

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владивостокский государственный университет»
Академический колледж

ОТЧЕТ
ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ
по профессиональному модулю
ПМ.03 Проектирование и разработка информационных
систем
ПМ.05 Соадминистрирование и автоматизация баз данных
и серверов

программы подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование

период с «03» июня по «22» июня 2024 года

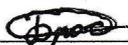
Наименование предприятия: ООО «РСТ-ТУР»

Студент гр. СО-ИП-21-2



Г.С. Погожев

Руководитель практики
от предприятия



В.В. Бросалин

Отчет защищен:
с оценкой

4

Руководитель
практики от ОО



Е.И. Ершкова

Владивосток 2024

Содержание

Введение.....	3
1 Общая характеристика деятельности предприятия ООО «РСТ ТУР».....	4
1.1 Программно-аппаратное обеспечение предприятия	4
1.2 Методы автоматизации процессов предприятия для повышения производительности	5
1.3 Описание разрабатываемой системы для предприятия ООО «РСТ-ТУР».....	6
2 Проектирование информационной системы для предприятия ООО «РСТ-ТУР».....	8
2.1 Анализ и выбор инструментальных средств для проектирования информационной системы	8
2.2 Проектирование информационной системы с помощью унифицированного языка моделирования.....	10
2.3 Проектирование интерфейса пользователя	12
3 Разработка информационной системы для предприятия ООО «РСТ-ТУР».....	14
3.1 Анализ и выбор инструментальных средств для разработки информационной системы	14
3.2 Описание этапов разработки.....	15
3.3 Тестирование информационной системы.....	20
Заключение.....	22
Список использованных источников	23

Введение

Место прохождения производственной практики – ООО «РСТ-ТУР», г. Владивосток.

Цель прохождения практики – является овладение профессиональными компетенциями по профессиональным модулям ПМ.03 «Проектирование и разработка информационных систем» и ПМ.05 «Сoadминистрирование и автоматизация баз данных и серверов».

Для выполнения цели, необходимо выполнить следующие задания:

- ознакомиться с предприятием;
- ознакомиться с перечнем и конфигурацией средств вычислительной техники, а также архитектурой сети. Ознакомиться с перечнем и назначением программных средств;
- описать базу данных предприятия;
- работать с информационной системой предприятия. При необходимости восстанавливать информацию в информационной системе при отказах системы;
- анализировать и оценивать предметную область и методы определения стратегии развития организации;
- предложить методы автоматизации процессов для повышения производительности предприятия;
- описать информационную систему, предложенную для внедрения;
- провести анализ существующих информационных систем по выбранной тематике;
- разработать план мероприятий для этапов проектирования, разработки, тестирования и внедрения информационной системы;
- проанализировать и выбрать инструментальные средства для проектирования и разработки информационной системы;
- спроектировать информационную систему для предприятия;
- разработать информационную систему для предприятия;
- оформить тестовую документацию с тест-кейсами для основной функциональности информационной системы;
- протестировать разработанную систему. Исправить возникшие ошибки.

Производственная практика является важным элементом обучения, она проводится с целью обрести и улучшить навыки в профессиональной деятельности. В производственную практику заложены требования к правильному выполнению обязанностей специалиста, выстраивание коммуникаций с руководителем практики. Производственная практика даёт возможность закрепить знания, которые были получены во время учебной деятельности [1].

1 Общая характеристика деятельности предприятия ООО «РСТ ТУР»

1.1 Программно-аппаратное обеспечение предприятия

Предприятие ООО «РСТ-ТУР» находится по адресу Приморский край, г. Хабаровск, ул. Ленина 18 В. Предприятие представляет собой ведущую российскую туристическую компанию, которая за годы своей деятельности зарекомендовала себя как надежный поставщик качественных туристических услуг. Основная миссия компании – предоставление полного спектра услуг в области туризма, что позволяет клиентам наслаждаться беззаботным отдыхом, будь то путешествие по России или за ее пределами. РСТ-ТУР выступает не только как организатор туров, но и как агрегатор предложений от ведущих туроператоров, отелей и авиакомпаний. Продажа туров:

- компания предлагает обширный ассортимент туров, включая экзотические и популярные направления. Каждый тур подбирается с учетом индивидуальных предпочтений клиентов, что обеспечивает максимальное удовлетворение их ожиданий;

- в состав туров входят перелеты, проживание в отелях различных категорий, питание, трансферы, а также дополнительные услуги, такие как экскурсии и страхование;

- удобный и функциональный сайт компании позволяет клиентам самостоятельно выбирать и бронировать туры, а также ознакомиться с актуальными спецпредложениями и акциями.

Организация поездок:

- РСТ-ТУР специализируется на организации как индивидуальных, так и групповых туров "под ключ", предлагая полный комплекс услуг по планированию и реализации путешествий;

- компания берет на себя все заботы по подбору оптимальных маршрутов, бронированию отелей, покупке билетов и организации досуга, что позволяет клиентам сосредоточиться на предвкушении предстоящего отдыха;

- опытные сотрудники компании обеспечивают качественное сопровождение и поддержку на всех этапах путешествия, гарантируя решение любых вопросов в кратчайшие сроки.

Платформа для бронирования и оплаты:

- сайт РСТ-ТУР предлагает удобный интерфейс для поиска, бронирования и оплаты туров и отелей онлайн, что значительно экономит время клиентов;

- благодаря прямому сотрудничеству с туроператорами и гостиницами, компания предлагает выгодные условия и специальные тарифы;

– встроенные инструменты управления бронированиями позволяют клиентам в любой момент изменить или отменить свое бронирование, а также получить всю необходимую информацию о предстоящей поездке.

РСТ-ТУР активно сотрудничает с проверенными партнерами и поставщиками туристических услуг, что позволяет компании гарантировать высокое качество и безопасность путешествий. Особое внимание уделяется подбору туроператоров и отелей, которые соответствуют строгим стандартам качества и сервиса. Забота о клиентах проявляется не только в предоставлении качественных услуг, но и в разработке индивидуальных туристических продуктов, способных удовлетворить самые разнообразные запросы и предпочтения. РСТ-ТУР постоянно следит за тенденциями рынка и инновациями в сфере туризма, чтобы предлагать своим клиентам только лучшее. Таким образом, РСТ-ТУР является надежным партнером для всех, кто ценит комфорт, безопасность и высокий уровень обслуживания во время своих путешествий [2].

1.2 Методы автоматизации процессов предприятия для повышения производительности

Основной деятельностью компании является автоматизация процессов. Поддержка платформы онлайн-бронирования, системы управления бизнес-процессами и взаимоотношениями с клиентами, а также систему управления недвижимостью.

В современных системах управления взаимоотношениями с клиентами (CRM) присутствует возможность сбора и хранения разнообразной информации о клиентах, включая контактные данные, личную информацию, историю взаимодействия, а также информацию о реакции клиента на различные рекламные кампании и многое другое. Такие высокофункциональные платформы позволяют использовать маркетинговые возможности CRM и эффективно монетизировать клиентскую базу. В то же время, более простые системы CRM предоставляют меньший объем информации и функциональности.

