Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технологический университет»

(ФГБОУ ВО «КНИТУ»)

КУРСОВОЙ ПРОЕКТ

по дисциплине «Методы и средства проектирования информационных систем и технологий»

Тема: Обучение на основе виртуальной реальности сборке разборке изделий машиностроения

**Выполнил:**

студент группы 7201-11

Туйчиев Равшан Рустамович

**Проверил:**

зав. каф. ИПМ, доцент

Старыгина С.Д.

**Оценка:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Казань, 2023 г

Содержание

Введение…………………………………………………………….....3-4

1. Анализ предметной области, цели и постановка задачи……………..5
   1. Формирование общих требований……………………………...6
2. Фаза исследования и уточнения…………………………………….....7
   1. Концептуальная модель на DFD…………………………...…8-9
   2. Декомпозиция концептуальной модели на IDEF-3……..….9-14
   3. Рабочая (контекстная) модель …………………………….…..15
3. Выбор средств разработки………………………………………........16
4. Архитектура ПО………………………………………………...…17-18
5. Требования к программному продукту ………………………….19-20
6. Фаза тестирования…………………………………………………21-23

Заключение………………………………………………………….…24

Список литературы……………………………………………………25

Приложения (программный код)……………………………...….26-34

**Введение**

В современных реалиях схемы всемирной сети прогрессируют с каждым днем. Ежегодно появляются новые, более усовершенствованные, версии программ и продукты. Если раньше не каждая организация могла себе позволить личный сайт, то уже сейчас даже небольшой бизнес и индивидуальные предприниматели обладают не одним собственным веб-сайтом для демонстрации своих товаров и услуг. Если говорить о крупных организациях, то там уже находится несколько десятков, а то и сотен страниц, объединенная база данных, поисковая система и т.д.

Для того, чтобы заявить о своем существовании, не нужно создавать большой и объемно-информационный сайт. Изначально можно призывать клиентов с помощью, так называемых “Сайтов-однодневок” или “Сайтов-визиток”, где будет указана минимальная, но полезная информация для пользователя. Как правило такие сайты не обладают большим количеством страниц. Нужно указать деятельность, какие услуги предоставляет организация, график работы, а также будет значительным плюсом колонки новостей, чтобы всегда оставаться в курсе актуальных событий. Обязательным пункт на любом сайте – это контактные данные и местоположение.

Нынешние технологии позволяют не только создавать сайты, но и делать их индивидуальными, например, добавив динамику, что значительно увеличит посещаемость и тем самым поднимая рейтинг, но стоит учитывать, если сайт имеет множество сложных объектов, то время на создание сайта, как и его стоимость, конечно же будет увеличиваться соответственно.

Не смотря на все плюсы сайтов, процесс их создания очень трудоемок, и порой требует даже не одного разработчика, поэтому следует провести точный анализ интернет ресурсов конкурентов, выявить их недостатки и сильные стороны, чтобы не повторять ошибок и сделать максимально

успешный сайт. Перед созданием, также нужно определиться с макетом сайта, как он будет выглядеть, ведь даже малейшие правки в систему отсрочат выпуск готово продукта. Важно определиться с типом сайта, ведь не всегда “Сайт-визитка” выигрышный вариант, в определенных случаях нужно выбрать бизнес сайт или страницу интернет-магазина. Всё зависит от вида деятельности.

Объектом исследования будет высшее учебное заведение “Казанский Национальный Исследовательский Технологический Университет”.

Работа над созданием информационной системы начинается с того, что нужно точно знать и учитывать требования заказчика к создаваемому сайту в виде спецификации (тз). Техническое задание (тз) – является основным документом, определяющим условия и требования при создании.

При разработке сайта нужно учитывать, что на главных страницах действительно будет вся важная информация, просмотр страниц будет грамотно смотреться с любого устройства и в дальнейшем не будет проблем с поддержкой этого сайта.

Т.к. сайт по большей части предназначен для абитуриентов, то необходимо предоставить всю нужную информацию на видном и удобном месте, чтобы поиск не занимал большое количество времени.

1. **Анализ предметной области, цели и постановка задачи**

Целью данной работы является создание программного продукта, позволяющий в виртуальной среде взаимодействовать с объектами машиностроения. Заказчиками и в том числе конечными пользователями реализуемого программного обеспечения являются производственный предприятия и обучающие организации. Деятельность таких компаний должна быть связана с производством, конструированием, сборкой или обслуживанием сложного оборудования, требующая высокую квалификацию для данных типов работ.

