1. **Обоснование темы**
   1. Постановка задачи автоматизации контроля качества знаний

Целью квалификационной работы является разработка клиентского приложения для системы контроля качества обучения в *IT*-компании, предоставляющего пользователю всю необходимую информацию по направлениям обучения.

Проект должен соответствовать следующим требованиям:

1. выполнение поставленных задач;
2. удобство пользовательского интерфейса;
3. обработка информации для контроля обучения и усвоения материала;
4. подробная документация.

Клиентское приложение должно реализовывать следующие механизмы:

* настройка соединения с сервером;
* настройка соединения с частью базы данных, не реализованной на сервере;
* загрузка и обновление cookies;
* авторизация пользователей в зависимости от введенного логина и пароля;
* разграничение доступного функционала;
* загрузка и отображение данных о направлениях подготовки, проводимых компанией;
* загрузка и отображение данных о персонах и кандидатах на курсы;
* загрузка и отображение данных о ресурсных менеджерах, которые проводят занятия;
* загрузка и отображение данных о реквизициях по каждому направлению подготовки;
* загрузка и отображение данных о посещаемости и успеваемости студентов на текущих курсах по направлениям;
* анализ прохождения студентами курсов, на основе которого строится диаграмма среднего отставания по темам.
  1. Технико-экономическое обоснование

При проведении каких-либо курсов важно понимать, как лучше построить учебную программу. Это является одним из решающих факторов успешного освоения материала. Правильная последовательность тем позволяется качественнее понять суть представленного материала и запомнить его. Анализируя предоставленное на каждую из тем время и сопоставляя его с общим доступным на курсы, можно корректно построить учебный план.

Рассмотрим основные причины, по которым студенты могут плохо понять программу курсов, и которые реализует представленное приложение.

*Выбор способа разделения и предоставления информации ресурсному менеджеру.* Правильное разделение актуальной и необходимой информации является важным фактором быстрого построения учебного плана и разграничения доступного функционала. Также это помогает избежать ошибок среди данных, обусловленных человеческим фактором.

*Отсутствие ссылочной целостности*. К этому недостатку можно отнести *unfriendly user-interface* («недружелюбный пользовательский интерфейс») и долгий переход между страницами приложения. В случае удобного доступа к ссылкам на другие вкладки происходит экономия временных затрат и отсутствие нагромождения данных.

*Корректное использование предоставляемой информации*. Правильно используемые данные – залог успешной работы. Вся получаемая информация должна быть верно проанализирована. Такие данные обладают большей информативностью и ценностью.

*Сильная зависимость клиентской части от базы данных*. Данный недостаток позволяет развернуть приложение целиком только на одном компьютере. В случае переполнения памяти подтягивание данных из базы и отображение их на клиенте значительно увеличивает временные затраты на загрузку и корректное отображение данных, а также на их анализ.

*Анализ данных в режиме реального времени*. Тщательный анализ предоставляемой информации не имеет особой ценности, если такая информация не актуальна. Своевременный обзор материала несет большое значение для любой системы.

В дальнейшем планируется перспектива развития проектируемого программного продукта:

* расширение пользовательского интерфейса относительно увеличению данных, посылаемых на сервер;
* более тщательно разделение получаемой информации, а также ее структуризация в зависимости от объема;
* добавление новых критериев для анализа.
  1. Выбор программных средств для решения поставленных задач

Разрабатываемая программа является клиентской частью веб-приложения. Поэтому основной выбор программных средств направлен на языки программирования и утилиты, предназначенные для верстки и прорисовки пользовательского интерфейса, а также для обработки данных на клиенте.

Средой программирования был выбран Eclipse IDE for Java EE Developers версии Neon 1.0. Данная среда является одной из последних версий Eclipse, разработанных компанией ORACLE. Она была выбрана, благодаря ее преимуществам:

* обеспечение согласованного набора возможностей для большинства платформ;
* поддержка не только Java, но и других высокоуровневых языков программирования;
* свободно распространяемая и с открытым кодом, но с полной поддержкой;
* имеет промышленный уровень;
* расширяемый и конфигурируемый.

Для работы на серверном уровне был выбран контейнер сервлетов Apache Tomcat v7.0. Веб-сервер Apache выбран в проектируемом ВКР из-за его гибкости, мощности и широкой распространенности. Он может быть расширен с помощью системы динамически загружаемых модулей и исполнять программы на большом количестве интерпретируемых языков программирования без использования внешнего программного обеспечения.

При разработке рассматриваемого приложения основной акцент был обращен на прорисовку пользовательского интерфейса. В качестве языка разметки был выбран HTML, а для его отображения – язык стилей CSS.

HTML – это самый распространенный инструмент создания сайтов. Он был выбран по следующим причинам:

* оптимально поддерживается и интерпретируется интернет-проводниками;
* читается на различных платформах;
* считается открытым стандартом.

Использование CSS обусловлено тем, что стили являются удобным и практичным инструментом при верстке веб-страниц и оформления различных элементов. Рассмотрим преимущества CSS, которые были использованы в разработываемом приложении:

1. *Разграничение кода и оформления*.

Идея состоит в том, чтобы код HTML был свободен от элементов оформления вроде установки цвета, размера шрифта и других параметров.

1. *Расширенные способы оформления элементов.*

Стили имеют гораздо больше возможностей по оформлению веб-страниц.

1. *Единое стилевое оформление различных страниц*.

Сайт это не просто набор связанных между собой документов, но и одинаковое расположение основных блоков, и их вид. Применение единообразного оформления заголовков, основного текста и других элементов создает преемственность между страницами и облегчает пользователям работу с сайтом и его восприятие в целом.

1. *Централизованное хранение*.

Стили, как правило, хранятся в одном или нескольких специальных файлах, ссылка на которые указывается во всех документах сайта. Благодаря этому удобно править стиль в одном месте, при этом оформление элементов автоматически меняется на всех страницах, которые связаны с указанным файлом.

JQuery представляет собой кроссплатформенную библиотеку. В представленной ВКР она использует для связи JavaScript и HTML. При помощи JQuery код пишется намного легче и быстрее, чем просто на JavaScript. Это эффективно с точки зрения экономии времени. К тому же, проект, разработанный в JQuery, будет работать в любых современных браузерах. Все версии JQuery совместимы между собой. А в стандартной библиотеке существует огромное количество компонентов, которые можно использовать в своих проектах.