CRM-система представляет собой комплекс программных средств, предназначенных для автоматизации, связанных с управлением взаимоотношениями с клиентами. Она позволяет управлять информацией, их потребностях и предпочтениях, а также обеспечивает возможность проведения анализа и прогнозирования поведения клиентов. В результате внедрения CRM-системы улучшается качество обслуживания клиентов, повышается лояльность их к компании, а также увеличивается эффективность работы сотрудников.

Основной деятельностью компании является автоматизация процессов. Поддержка платформы онлайн-бронирования, системы управления бизнес-процессами и взаимоотношениями с клиентами, а также систему управления недвижимостью.

Поскольку ООО «РСТ-ТУР» является агрегатором туристических объектов и ежегодно заключает контракты на сотрудничество с новыми отелями и базами отдыха, компании необходима система CRM для автоматизации бизнес-процессов. CRM-система представляет собой комплекс программных средств, предназначенных для автоматизации и оптимизации бизнес-процессов, связанных с управлением взаимоотношениями с клиентами. Она позволяет эффективно управлять информацией о клиентах, их потребностях и предпочтениях, а также обеспечивает возможность проведения анализа и прогнозирования поведения клиентов. В результате внедрения CRM-системы улучшается качество обслуживания клиентов, повышается лояльность их к компании, а также увеличивается эффективность работы сотрудников.

Такие многофункциональные CRM-платформы позволяют эффективно использовать собираемые данные для проведения таргетированных маркетинговых мероприятий и монетизации клиентской базы. Иными словами, они дают возможность компаниям глубже понять своих клиентов и предлагать им наиболее подходящие продукты и услуги.

В то же время, более простые, базовые системы CRM предоставляют более ограниченный набор возможностей по сбору и анализу клиентской информации. Они не обладают такими широкими маркетинговыми возможностями и не позволяют в полной мере монетизировать накопленные данные о клиентах [3].

1.3 Описание разрабатываемой системы для предприятия ООО «РСТ-ТУР»

Информационная система представляет собой специализированное программное решение, предназначенное для эффективного управления и отслеживания уведомлений, связанных с деятельностью туристического агентства. Основные функциональные возможности данной информационной системы:

1) регистрация и хранение уведомлений:

- система позволяет регистрировать различные типы уведомлений, такие как изменения, задержки или изменения в туристическом обслуживании;
- все уведомления хранятся в централизованной базе данных, обеспечивая их доступность и структурированность.

2) отслеживание и управление уведомлениями:

- сотрудники турагентства могут просматривать, фильтровать и сортировать зарегистрированные уведомления по различным параметрам, таким как дата, тип уведомления, клиент, статус обработки;

- система предоставляет возможность назначения ответственных сотрудников для обработки уведомлений, а также отслеживания статуса их обработки.

3) уведомления и оповещения:

- система автоматически генерирует и отправляет уведомления клиентам турагентства о произошедших изменениях, отмененных рейсах или других важных событиях;

- уведомления могут быть доставлены через различные каналы, такие как электронная почта, SMS, push-уведомления или мобильное приложение.

4) отчетность и аналитика:

- система предоставляет инструменты для формирования отчетов, позволяющих анализировать статистику по уведомлениям, частоту их возникновения, эффективность обработки и другие ключевые показатели;

- аналитические данные помогают руководству турагентства принимать обоснованные управленческие решения и повышать качество обслуживания клиентов.

5) интеграция с другими системами:

- информационная система может быть интегрирована с другими программными решениями, используемыми в турагентстве, такими как CRM, бронирование туров, управление финансами;

- интеграция обеспечивает синхронизацию данных и повышает эффективность работы всех взаимосвязанных систем.

В целом, внедрение данной информационной системы позволяет туристическому агентству значительно повысить качество обслуживания клиентов, оперативно реагировать на изменения в туристической индустрии, оптимизировать внутренние бизнес-процессы и, как результат, улучшить общую эффективность и конкурентоспособность бизнеса. Встроенные отчеты и аналитические инструменты позволяют руководству турагентства анализировать статистику по уведомлениям, выявлять проблемные области и принимать обоснованные управленческие решения. Данные об эффективности обработки уведомлений помогают оптимизировать внутренние бизнес-процессы и повысить качество обслуживания.

2 Проектирование информационной системы для предприятия ООО «РСТ-ТУР»

2.1 Анализ и выбор инструментальных средств для проектирования информационной системы

Для проектирования веб-страниц интернет-магазина и его работы будут использованы приложения Figma, Adobe Illustrator и Draw.io

Figma – это популярный онлайн-инструмент для создания интерфейсов и прототипирования. Он имеет ряд преимуществ:

1) совместная работа в реальном времени: несколько дизайнеров, разработчиков и других участников проекта могут работать над одним файлом одновременно, наблюдать за изменениями друг друга и вносить свои правки. Это значительно ускоряет процесс разработки [4];

2) версионирование: Figma автоматически сохраняет историю изменений, что позволяет вернуться к предыдущим версиям проекта в любой момент;

3) библиотеки компонентов: Figma позволяет создавать собственные библиотеки повторно используемых компонентов интерфейса (кнопки, иконки, блоки и т.д.);

4) комментарии: необходимо добавлять комментарии к коду, особенно к сложным или непонятным частям, чтобы облегчить понимание.

Благодаря своей кроссплатформенности, коллаборативным возможностям и мощному функционалу, Figma становится одним из самых популярных инструментов среди UI/UX-дизайнеров и командных проектов по разработке интерфейсов [5].

Adobe Illustrator – это второе по популярности графическое приложение после Photoshop, входящее в пакет Adobe Creative Cloud. Оно широко используется дизайнерами, иллюстраторами, художниками и другими творческими профессионалами для создания и редактирования векторных изображений [6]. Рассмотрим основные преимущества Adobe Illustrator:

1) векторная графика: Illustrator работает с векторными изображениями, что позволяет масштабировать их без потери качества. Это идеально подходит для создания логотипов, иконок, иллюстраций и других элементов, которые должны оставаться четкими при любом размере;

2) инструменты рисования: приложение предлагает широкий набор инструментов для рисования, включая перо, кисти, каллиграфические кисти, фигуры и многое другое. Это позволяет создавать сложные иллюстрации и художественные работы;

3) работа с текстом: Illustrator обладает мощными инструментами для работы с текстом, позволяя применять различные эффекты, стили и манипуляции с текстом;

4) слои и управление объектами: приложение поддерживает работу со слоями, что упрощает организацию и редактирование объектов в сложных проектах;

5) цветовые режимы и управление цветом: Illustrator поддерживает различные цветовые режимы, включая RGB, CMYK и Pantone, что важно для печатной продукции и брендинга;

6) интеграция с другими приложениями Adobe: Illustrator легко интегрируется с другими приложениями;

7) плагины и расширения: существует большое количество плагинов и расширений, которые добавляют новые функции и инструменты в Illustrator, расширяя его возможности;

8) экспорт в различные форматы: Illustrator позволяет экспортировать работы в различные форматы файлов, такие как SVG, PDF, EPS, PNG и другие, что обеспечивает гибкость в дальнейшем использовании изображений.