Обучающий процесс в подобного рода компаниях, согласно полученными мной информацией, организован таким образом, что обучающиеся проектируют, конструируют, агрегируют и разбирают сложное оборудование. Данные процессы, в ходе своего выполнения влекут за собой издержки: затраты физических ресурсов, огромные затраты времени, а также, что очень важно и нельзя оставить без внимания риск для здоровья.

Издержки физических ресурсов заключаются в том, что при вышеописанных процессах, все действо происходит с реальными физическими объектами, на создание которых требуются много сил, времени и природных ресурсов. Так если какой-то студент, при не удачной сборке каким-либо образом повредит деталь, то дальнейшее её использование будет маловероятно. При этом во время обучения данное событие априори неизбежно. В конечно итоге это выливается в огромные счета для компании, требующие оплату за поврежденный инвентарь и детали.

Затраты времени, на первый взгляд, не такие очевидные, как может показаться. Однако целевая деятельность требует, из-за фактора обучения, долгосрочное, с точки зрения сессии многочасового вовлечения студента, внимание к конкретному процессу. Можно заметить, что для ускорения обучения, компании проводят курсы, занятия в группах, что требует отдельных затрат на организацию и подготовку конкретного экземпляра продукта к работе.

Риск для здоровья уже более очевидный и понятный фактор издержек. В данной области он проявляет себя в возможностях обучающимися получить травмы во время сборки, либо работы по обслуживанию конкретных изделий машиностроения. Так, например, поставив деталь в неправильную позицию, при запуске оборудования может произойти, либо разлет составных частей и их деформация. Что приводит как минимум с лишним затрат ресурсов, а максимум может повлечь тяжкий вред жизни человека, если даже не летальны исход.

Исходя из вышеописанных издержек обучения, можно выделить бросающиеся в глаза проблемы компаний, требующие решения.

Проблемы:

* Снизить издержки процесса обучения
* Повысить безопасность целевых процессов
* Повысить эффективность обучения

Проблема повышение эффективности обучения мною проглядывается в том, что единоличный процесс обучения по сравнению с групповым является более эффективным. Сразу стоит отметить, что этот тезис не применим ко всему процессу обучения, а лишь только к конкретному. А именно ситуация, при которой один обучающийся работает с оборудованием, в то время как другие лишь тихо и смирно стоят в стороне и наблюдают, по понятной причине – оборудование в одном экземпляре, из-за своей дороговизны.

Возвращаясь к программному продукту, можно увидеть в нем инструмент для решения приведенных проблем. Начиная с наиболее важного – безопасность жизни и деятельности. Виртуальная среда имеет максимальный уровень безопасности, беря во внимание незначительную нагрузку на зрительную систему и другие последствия, сопровождаемые работой с электронными устройствами, работа производится с цифровым двойником, а не физическим оригиналом и чисто логически понятно, что никакой физический тяжкий вред в таких условиях не может быть получен.

Передвигаясь от более значительных к менее значительным проблемам, рассматриваем решение снижения издержек. Виртуальная среда, не требует огромных физических ресурсов, для представления свойств реальных объектов т.е. не требуется физический экземпляр, имеющий ощутимую стоимость. Цифровой двойник не может быть поврежден, по очевидным причинам, что снижает издержки ресурсов. Не придется закупать множество деталей и оборудований по несколько раз, потому что единожды созданный качественный цифровой двойник, будет передавать основные свойства реального объекта и давать бесконечную возможность на создании копий и многократного и одновременного использования.

Одновременное использования решает следующую проблему: повышения эффективности обучения. Стажеры и студенты смогут, благодаря виртуальной среде, параллельно производить манипуляции каждый со своим цифровым двойником, а не быть зрителем в стороне. Ведь один из важнейших принципов в обучении это практика. Так каждый обучающийся, своими руками, вместе со всеми будет выполнять целевые задачи, а не ожидать в очереди к дорогостоящему оборудованию.

Имея достаточно полно описанные проблемы, можно поставить задачи проекта.

Задачи:

* Моделирование архитектуры программного обеспечения
* Разработка MVP ПО удовлетворяющего нашим ожиданиям
* Тестирование и пилотный ввод в эксплуатацию

MVP – минимально жизнеспособный продукт (eng. Minimum Viable product), продукт обладающий минимальными, но достаточными для удовлетворения первых потребителей функциями. Основной задачей которого является сбор обратной связи для дальнейшего формирования гипотез и уточнений требований и задач дальнейшего развития продукта.

* 1. **Формирование требований**

Требования к программному продукту.

