_2019>, Москва 05.12.2019
3. <https://ru.wikipedia.org/wiki/Microsoft_SQL_Server>, Москва 05.12.2019
4. <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>, Москва 10.12.2019
5. <https://ru.wikipedia.org/wiki/Windows_10>, Москва 10.12.2019
6. <https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%BE%D0%B2>, Москва 15.12.2019
7. <https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8>, Москва 15.12.2019
8. <https://www.pinterest.ru/pin/826832812814074574/>, Москва 15.12.2019
9. <https://zen.yandex.ru/media/id/5d33234f31878200acc7007f/pravda-li-tak-horosha-visual-studio-obzor-sredy-razrabotki-5d3d89853f548700b23e2e03>, Москва 15.12.2019
10. <https://www.allware.ru/index.php?id=436>, Москва 15.12.2019
11. <https://iapplenews.ru/2018/04/12/kak-optimizirovat-windows-10/>, Москва 15.12.2019
12. <https://docs.microsoft.com/en-us/sql/ssms/tutorials/connect-query-sql-server?view=sql-server-ver15>, Москва 15.12.2019
13. <https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F>, Москва 18.12.2019
14. <https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8>, Москва 18.12.2019
15. <https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0>, Москва 18.12.2019
16. <https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80>, Москва 23.12.2019
17. <https://ru.wikipedia.org/wiki/%D0%92%D1%80%D0%B0%D1%87>, Москва 23.12.2019
18. <https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D0%BB%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%81%D0%BF%D1%80%D0%B0%D0%B2%D0%BE%D1%87%D0%BD%D0%B8%D0%BA>, Москва 23.12.2019

# **Приложение**

## **Глоссарий**

**Блок-схема** - графическое представление программы или алгоритма с использованием стандартных графических элементов (прямоугольников, ромбиков, трапеций и др.), обозначающих команды, действия, данные и т.п.

**Линейный алгоритм** - алгоритм, все этапы которого выполняются однократно и строго последовательно.

**Язык программирования** - искусственный (формальный) язык, предназначенный для записи алгоритмов. Язык программирования задается своим описанием и реализуется в виде специальной программы: компилятора или интерпретатора.

**База данных** - Один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

**Графический редактор -** Программа или комплекс программ, позволяющих создавать и редактировать изображения на экране компьютера: рисовать линии, раскрашивать области экрана, создавать надписи различными шрифтами, обрабатывать изображения, полученные с помощью сканеров. Некоторые редакторы обеспечивают возможность получения изображений трёхмерных объектов, их сечений и разворотов.

**Инструментальные программные средства -** Программы, используемые в ходе разработки, корректировки или развития других программ: редакторы, отладчики, вспомогательные системные программы, графические пакеты и др. По назначению близки системам программирования.

**Интерфейс -** Электронная схема сопряжения двух устройств, обменивающихся информацией.

**Информационно-поисковая система (ИПС) -** Система, выполняющая функции хранения большого объёма информации, быстрого поиска требуемой информации, добавления, удаления и изменения хранимой информации, вывода её в удобном для человека виде.

**Логическое высказывание -** Любое предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

**Операционная система -** Комплекс взаимосвязанных программ, предназначенных для автоматизации планирования и организации процесса обработки программ, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программ, других вспомогательных операций обслуживания. Важнейшая часть программного обеспечения.

**Описание** - Раздел программы, идентифицирующий структуры данных, которыми должна манипулировать программа, и описывающий их типы.