Adobe Illustrator является одним из самых популярных и мощных инструментов для создания векторной графики благодаря своим богатым возможностям, интеграции с другими приложениями Adobe и широкому признанию в индустрии дизайна и иллюстрации [7].

DrawIO (также известный как Diagrams.net) – это мощный инструмент для создания диаграмм и схем, обладающий рядом преимуществ:

1) бесплатность и открытость:

– DrawIO является бесплатным и open source инструментом, что делает его доступным для широкого круга пользователей;

– отсутствие необходимости платить за лицензию позволяет использовать инструмент без дополнительных расходов.

2) кроссплатформенность:

– DrawIO работает в любом современном веб-браузере, что обеспечивает его доступность на различных операционных системах;

– пользователи могут создавать и редактировать диаграммы как на ПК, так и на мобильных устройствах.

3) простота и интуитивность:

– интерфейс DrawIO отличается простотой и понятностью, что позволяет быстро освоить инструмент даже пользователям без опыта;

– широкий набор готовых шаблонов и элементов диаграмм упрощает процесс создания и редактирования схем.

4) расширяемость и гибкость:

– DrawIO предоставляет возможность расширения функциональности с помощью плагинов и интеграций;

- пользователи могут настраивать интерфейс и добавлять собственные библиотеки элементов под свои потребности.

5) удобное сотрудничество:

- DrawIO поддерживает совместную работу над диаграммами в реальном времени, что позволяет командам эффективно взаимодействовать;

- диаграммы могут быть легко экспортированы в различные форматы (PDF, PNG, SVG) для дальнейшего использования и распространения.

6) интеграция с другими сервисами:

- DrawIO интегрируется с популярными облачными хранилищами

Таким образом, DrawIO предлагает пользователям широкий спектр преимуществ, включая бесплатность, кроссплатформенность, простоту использования, расширяемость, удобство совместной работы и интеграцию с другими сервисами.

2.2 Проектирование информационной системы с помощью унифицированного языка моделирования

UML (Unified Modeling Language) – это универсальный язык моделирования, предназначенный для визуального представления, спецификации, проектирования и документирования программных систем.

Основные характеристики UML:

1) универсальность: UML является стандартным языком моделирования, который может применяться для различных предметных областей, программных платформ и методологий разработки ПО;

2) графическое представление: UML использует набор графических нотаций для описания структуры и поведения программных систем;

3) независимость от процесса разработки: UML является независимым от процесса разработки программного обеспечения, что означает, что он может применяться в различных методологиях (Waterfall, Agile, Iterative и др.).

Основные виды UML-диаграмм:

- диаграммы структуры (классов, компонентов, развертывания, пакетов);

- диаграммы поведения (вариантов использования, деятельности);

- диаграммы взаимодействия (коммуникации, последовательности).

Проектирование сайта проведено с целью на создание удобного и функционального веб-ресурса, который предоставляет пользователям возможность ознакомления с ассортиментом, а администратору – эффективное управление содержимым сайта и взаимодействием с клиентами. Диаграмма навигации сайта представлена на рисунке 1.

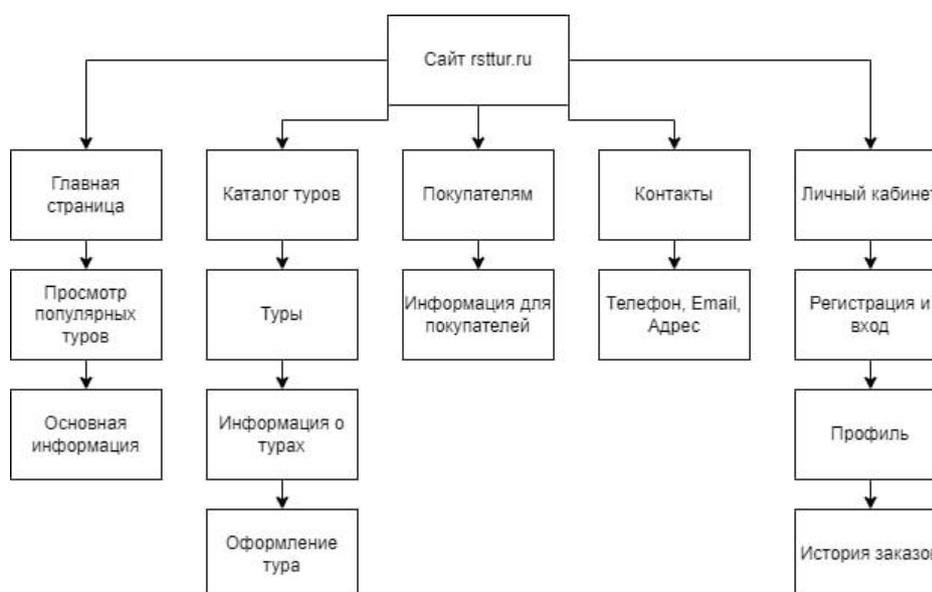


Рисунок 1 – Диаграмма компонентов сайта

Представленная диаграмма отображает возможности сайта, а также ключевые компоненты, с которыми могут взаимодействовать как пользователи, так и администратор. Это позволяет понять, какие функции доступны на сайте и как они распределены.

Диаграмма use-case для пользователей сайта и администратора представлена на рисунке 2.