Поскольку любой программный продукт является системой, а подчас и даже сложной суперсистемой, то рассматривать его и описывать требования нужно с точки зрения систем. А наименьшей, функциональной, неделимой частью системы является элемент. Таким образом требования к системе представлены в виде элементов системы, которые должны быть в нее включены.

Требования, элементы системы.

* Передвижение пользователя в пространстве
* Цифровой двойник реального объекта
* Взаимодействие пользователя и цифровых двойников
* Чертеж собираемой модели
* Логика отображения чертежей
* Создание цифровых двойников в виртуальной среде
* Окружающее виртуальное пространство

1. **Фаза исследования и уточнения**

В фазе исследования будут представлены:

* IDEF3 модель (каждый блок по отдельности, для более подробного и точного объяснения)
* Описание элементов системы и их свойств
  1. **IDEF3 модель и разбор на составляющие**

IDEF3 нотация, она же методология моделирования и общепризнанный стандарт предназначенный для документирования процессов, происходящих в системе. Данная нотация описывает связи – причины и следствия, между ситуациями и событиями, являющиеся подсистемами, в доступном для профессионалов представлении.

Описание этапов в IDEF3 происходит графическим образом. Показываются взаимосвязи функций и объектов друг с другом, а также обработка информации. Данный стандарт отличный инструмент для аналитики, поскольку дает возможность детально описать отдельные этапы или сущности, которые принимают участие в одном и том же процессе.

* 1. **Декомпозиция модели проектирования на IDEF-3**

Каждый отдельный этап в IDEF-3 модели содержит задания, отношения и пересечения.

Первый блок включает в себя несколько задач, которые требуется решить, прежде чем приступать к следующем этапу, со схемой можно ознакомиться на рис.1. Ниже приведен список задач данного этапа с описанием блоков, идущих по порядку, согласно логики процесса разработки ПО.

Получение задания – начальный этап при любом проектировании, без которого чисто физически и логически не будет возможности для дальнейших действий. На данном этапе узнаётся желаем результат конечного продукта, какие свойства он должен иметь, какие технологии и ресурсы должны быть использованы в ходе реализации проекта. Так же формулируются проблемы, которые должен решать конечный продукт. На основании проблем формируются задачи, которые должны быть решены с соблюдением заявленных требований.

Описание функционала – прежде чем приступать к самому процессу разработки: написание программного кода, создание макетов и т.д., требуется четко понимать, какие возможности нужно будет реализовывать. Потому что, когда задача описана по пунктам, с указанием свойств и желаемого результата, дальнейшая разработка становится многократно качественнее и быстрее. Это обусловлено физиологией мозга человека, который способен долго удерживать фокус внимания лишь на конечном, представимом списке образов и задач, требующих решения

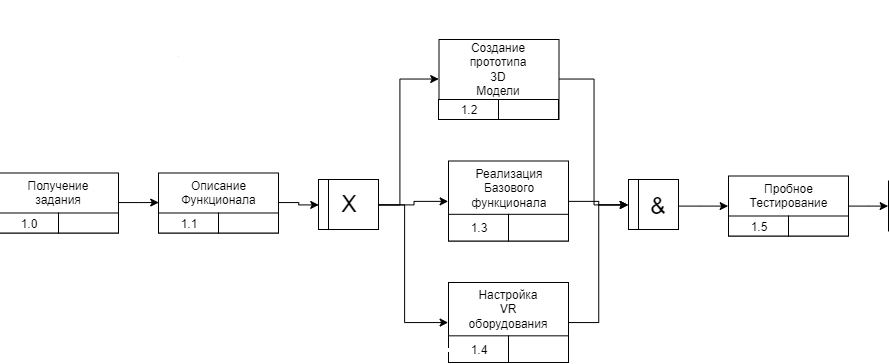
Следующие этапы происходят параллельно, т.е. можно приступить к любому из них или начать сразу все одновременно, поскольку это дает гибкость при работе. Так как этапы не сильно связаны друг с другом, то можно использовать максимум возможностей от каждого момента времени разработки.

Создание прототипа 3D модели – на данном этапе выбирается первая модель, которая будет цифровым двойником на этапе проектирования. Для повышения качества и эффективности конечного продукта, на данном этапе эта модель должна быть среднего уровня детализованности. То есть состоять не из малого количества составных деталей будь то три или четыре части и не из огромного множества, на создание и настройку которых уйдет несколько месяцев работы. Дюжина деталей является предельно достаточным для данного этапа. Поскольку этого количества хватит для описания функционала, который впоследствии может быть экстраполирован на другие аналогичные компоненты.