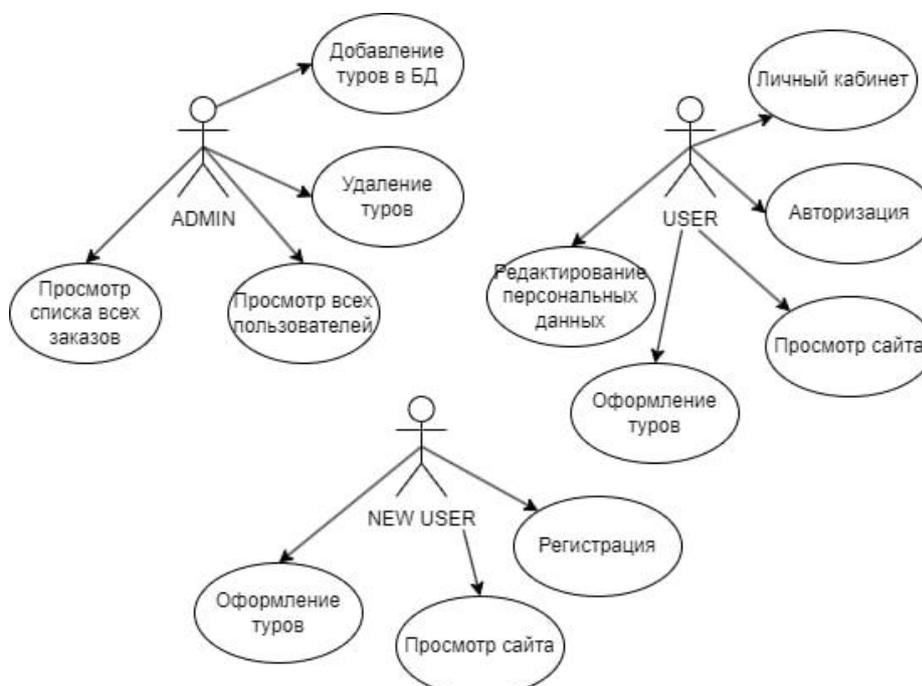


Рисунок 2 – Use-case диаграмма

Эта диаграмма демонстрирует чёткое разделение ролей и возможностей между администратором, зарегистрированными пользователями и незарегистрированными гостями. Администратор может редактировать базу данных информационной системы, добавлять и

просматривать данные всех клиентов. Авторизованный пользователь информационной системы может зайти в свой личный кабинет, добавить нового клиента. Данные клиента будут видны только тому пользователю, который его добавил.

3. Диаграмма последовательности сайта со стороны пользователя представлена на рисунке

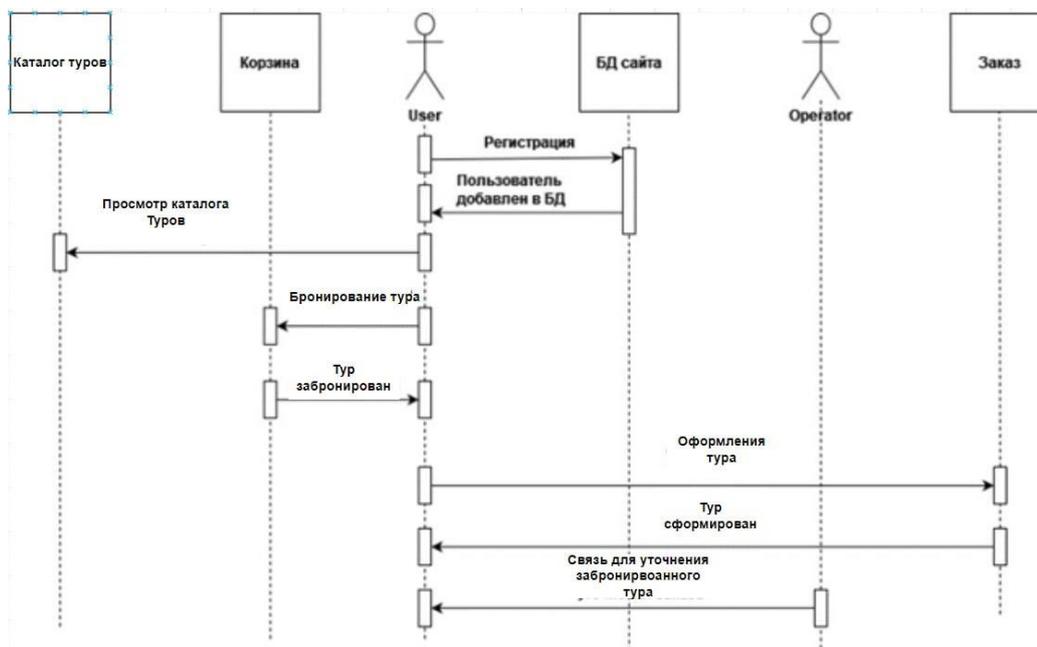


Рисунок 3 – Диаграмма последовательности пользователя сайта

Эта диаграмма демонстрирует ключевые процессы, с которыми взаимодействует пользователь при работе с сайтом: регистрация, просмотр каталога, добавление товаров в корзину и оформление заказа. Эти процессы обеспечивают пользователям интуитивный и удобный способ совершения покупок на сайт. Таким образом, данная диаграмма отражает ключевые функциональные возможности, которые обеспечивают пользователям интуитивный и удобный процесс совершения покупок на сайте. Эти процессы являются критически важными для создания положительного пользовательского опыта и увеличения конверсии на сайте. Успешное оформление заказа подтверждается пользователю и инициирует дальнейшие действия по обработке и исполнению заказа.

2.3 Проектирование интерфейса пользователя

Проектирование интерфейса пользователя – это ключевой процесс создания визуальной структуры, с которой взаимодействует пользователь при использовании приложения или веб-сайта. Главная цель проектирования заключается в обеспечении удобства, простоты и интуитивности взаимодействия с системой, а также в создании приятного визуального опыта. Эффективный интерфейс позволяет пользователям быстро

находить нужную информацию, совершать действия и достигать целей с минимальными усилиями. Макет уведомления представлен на рисунке 4.

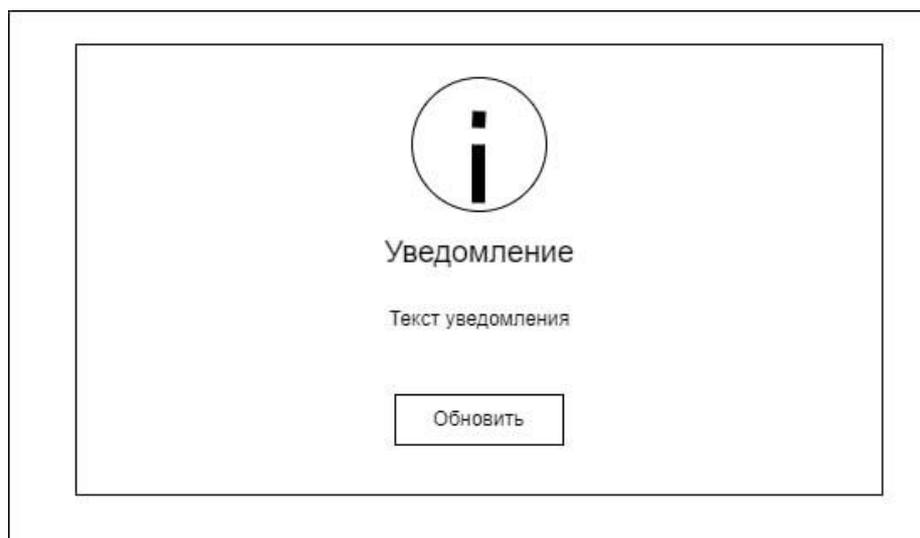


Рисунок 4 – Макет уведомления.

Навигация играет решающую роль в проектировании интерфейса, так как пользователи должны легко перемещаться между разделами приложения или сайта. Для этого важно продумать логичную структуру меню, наличие хлебных крошек и удобные кнопки для возврата на предыдущие страницы. Эффективная навигация помогает пользователю быстро находить нужную информацию, снижая вероятность путаницы и затруднений. Важным аспектом также является консистентность интерфейса. Все элементы, такие как кнопки, ссылки, выпадающие меню, должны иметь единообразный внешний вид и поведение на всех страницах приложения. Это способствует созданию предсказуемого и понятного пользовательского опыта, что значительно упрощает работу с интерфейсом. Консистентность формирует ощущение целостности продукта и облегчает его восприятие, делая взаимодействие более комфортным.

3 Разработка информационной системы для предприятия ООО «РСТ-ТУР»

3.1 Анализ и выбор инструментальных средств для разработки информационной системы

Visual Studio Code (VS Code) – это бесплатный кроссплатформенный редактор кода с открытым исходным кодом, разработанный Microsoft. Он предназначен для написания, отладки и развертывания кода, а также поддерживает несколько языков программирования. Вот основные преимущества VS Code:

1) легкий и быстрый: VS Code построен на базе Electron и отличается малым потреблением ресурсов, высокой производительностью и быстрым временем запуска;

2) кроссплатформенность: доступен на Windows, macOS и Linux, что обеспечивает гибкость для разработчиков;

3) интеграция Git: встроенная поддержка системы контроля версий Git позволяет эффективно управлять изменениями в коде. Обеспечивает просмотр изменений кода, совершать коммиты, создавать ветки и выполнять другие операции. Также предоставляет визуальное представление коммитов;

4) расширения: существует огромное количество расширений, добавляющих новый функционал, поддержку языков, отладчики, интеграцию с различными сервисами и многое другое, обеспечивая настройку VS Code под свои нужды;

5) IntelliSense: встроенная функция автодополнения кода (IntelliSense) ускоряет процесс написания кода, предлагая возможные варианты дополнения кода, описания функций, параметров и другую полезную информацию;

6) отладчик: встроенный отладчик позволяет отлаживать приложения непосредственно в редакторе. Позволяет устанавливать точки останова, просматривать значения переменных и шагать по коду во время отладки;

7) терминал: интегрированный терминал позволяет запускать команды и выполнять задачи из среды разработки, поддерживает вкладки, что позволяет открывать несколько сеансов терминала одновременно;

8) многооконный режим: поддерживается разделение редакторов на несколько окон для более удобной работы. Есть возможность расположить окна вертикально, горизонтально или в виде сетки;

9) синхронизация настроек: пользовательские настройки можно синхронизировать между разными устройствами через облачное хранилище.

VS Code стал популярным благодаря своей гибкости, производительности, богатой экосистеме расширений и интеграции с инструментами разработки. Он предлагает

комплексное решение для написания, отладки и развертывания кода для различных языков программирования и платформ.

3.2 Описание этапов разработки

В первую очередь необходимо осуществить корректную настройку Webpack, которая включает добавление значения хэша в поле `chunkFilename`. Это позволит указать хэш для файлов и предотвратить загрузку устаревших файлов, которые больше не существуют в новой сборке приложения.

```
chunkFilename: isEnvProduction
  ? ${folderName}/js/[name].chunk.js?v=${VERSION}
  : isEnvDevelopment && 'static/js/[name].chunk.js',
```

Листинг 1 – Свойство `chunkFilename` в настройке Webpack.

Свойство `chunkFilename` в настройках Webpack является очень важным для обеспечения стабильной работы приложения и предотвращения ошибок, связанных с загрузкой устаревших файлов.

Когда Webpack собирает приложение, он разделяет его на несколько частей, называемых "чанками" (chunks). Каждый чанк представляет собой отдельный файл, который может быть загружен по запросу. Это позволяет оптимизировать загрузку приложения, загружая только необходимые части, а не все сразу.

Свойство `chunkFilename` определяет имя файла, которое будет использоваться для этих чанков. Правильная настройка `chunkFilename` важна, так как она влияет на способ загрузки этих чанков в приложении.

Например, если `chunkFilename` не настроен правильно, браузер может загружать устаревшие версии чанков, что приводит к ошибкам и некорректному отображению приложения. Это может случиться, когда пользователь обновляет приложение, а браузер все еще кэширует старые версии файлов.

Для загрузки чанков в React-приложении часто используется встроенная в React библиотека функция `React.lazy()`. Эта функция позволяет отложить загрузку компонента до тех пор, пока он не понадобится для отображения пользователю. Это помогает ускорить начальную загрузку страницы, так как не все компоненты нужны сразу, и также уменьшает объем передаваемых данных, что особенно важно для мобильных устройств с медленным интернет-соединением.

Использование `React.lazy()` может также улучшить производительность приложения, так как компоненты будут загружаться по мере необходимости, а не все сразу. Это помогает

избежать ситуаций, когда пользователь вынужден ждать загрузки всего приложения, прежде чем он сможет начать взаимодействовать с ним.

В итоге, правильная настройка `chunkFilename` в Webpack и использование `React.lazy()` являются ключевыми для обеспечения стабильной работы приложения, улучшения его производительности и повышения удовлетворенности пользователей.

Для реализации всплывающего окна оповещения была использована библиотека `React, sweetalert2`. Настройка вызова всплывающего окна через библиотеку `sweetalert2` представлен на листинге 2.

```
Swal.fire({
  type: 'info',
  title: 'Обновление',
  text: Приложение устарело. Требуется обновление!,
  showCancelButton: false,
  showConfirmButton: true,
  confirmButtonText: 'Обновить'
}).then(result => {
  if (result.value) {
    window.localStorage.setItem(
      'page-has-been-force-refreshed',
      'false'
    );
    window.location.reload();
  }
});
```

Листинг 2 – Настройки вызова всплывающего окна через библиотеку `sweetalert2`

Для реализации этой функциональности можно использовать библиотеку "SweetAlert2". Она позволяет создавать красивые и интерактивные всплывающие окна, которые отображают информацию об обновлениях и дают пользователю возможность обновить страницу одним нажатием кнопки.

Для использования `SweetAlert2`, необходимо подключить библиотеку к проекту. Это можно сделать через CDN или установив ее через менеджер пакетов, например, `npm`. После подключения, можно использовать функции библиотеки для вывода всплывающего окна с информацией об обновлениях и кнопкой "Обновить".

Для отображения уведомления необходимо использовать библиотеку "sweetalert2". Пользователю будет отображаться всплывающее окно, в котором будет указано название и описание обновления. Библиотека "sweetalert2" позволяет настроить поведение окна в соответствии с требованиями.

Внутри всплывающего окна будет присутствовать кнопка "Обновить". При нажатии на эту кнопку будут происходить следующие действия:

- 1) обновление страницы, чтобы отобразить актуальные данные;
- 2) очистка значения, отвечающего за показ оповещения, чтобы избежать повторного отображения уведомления.