Настройка VR оборудования – необходимый этап, реализация которого дает возможность разработки компонентов, предназначенных для интегрирования VR контроллеров со средой разработки. В дальнейшем это даст возможность, производить все действия в виртуальной среде при помощи обеих рук и головы используя все 3 степени свободы пространства. Что по сравнению с обычными клавиатурой и мышью является более эффективным и качественным вариантом.

К следующему этапу можно приступить лишь только при выполнение всех трех предшествующих этапов разработки, поскольку интегрирование результатов этих трех этапов, даст более полное начальное представление о конечном продукте.

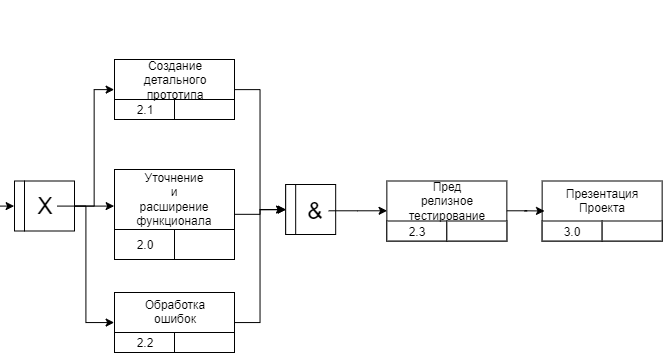
Пробное тестирование – этап без которого невозможно проводить разработку программного продукта. На данном этапе мы пытаемся «сломать» приложение, используя лишь те возможности, которые были даны пользователю. Таким образом находятся ошибки, которые мог встретить пользователь во время реальной работы с программой и получить негативный опыт, вследствие которого мог навсегда отказаться от использования данного приложения.



**Рис. 1. Первый блок**

* Второй блок – ознакомление с макетом сайта. Здесь нужно учитывать, что оба требования обязательны для выполнения. Просмотр макета – это ознакомление с внешним видом, как будет показан итоговый продукт. Работа этого блока продемонстрирована на рис. 2

Ознакомление с требованиями – учет современных требований для создания, так называемый Code Style.



**Рис. 2. Второй блок**

Второй блок – это следующий цикл по спиральной модели разработки, при котором уточняются, расширяются возможности, свойства и функции описанные в предыдущем цикле разработки. Также, как и на предыдущем этапе важно учитывать каждое требование. Работа этого блока продемонстрирована на рис. 2.

На данном этапе имеется возможность осуществлять разработку сразу по нескольким направлениям, поскольку предыдущие этапу выполнены и проект позволяет. Сам действия могут быть делегированы среди разных специалистов, это отличное подспорье для дальнейшего коммерческого развития продукта, поскольку командная работа дает огромные бонусы при работе над проект, учитывая, что модель проект позволяет параллельное исполнение множества задач.

Важное требование, которое должно соблюдаться на протяжении всего проекта это стиль кода, он же чистый код. Поскольку C# язык используется для описание сложных функций, то чистый код - это необходимый компонент, который позволит в дальнейшем поддерживать разработанный продукт. Ибо, написав нечто понятное сейчас, при взгляде на тот же самой программный код спустя даже месяц, может потребоваться колоссальное количество времени чтобы просто вникнуть в суть происходящего, не говоря уже об изменении функционала.

Создание детального прототипа – на этом этапе предыдущий базовый прототип дорабатывается, уточняются детали конкретного объекта. Если до этого прототип первого цикла, представлял собой простые фигуры будь то куб, сфера или цилиндр, то на данном этапе геометрия деталей и конечного объекта в целом становится узнаваемой. Так узкий цилиндр, приобретает форму колеса с резиной и ободом. Взглянув на который, любой дееспособный человек ответственно сможет заявить, что это колесо. И так абсолютно со всеми деталями. Так же на этой фазе создаются материалы, которые будут покрывать поверхность каждой детали, что будет повышать эффективность интуитивного определения сущности каждого конкретного объекта.

Уточнение и расширение функционала – если до этого у нас были базовые функции, дающие возможность перемещаться в пространстве, вращать камерой, и передвигать объекты, а сам чертеж лишь статично без всякой реакции на действия пользователя находился в пространстве. То теперь на этом этапе добавиться реакции чертежа на действия пользователя, так что он сможет при перемещении детали в пространстве той же детали на чертеже установить её и это будет сопровождаться визуальными изменениями.