Функция загрузки чанков с проверкой на ошибку загрузки представлена на листинге 3.

```
const component = await componentImport()
return component
} catch (error) {
const pageHasAlreadyBeenForceRefreshed = JSON.parse(
  window.localStorage.getItem('page-has-been-force-refreshed') || 'false'
)
if (!pageHasAlreadyBeenForceRefreshed) {
  // Assuming that the user is not on the latest version of the application.
  // Let's refresh the page immediately.
  window.localStorage.setItem('page-has-been-force-refreshed', 'true')
  Swal.fire({
    type: 'info',
    title: 'Обновление',
    text: Приложение устарело. Требуется обновление!,
    showCancelButton: false,
    showConfirmButton: true,
    confirmButtonText: 'Обновить',
  }).then(result => {
    if (result.value) {
      window.localStorage.setItem(
        'page-has-been-force-refreshed',
```

Листинг 3 – Функция загрузки чанков с проверкой на ошибку загрузки.

Таким образом, пользователь будет вовремя информирован об обновлениях на странице, и сможет легко обновить ее, нажав на кнопку "Обновить" во всплывающем окне. Библиотека "sweetalert2" позволяет реализовать данную функциональность в удобном и интуитивно понятном для пользователя виде.

Для этого можно создать функцию `lazyWithNotification`, и через стандартную конструкцию обработки ошибок `try/catch` функция будет улавливать любые ошибки, возникающие в переданном ей `callback`-функции, и выводить оповещение пользователю. Важно, что она будет проверять, было ли уже показано оповещение, и не будет показывать его повторно, пока пользователь не обновит страницу. Это предотвращает спам оповещениями в случае возникновения ошибки повторно. Типизация функции загрузки чанков представлены на листинге 4.

```
type LazyWithRetryType = <T extends ComponentType<any>>(
  fn: () => Promise<{ default: T }>
) => LazyExoticComponent<T>
```

Листинг 4 – Типизация функции загрузки чанков.

Так как функция реализуется на языке `typescript`, то необходимо её типизировать. Начнём с определения типа `LazyWithRetryType`. Это тип функции, которая принимает на вход параметр `fn` и возвращает тип `LazyExoticComponent<T>`.

1) `<T extends ComponentType<any>>`:

- это генерический тип параметр `T`, который ограничен типом `ComponentType<any>`;
- это означает, что `T` должен быть типом, который наследуется от

`ComponentType<any>`;

- `ComponentType<any>` – это тип, который представляет собой компонент `React`.

2) `(fn: () => Promise<{ default: T }>)`:

– это функция, которая принимает на вход параметр `fn`, который также является функцией;

- функция `fn` возвращает `Promise`, который разрешается в объект с полем `default`, значение которого имеет тип `T`.

3) `=> LazyExoticComponent<T>`:

- Функция, определённая в `LazyWithRetryType`, возвращает `LazyExoticComponent`;

– `LazyExoticComponent<T>` – это тип, который представляет собой ленивый компонент `React` с типом `T`.

1) мы вызываем функцию с типом `LazyWithRetryType`, передавая ей `fn` в виде аргумента;

2) функция `fn` возвращает `Promise`, который разрешается в объект с полем `default`, значение которого имеет тип `T`;

3) функция, определённая в `LazyWithRetryType`, возвращает `LazyExoticComponent<T>`, который представляет собой ленивый компонент `React` с типом `T`.

Этот тип может быть полезен, когда хотите ленивую загрузку компонентов `React` с возможностью повторной попытки, если первая попытка загрузки завершилась неудачно. Он помогает обеспечить правильную типизацию компонентов, что может быть полезно для поддержки и масштабируемости приложения [8]. Результат работы функции с проверкой на ошибку загрузки чанков представлен на рисунке 5.

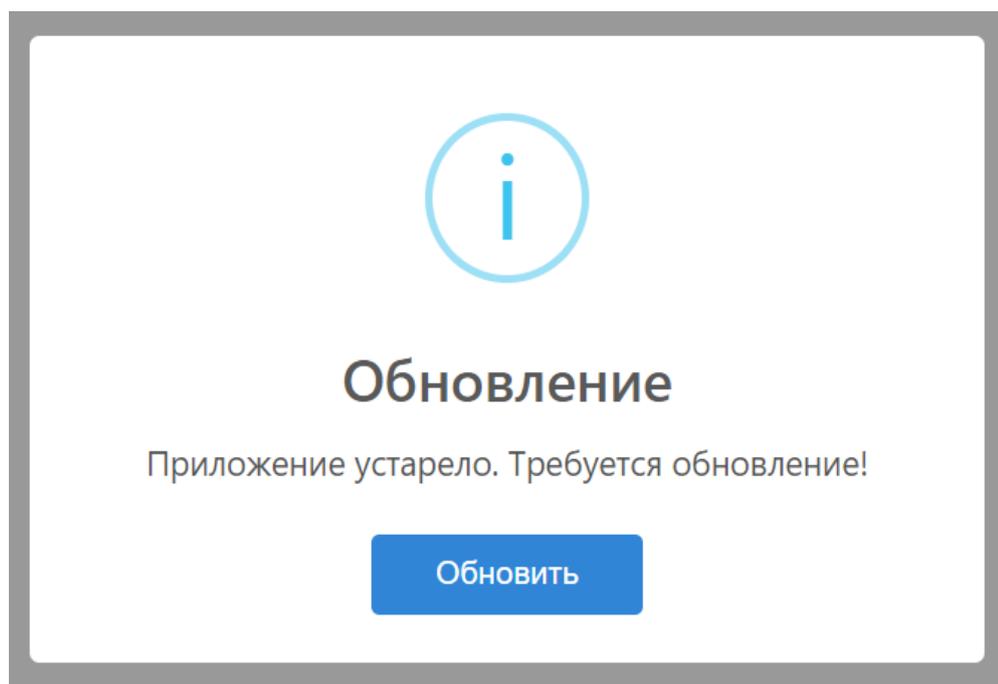


Рисунок 5 – Результат работы функции с проверкой на ошибку загрузки чанков.

Описание работы функции:

1) процесс отслеживания изменений в сборке приложения:

- сравнение версий: при каждом обновлении разработчики увеличивают номер версии приложения. Функция автоматически сравнивает текущую версию приложения с версией, доступной на сервере. Если версия на сервере новее, это означает, что доступно обновление;

- хэш-суммы: для каждой сборки генерируется уникальная хэш-сумма, основанная на содержимом файлов сборки. Функция сравнивает хэш-сумму текущей сборки с хэш-суммой на сервере. Различие в хэш-суммах указывает на изменения в сборке и необходимость обновления;

– временные метки: функция может использовать временные метки последнего обновления, сравнивая дату и время последнего обновления на клиенте с датой и временем обновления на сервере;

2) определение наличия обновлений:

– после отслеживания изменений функция анализирует полученные данные и определяет, было ли обновление сборки. Это делается путем сравнения версий, хэш-сумм или временных меток, как описано выше. Если обнаруживается, что версия на сервере новее, функция активирует механизм оповещения.