Обработка ошибок – безусловный компонент, который невозможно пропустить. На предыдущем этапе было обнаружено множество ошибок, так называемых багов. Когда ожидалось одно поведение от сущности, а по факту происходит совершенно другое. Одной из краеугольных ошибок, которое могло поставить под вопрос возможность реализации проекта была оптимизация. На одной из стадий разработки FPS(Frames Per Second eng. Кадров в секунду. Общепризнанный параметр, указывающий на эффективность использования ресурсов компьютера приложением) в демонстрационном режим падал до 10-20 кадров в секунду, что является недопустимым. Множество возможных причин появления этой и проблемы и пути их возможных решений были обработаны, что внесло существенную долю издержек времени.

Только когда все эти три этапа выполнены, мы можем перейти к предрелизному тестированию, на котором происходит так называемый полишинг (polishing eng. Полировка, общепринятый в среде разработчиков компьютерных игр термин, обозначающий процесс доработки мелких деталей, которые на первый взгляд не бросаются в глаза)

Этап презентации содержит в себе само создание презентации готового продукта. Описания достигнутого результата в ходе его реализации и так же сама демонстрация перед людьми.

1. **Выбор средств разработки**

Код используемый в проекте был написан в IDE Visual Studio 2022. IDE -интегрированная среда разработки (integrated development enviroment), программный комплекс включающий в себя: текстовый редактор, компилятор, средства отладки. Все это способствует ускорению процесса разработки. Visual Studio 2022 – это бесплатный редактор кода, который поддерживает множество языков программирования, в том числе и объектно-ориентированный язык программирования C#, разработанный компанией Microsoft. Этот редактор без сомнения удобен тем, что имеет объединение с Git, автоматическим дополнением разрабатываемого кода, подсветке синтаксиса, возможности отладки и быстрому доступу к файлам проекта. Помимо этого, редактор имеет удобную интеграцию с другим средством разработки – Unity. Постоянные обновления делает его, по моему мнению, одним из лучших редакторов для разработки приложений на языке C#.

C# - Высокоуровневый объектно-ориентированный язык программирования, поддерживающий в себе все принципы ООП, такие как полиморфизм, наследование, перегрузка операторов, статическую типизацию. На этом языке программирования разработано множество программ на самые разные платформы начиная с приложения на мобильные устройства и заканчивая веб-приложениями. Наиболее часто на C# пишут программы для десктопных устройств, игры и веб-сервисы.

Unity – межплатформенная среда разработки компьютерных игр. Межплатформенность дает возможность разрабатывать один продукт, выпуская его на множество платформ, начиная с наиболее популярных Windows, macOS, Android, iOS и заканчивая портативными консолями и веб-приложениями, что делает его объединяющим (Unity с англ. - единство). Самым большим достоинством, которыми может похвастаться Unity это современный графический интерфейс с Drag&Drop интерфейсом, позволяющий перетаскивать с помощью мыши множество объектов разработки (звуки, 2D и 3D объекты, программный код, модули). Unity Asset Store – популярный магазин содержащий готовые решения для множества разрабатываемых проектов, отличное подспорье, ускоряющий процесс разработки в разы.

КОМПАС-3D – Российский программный САПР комплекс, позволяющий оперативно создавать и выпускать чертежи сложных изделий, таблицы, макеты и прочие документы. САПР технология – система автоматизированного проектирования, позволяющий эффективно с минимальными затратами времени и ресурсов оперативно проводить разработку моделей, макетов и чертежей. Данный инструмент стал стандартом при проектировании среди множества производственных компаний.

Blender – Программное обеспечение используемое при разработке 3D моделей. Имеет плотную интеграцию со средой разработки Unity, что делает его отличным кандидатом. Данный инструмент распространяется в том числе и по свободный лицензии, что делает его доступным и бесплатным.

1. **Архитектура ПО**

Основными требованиями при выборе архитектуры для ИС являются быстродействие, надежность, защита, безопасность и масштабируемость. В настоящее время наиболее распространенными архитектурами являются: файловый сервер; клиент-сервер; многоуровневая архитектура.

Файловый сервер – это программное обеспечение, которое позволяет создать общее хранилище файлов и предоставляет доступ к ним через сеть. Он обеспечивает управление доступом к файлам, резервное копирование и восстановление данных, а также защиту информации от злоумышленников. Из плюсов файлового сервера можно отметить: централизованное хранение файлов, улучшенная безопасность, легкий доступ к файлам для всех пользователей в сети, возможность резервного копирования и восстановления данных. Минусы такой архитектуры являются: низкая пропускная способность пользователей, необходимость поддержки и обслуживания сервера, высокие затраты на оборудование, риск потери данных при технических неполадках.

У архитектуры клиент-сервера плюсы и минусы такие же, как и у файлового сервера, но у клиент-сервера нет проблем с пропускной способностью пользователей.

Многоуровневая архитектура более сложная архитектура, имеющая свои плюсы, а именно: улучшенная надежность и безопасность, более простое и понятное проектирование и разработка ПО, возможность изменения ПО благодаря разным уровням, изменения в одном уровне не коснуться этих же изменений на другом. Минусы: сложность проектирования из-за множества уровней, т.к. их нужно связывать между собой, затраты на обеспечение связи и взаимодействия между уровнями, возможны сбои между передачи данных между уровнями.

Рассмотрев плюсы и минусы каждой архитектуры, для реализации сайта для института была выбрана клиент-серверная архитектура. Эта архитектура дает возможность оптимально разделить обязанности между клиентской и серверной частями сети. Запросы отправляются на сервер, где обработка идет последовательно, а результаты данной обработки отправляются на рабочую станцию, что снижает информативный поток и снимает нагрузку с локальной сети.

1. **Требования к программному продукту**

Требования к программному продукту нужны для того, чтобы понять какие основные части необходимы для реализации того или иного проекта. Требований бывает несколько и каждое отвечает за разные сферы, хоть и они все соответствуют одной теме.

В первую очередь стоит отметить бизнес-требования. Они определяют цели заказчика и созданы с целью помощи для пользователей или клиентов. В бизнес-требованиях предполагается, что продукт будет вспомогательным инструментом, как для пользователя, так и для заказчика.

Для пользователей это удобная возможность быстро найти нужную информацию, сэкономив множество ресурсов. Для заказчика – возможность увеличить конверсию сайта, т.е. увеличить будущих клиентов.

Благодаря сайту ВУЗ открывает возможность абитуриентам из других регионов и городов ознакомиться с правилами подачи документов, оплаты и других интересующих новостей, без личного присутствия.

Пользовательские требования – эти требования дают четкое представление, как по желанию пользователя должен выглядеть продукт. В данной работе можно представить несколько требований от пользователей, а именно: быстрый поиск интересующей специальности и проходные баллы на нее, небольшая информация о том, какую профессию получит абитуриент по окончании учебного заведения, ознакомление и скачивание необходимых документов и справок, получение реквизитов для оплаты. В случае если абитуриенту хватает баллов на поступление, но не хватает на бюджетное место, то вывод стоимости обучения за год.

Атрибуты качества – эти качества включают в себя стандарты, теорию, оценки и обеспечения качества.

В данной работе атрибутами качества должна служить скорость загрузки страницы, запросы не должны превышать 3-5 секунд при нагрузке на сайт до 20 посетителей, база данных должна устанавливаться на серверах MySql, MS SQL Server или Oracle, сайт должен быть адаптирован для мобильных устройств.

Стоит заметить, что вышеперечисленные примеры не совсем атрибуты качества, а именно требования к ним. Быстрая загрузка страницы и решение проблемы с нагрузкой называется производительностью и надежностью. Работа с серверами MySql, MS SQL Server или Oracle называется переносимостью, а адаптивность под мобильные устройства – удобством использования.

Функциональные требования – простыми словами, это те требования, которые не входят в другие категории. В описании этих требований должно быть указано о том, как должна функционировать система, и она должна быть представлена в разных форматах.

В данной работе к функциональным требованиям можно отнести поисковые запросы по ключевым словам и разрешение скачивания документов и справок только после авторизации.

1. **Фаза тестирования**

Тестирование — важная часть процесса разработки программного обеспечения. Существует множество видов тестирования, которые помогают проверить качество и функциональность продукта.

Одним из основных видов тестирования является функциональное тестирование. Его цель — проверить, выполняются ли функциональные требования к продукту. Еще один важный тип — нагрузочное тестирование. Он позволяет проверить работоспособность изделия при большой нагрузке.

Регрессионное тестирование помогает проверить корректность работы после внесения изменений в продукт. Тестирование безопасности направлено на проверку защищенности продукта от взломов и хакерских атак.