3) когда функция определяет наличие обновления, она генерирует уведомление для пользователя. Это уведомление может быть реализовано в различных формах, например:

– всплывающее окно: появляется всплывающее окно с сообщением о необходимости обновления и кнопкой для перезагрузки страницы;

– баннер на странице: на веб-странице отображается баннер с информацией об обновлении и предложением обновить страницу;

– автоматическое обновление: в некоторых случаях, особенно при критических обновлениях, приложение может автоматически перезагружать страницу после предварительного уведомления пользователя.

4) значение функции для стабильности работы приложения:

– эта функция критически важна для поддержания актуальности приложения у конечных пользователей. Она не только обеспечивает корректное отображение содержимого и функциональность веб-приложения, но и способствует повышению безопасности, поскольку обновления часто содержат исправления уязвимостей. Своевременное информирование пользователей о необходимости обновления помогает избежать проблем, связанных с использованием устаревших версий приложения, и обеспечивает лучший пользовательский опыт [9].

Таким образом, реализация функции автоматического оповещения о необходимости обновления является ключевым элементом в стратегии поддержки и обновления веб-приложений, обеспечивая их стабильность, безопасность и актуальность для пользователей [10].

3.3 Тестирование информационной системы

Тестирование системы уведомлений в React является ключевым процессом, обеспечивающим стабильную работу приложения и его взаимодействие с пользователями. Система уведомлений служит для информирования пользователей о различных событиях, таких как успешные действия, ошибки или предупреждения. Для проверки корректности

работы уведомлений необходимо охватить несколько типов тестирования: юнит-тестирование, интеграционное тестирование и тестирование пользовательского интерфейса.

Юнит-тестирование системы уведомлений предполагает проверку отдельных компонентов на предмет их правильного функционирования. На данном этапе важно удостовериться, что каждый компонент уведомления обрабатывает корректно, включая отображение текста, иконок и других визуальных элементов, а также реагирует на изменения в состоянии приложения. Также необходимо проверить корректность передачи данных в уведомления, таких как тип сообщения (ошибка, предупреждение или успех), и убедиться, что эти данные отображаются правильно. Особое внимание уделяется функциям отображения и скрывания уведомлений, которые должны корректно вызываться по завершении определенных событий или по истечении заданного времени.

Интеграционное тестирование включает проверку взаимодействия системы уведомлений с другими частями приложения. Здесь важно убедиться, что уведомления правильно появляются в ответ на различные события, такие как успешное выполнение операций, ошибки при отправке данных или завершение загрузки. Необходимо проверить, что уведомления не только корректно отображаются, но и исчезают по истечении заданного времени или при взаимодействии пользователя с элементами управления (например, при нажатии на кнопку "закрыть"). Также важно убедиться, что уведомления не конфликтуют с другими элементами интерфейса и не перекрывают важную информацию на экране.

Тестирование пользовательского интерфейса (UI) позволяет оценить, насколько удобно и понятно пользователю взаимодействовать с уведомлениями. Здесь необходимо проверить, что уведомления видны на экране и не вызывают проблем с восприятием основного контента. Важно протестировать адаптивность системы уведомлений на разных устройствах и экранах, убедиться, что уведомления корректно отображаются как на мобильных, так и на настольных версиях приложения. Пользовательские тесты могут включать проверку визуального стиля уведомлений, таких как цветовые схемы, шрифты и анимации, чтобы убедиться, что они соответствуют общему дизайну и не вызывают дискомфорта у пользователей.

Функциональное тестирование – это вид тестирования, который сосредоточен на проверке соответствия программного обеспечения заданным функциональным требованиям. Основная цель функционального тестирования – убедиться, что программное обеспечение работает в соответствии с ожиданиями пользователей и требованиями бизнеса, выявляя любые несоответствия или дефекты на ранних этапах разработки.

Заключение

В ходе прохождения производственной практики была выполнена поставленная цель, а именно овладение профессиональными компетенциями по модулям ПМ.03 «Проектирование и разработка информационных систем» и ПМ.05 «Сoadминистрирование и автоматизация баз данных и серверов». Также были выполнены следующие задачи:

- ознакомлен с предприятием;
- ознакомлен с перечнем и конфигурацией средств вычислительной техники, а также архитектурой сети. Ознакомление с перечнем и назначением программных средств;
- описана база данных предприятия;
- анализ и оценка предметной области и методов определения стратегии развития организации;
- предложение методов автоматизации процессов для повышения производительности предприятия;
- описана информационная система, предложенной для внедрения;
- разработка плана мероприятий для этапов проектирования, разработки, тестирования и внедрения информационной системы;
- анализ и выбор инструментальных средств для проектирования и разработки информационной системы;
- проектирование информационной системы для предприятия;
- разработка информационной системы для предприятия;
- оформление тестовой документации с тест-кейсами для основной функциональности информационной системы;
- тестирование разработанной системы. Исправление возникших ошибок.

Пройденная производственная практика успешно закрепила знания, полученные во время обучения, а также улучшила их.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Очно.рф – информация о производственной практике, решение вопросов, связанных с учебной и производственной практиками. Имеется форум для вопросов: <https://xn--m1abb8a.xn--p1ai/uchebnaya-praktika/> (дата обращения: 04.06.2024).

2 РСТ-ТУР – информация о предприятии в целом: <https://rsttur.ru/about> (дата обращения: 05.06.2024).

3 Хабр – русскоязычный веб-сайт в формате системы тематических коллективных блогов с элементами новостного сайта, созданный для публикации новостей, аналитических статей: <https://habr.com/ru/all/> (дата обращения: 06.06.2024).

4 Figma – инструмент для дизайна интерфейсов и их редактирования. На сайте содержится информация о приложении, инструкции, решения: <https://www.figma.com/> (дата обращения: 07.06.2024).

5 Фабио С. FIGMA. Проектирование и прототипирование интерфейсов в соответствии с принципами UX/UI / Издательство ДМК Пресс, 2024 – 370 с. – ISBN 978-5-93700-302-7 – Текст: электронный. – URL: <https://www.ozon.ru/product/figma-proektirovanie-i-prototipirovanie-interfeysov-v-sootvetstvii-s-printsipami-ux-ui-1549295097/?asb=yXL6TjyW> (дата обращения: 08.06.2024).

6 Adobe – инструменты для создания и редактирования изображений, создания интерфейсов. На сайте содержится множество продуктов компании Adobe, в том числе Adobe Illustrator, а также другие разделы сайта: <https://www.adobe.com/> (дата обращения: 09.06.2024).

7. Тучкевич Е. И. Adobe Illustrator CC 2022. Мастер-класс / Издательство БХВ Санкт-Петербург, 2023 – 250 с. – ISBN 978-5-9775-1184-1 – Текст: электронный и печатный. – URL: <https://www.ozon.ru/product/adobe-illustrator-cc-2022-master-klass-evgenii-tuchkevich-tuchkevich> (дата обращения: 10.06.2024).

8. Output – информация о webpack output: <https://webpack.js.org/configuration/output/> (дата обращения: 11.06.2024).

9. Sweetalert 2.io – документация всплывающих уведомлений и CSS стилей для них: <https://sweetalert2.github.io/> (дата обращения: 12.06.2024).

10. Хавербеке М. Выразительный JavaScript. Современное веб-программирование/ Издательство Питер, 2022 г. – 337 с. – ISBN 978-5-4461-1226-5. – Текст: электронный. – URL: <https://www.ozon.ru/product/vyrazitelnyu-javascript-sovremennoe-veb-programmirovanie-pdf-erub-haverbeke-mareyn-935402598/> (дата обращения: 13.06.2024).

Вид профессиональной деятельности	Код и формулировка формируемых профессиональных компетенций	Виды работ, выполненных обучающимся во время практики в рамках овладения компетенциями	Качество выполнения работ (отлично, хорошо, удовлетворительно, неудовлетворительно)
ПМ.05 Сoadминистрирование и автоматизация баз данных и серверов	ПК 7.1 Выявлять технические проблемы, возникающие в процессе эксплуатации баз данных и серверов	Описать базу данных предприятия. Разработать дополнительные политики безопасности для базы данных	Отлично
	ПК 7.2 Осуществлять администрирование отдельных компонентов серверов	Настроить резервное копирование данных для базы данных	Отлично
	ПК 7.3 Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов	Формировать необходимые для работы информационной системы требования к конфигурации локальных компьютерных сетей.	Отлично
	ПК 7.4 Осуществлять администрирование баз данных в рамках своей компетенции	Настроить резервное копирование данных для базы данных информационной системы.	Отлично
	ПК 7.5 Проводить аудит систем безопасности баз данных и серверов, с использованием регламентов по защите информации	Обеспечить базовую безопасность базы данных.	Отлично
Итоговая оценка по ПМ.05 Сoadминистрирование и автоматизация баз данных и серверов			Отлично

Заключение об уровне освоения обучающимися профессиональных компетенций:

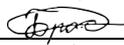
Освоены на базовом уровне

(освоены на продвинутом уровне / освоены на базовом уровне / освоены на пороговом уровне / освоены на уровне ниже порогового)

Дата 22 июня 2024 г.

Оценка за практику Отлично

Руководитель практики от предприятия



подпись

Бросалин В.В.

Ф.И.О.

М.П.



ДНЕВНИК

прохождения производственной практики

Студент Погожев Герман Сергеевич

Фамилия Имя Отчество

Специальность 09.02.07 «Информационные системы и программирование»

Группа СО-ИП-21-2

Место прохождения практики ООО «РСТ-ТУР»

наименование предприятия

Сроки прохождения с 03.06.2024 г. по 22.06.2024 г.

Инструктаж на рабочем месте

«03» июня 2024 г.

дата


подпись

Бросалин В.В.

Ф.И.О. инструктирующего

Дата (период)	Описание выполнения производственных заданий (виды и объем работ, выполненных за день)	Оценка	Подпись руководителя практики
03.06.2024	Прохождение инструктажа по охране труда и технике безопасности, установочная лекция. Анализ деятельности предприятия	Отлично	
04.06.2024	Ознакомление с предприятием.	Отлично	
05.06.2024 – 06.06.2024	Ознакомление с перечнем и конфигурацией средств вычислительной техники, а также архитектурой сети. Ознакомление с перечнем и назначением программных средств, установленных на ПК предприятия.	Отлично	
07.06.2024	Описание базы данных предприятия.	Отлично	
08.06.2024 – 10.06.2024	Работа с информационной системой предприятия. Анализ и оценка предметной области.	Отлично	
11.06.2024	Предложение методов автоматизации процессов. Описание информационной системы. Анализ существующих информационных систем по выбранной тематике. Разработка планов мероприятий для проектирования.	Отлично	
12.06.2024	Анализ и выбор инструментальных средств для проектирования и разработки.	Отлично	
13.06.2024 – 14.06.2024	Проектирование информационной системы.	Отлично	
15.06.2024 – 18.06.2024	Разработка информационной системы для предприятия.	Отлично	
19.06.2024	Оформление текстовой документации с тест-кейсами.	Отлично	
20.06.2024	Тестирование разработанной системы.	Отлично	
21.06.2024 – 22.06.2024	Обобщение материалов практики, оформление отчета и необходимых документов по практике.	Отлично	

Руководитель практики от предприятия



подпись

Бросалин В.В.

Ф.И.О.



П.

ХАРАКТЕРИСТИКА

о прохождении производственной практики студента

Студент Погожев Герман Сергеевич СО-ИП-21-2

ФИО студента

группа

проходил практику с «03» июня 2024 г. по «22» июня 2024 г.

на базе ООО «РСТ-ТУР», 68000, Хабаровский край, г. Хабаровск, ул. Ленина, д. 18В

наименование организации, юридический адрес

в подразделении ООО «РСТ-ТУР», г. Хабаровск, ул. Ленина, д. 18В

название подразделения

За период прохождения практики студент посетил 18 дней, из них по уважительно причине отсутствовал 0 дней, пропуски без уважительной причины составили 0 дней.

Студент соблюдал трудовую дисциплину и/или правила техники безопасности.

Отмечены следующие нарушения трудовой дисциплины и/или правил техники безопасности: -

Студент не справился со следующими видами работ: -

За время прохождения практики студент показал, что умеет планировать и организовывать собственную деятельность, способен налаживать взаимоотношения с другими сотрудниками, имеет хороший уровень культуры поведения, умеет работать в команде, высокая степень сформированности умений в профессиональной деятельности.

В отношении выполнения трудовых заданий проявил себя ответственно и внимательно.

В рамках дальнейшего обучения и прохождения производственной практики студенту можно порекомендовать: Углублять свои знания в области информационных технологий, а также самостоятельно осваивать новые инструменты и технологии разработки программных средств.

Директор

Должность наставника/куратора

Бросалин

подпись

Бросалин В.В.



М.П.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владивостокский государственный университет»
(ФГБОУ ВО «ВВГУ»)

НАПРАВЛЕНИЕ НА ПРАКТИКУ

Студент Погожев Герман Сергеевич

Подразделение Академический колледж Группы СО-ИП-21-2

согласно приказу ректора № _____ от _____ года

направляется в ООО "РСТ-ТУР", г. Хабаровск

для прохождения производственной (по профилю специальности) практики по
специальности 09.02.07 «Информационные системы и программирование» на срок 3
недели с 03.06.2024 года по 22.06.2024 года.

Руководитель практики Атабасов Д.А.



Отметки о выполнении и сроках практики

Наименование предприятия	Отметка о прибытии и убытии	Печат
ООО "РСТ ТУР"	03.06.2024	Бра
ООО "РСТ ТУР"	22.06.2024	Бра