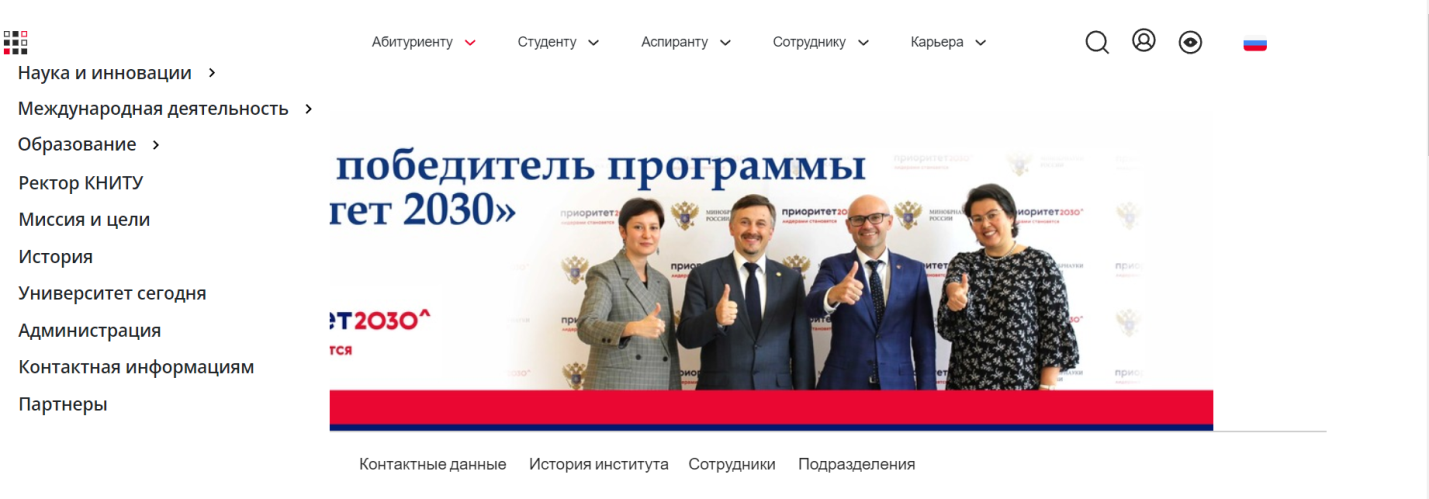
Автоматизированное тестирование проводится с помощью специального программного обеспечения.

Каждый тип тестирования имеет свои особенности и цели. Использование нескольких видов тестирования позволяет более точно и безошибочно проверить качество продукта и повысить его надежность.

В данной работе будет употреблено только функциональное тестирование, т.к. оно позволяет визуально проверить реализацию заявленных возможностей.

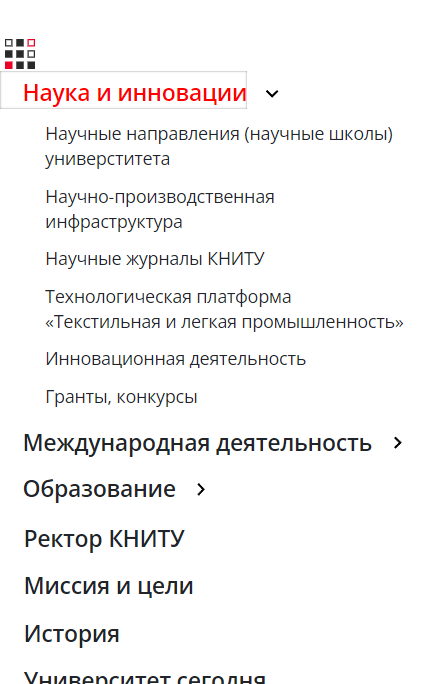
Для функционального тестирования сайта были смоделированы различные ситуации, которые могут возникнуть при работе веб-страницы.

* Раскрывающее меню для быстрой навигации по сайту. Работа указана на рис. 9.



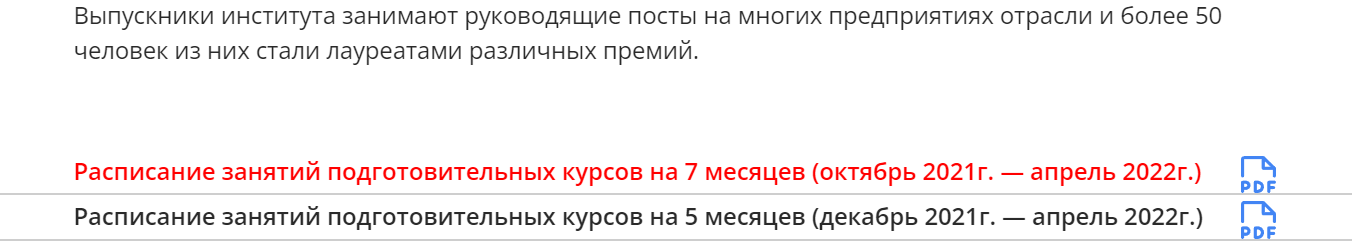
**Рис. 9. Демонстрация меню**

* Подробные подразделы в меню. Работа указана на рис. 10.



**Рис. 10. Демонстрация раскрытия подразделов**

* Реализация метод “Якоря”. При клике на кнопки на главной странице, сайт автоматически прокручивает до нужного раздела.
* Автоматическое скачивание расписания за семестр. Работа указана на рис. 11.



**Рис. 11. Демонстрация скачивания файлов**

* В разделе сотрудники пользователь может нажать на ФИО сотрудника и перейти на страницу с более подробной информацией о нем. На вкладке есть информация о достижениях преподавателей, их стаж работы, по каким профильным предметам обучает. Дополнительная информация касается не только преподавателей, но и других сотрудников института.
* Т.к. в институте множество подразделений, то было решено добавить небольшие фрагменты, куда пользователь также может перейти, кликнув по названию интересующего подразделения. Будет указана деятельность, подробная информация, график и часы работы.
* Контактные данные. Вся необходимая информация о местоположении, контактах и так далее находится здесь. При клике на значки социальных сетей, пользователя автоматически перекинет на страницу института в выбранной соц.сети.

**Заключение**

В ходе курсовой работы был создан сайт для института технологии легкой промышленности, моды и дизайна в приложении Visual studio Code. В разработке были использованы: язык гипертекстовой разметки html, таблица стилей css и язык программирования JavaScript для создания анимации на странице. Макет контекстно создан с помощью сервиса Figma, все детали и уточнения были учтены и реализованы с помощью данного сервиса.

Данная работа включает в себя анализ требований, разработку и внедрение программных инструментов, которые позволяют пользователю выполнять все функции веб-страницы.

В результате работы была создана веб-страница сайта института с более свежим дизайном и обновленными механиками.

Сайт выполняет следующие функции:

* Облегчает поиск информации для пользователей. Так как основная информация вынесена на главные страницы
* Сайт соответствует современному макету, от чего будет меньше проблем с браузером и вся страница будет отображена корректно.
* Учтено то, что пользователи могут посещать сайт с разных устройств без потери “картинки”. В расчет взято адаптивное создание сайтов.
* Современный и приятный дизайн. Данный пункт очень важен, чтобы нахождение на сайте было комфортным, были учтены цветовые гаммы с дизайнерами, чтобы снизить напряжение для глаз.
* Благодаря вышеперечисленным функциям, увеличивается и конверсия сайта, а это тоже очень важно. Ведь частота посещения сайта увеличивает его место в поисковых запросах.
* Защита пользовательских и общих данных. В защиту системы входит: система управления, хостинг, надежный пароль, бекап сайта.
* Сайт соответствует выбранной тематике, т.е. не должно быть лишней информации не по теме.

**Список литературы**

1. Брюс Лоусон, Реми Шарп - “Изучаем HTML5. Библиотека специалиста”, Питер, 2012, 304 стр.
2. Кристофер Шмитт, Кайл Симпсон - "HTML5: Рецепты программирования", Питер, 2012, 288 стр.
3. Эстель Вейл - "HTML5: Разработка приложений для мобильных устройств", 2015, 480 стр.
4. Дженнифер Нидерст Роббинс - "HTML5. Карманный справочник", Вильямс, 2016, 192 стр.
5. Джон Дакетт - "HTML и CSS. Разработка и дизайн веб-сайтов", Эксмо, 2017, 480 стр.
6. Крис Минник, Эд Титтел - "HTML5 и CSS3 для чайников ", Диалектика, 2016, 400 стр.
7. А. Хрусталев, А. Кириченко "HTMLS + CSS3. Основы современного WEB-дизайна", Наука и Техника, 2018, 352 стр.
8. Бен Фрейн (Ben Frain) - "HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств", Питер, 2017, 272 стр.
9. Лия Веру - "Секреты CSS. Идеальные решения ежедневных задач", Питер, 2017, 336 стр.
10. Дэвид Макфарланд - "Новая большая книга CSS", Питер, 2016, 720 стр.
11. Эрик А. Майер - "CSS. Карманный справочник", Вильямс, 2017, 288 стр.
12. Эспозито, Дино Разработка приложений для Windows 8 на HTML5 и JavaScript / Дино Эспозито. - М.: Питер,**2016. - 455 c.**
13. <https://metanit.com/web/html5/>
14. Дженнифер Нидерст Роббинс "HTML5, CSS3 и JavaScript. Исчерпывающее руководство". 4-ое издание
15. <https://webref.ru/>
16. Фрэйн, Бен HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Бен Фрэйн. - М.: Питер, 2017. - 272 c.