**РЕФЕРАТ**

БАЗА ДАННЫХ, ЦЕЛОСТНОСТЬ ДАННЫХ, ФИЗИЧЕСКАЯ МОДЕЛЬ, ЛОГИЧЕСКАЯ МОДЕЛЬ, ER-ДИАГРАММА, ВЕБ-ПРИЛОЖЕНИЕ.

Объектом исследования (разработки) является деятельность администраторов, работников, менеджеров, заказчиков.

Цель проекта – разработка веб-системы для удобного и эффективного управления проектами.

В процессе работы (проектирования) выполнены следующие исследования (разработки): дизайн сайта, моделирование и реализация пользовательского интерфейса.

Элементами научной новизны (практической значимости) полученных результатов являются осуществление методов эффективной и удобной организации работы над проектами, возможность обсуждения вопросов, касающихся рабочего процесса между участниками проекта, прозрачность рабочего процесса, возможность получения информации о всех сотрудниках, участвующих в разработке проекта.

Областью возможного практического применения являются компании и отдельные лица, занимающиеся разработкой проектов в команде.

В ходе дипломного проектирования прошли апробацию такие предложения, как использование ReactJs для написания пользовательского интерфейса используя компонентный подход, и придания ему динамичности, Webpack для удобной организации проекта и его сборки, использование HTML и CSS для верстки и стилизации страниц веб-приложения, библиотека Material UI для ускорения и упрощения стилизации приложения.

Результатами внедрения является веб-приложение, позволяющее оптимизировать работу администраторов, работников, менеджеров и заказчиков. Обеспечить удобное и эффективное управление проектами и работу в команде.

Студент-дипломник подтверждает, что приведенный в дипломном проекте расчетно-аналитический материал объективно отражает состояние исследуемого процесса (разрабатываемого объекта), все заимствованные из литературных и других источников теоретические и методологические положения и концепции сопровождаются ссылками на их авторов.

Дипломный проект: 63 с., 22 рис., 13 табл., 13 источник, 1 прил.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc9175381)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc9175382)

[2 ПОСТАНОВКА ЗАДАЧИ 9](#_Toc9175383)

[2.1 Общие определения 9](#_Toc9175384)

[2.2 Требования к программному продукту 11](#_Toc9175385)

[3 ПРОЕКТИРОВАНИЕ И МОДЕЛИРОВАНИЕ 12](#_Toc9175386)

[3.1 Логическая модель 12](#_Toc9175387)

[3.1.1 Основные роли в системе 12](#_Toc9175388)

[3.1.2 Варианты использования системы 12](#_Toc9175389)

[3.3 Моделирование пользовательского интерфейса 17](#_Toc9175390)

[4 РЕАЛИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА 20](#_Toc9175391)

[4.1 Интегрированная среда разработки Visual Studio Code 20](#_Toc9175392)

[4.1 Визуальная часть 21](#_Toc9175393)

[4.2 Программная часть 21](#_Toc9175394)

[4.2.1 JavaScript (ReactJs) 22](#_Toc9175395)

[4.2.2 NPM 23](#_Toc9175396)

[4.2.3 Webpack 24](#_Toc9175397)

[4.2.4 React Router 24](#_Toc9175398)

[4.2.5 Material UI 25](#_Toc9175399)

[4.2.6 HTML 28](#_Toc9175400)

[4.2.7 CSS 3 (SASS) 29](#_Toc9175401)

[5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 31](#_Toc9175402)

[6 ТЕСТИРОВАНИЕ ПРОРАММНОГО ОБЕСПЕЧЕНИЯ 36](#_Toc9175403)

[6.1 Критическое тестирование 36](#_Toc9175404)

[6.2 Углубленное тестирование 39](#_Toc9175405)

[7 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО СРЕДСТВА 42](#_Toc9175406)

[7.1 Расчет сметы затрат, цены и прибыли на программное средство 42](#_Toc9175407)

[***6.1.1 Исходные данные*** 43](#_Toc9175408)

[Обозначения 56](#_Toc9175409)

[***5.1 Производственная санитария, техника безопасности и пожарная профилактика*** 63](#_Toc9175410)

[***5.2 Требования к ВДТ, ЭВМ, ПЭВМ и периферийным устройствам*** 70](#_Toc9175411)

[ЗАКЛЮЧЕНИЕ 72](#_Toc9175412)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 3](#_Toc9175413)

# ВВЕДЕНИЕ

В наши дни сложно представить себе компанию в той или иной сфере, которая ведет проекты без системы управления проектами. В условиях командной работы часто приходится планировать и распределять задачи между членами команды, а также оптимизировать рабочий процесс. Устная постановка задач, в условиях даже небольшого проекта, неудобна в использовании. На проектах, где работу выполняют несколько человек, сложно запомнить какие задачи необходимо выполнить, и кто закреплен за ними, обеспечив при этом прозрачность. Кроме того, у работника может часто возникать необходимость переключаться между задачами или даже проектами.

Зачастую вначале проектирования сложно предвидеть всю архитектуру проекта и учесть ряд прошлых ошибок, что может плохо отразиться на качестве и времени выполнения поставленной цели. Выше перечислены лишь немногие проблемы, которые могут возникать в процессе разработки некого продукта. Источником подобных проблем является то, что принятые решения, описания требований и того что должен представлять из себя конечный продукт нигде не фиксируется.

Из этого можно сделать вывод, что стоит ввести письменную документацию, но за этим может так же последовать ряд проблем. Требования, описанные в письмах или сообщениях, иногда затруднительно собрать воедино и можно просто забыть о них. Так же могут возникать трудности с анализом прошлых ошибок и внесением корректировок в текущие задачи.

Лучшим решением данного вопроса является автоматизация управленческой деятельности, которая может быть реализована в виде системы управления проектами.

Целью данного дипломного проекта является решение вышеупомянутых вопросов и создание системы для ведения документации, управления задачами и анализа проблем в разработке проекта, а также возможности коммуникации между участниками одного проекта.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Система управления проектами – является комплексом технологических и организационных методов и инструментов, поддерживающих управление проектами и повышающих эффективности их реализации. Такая система обычно соединяет в себе ряд других приложений, таких как:

1. планирование задач;
2. составление расписания;
3. распределение совместной работы;
4. общения;
5. документирование и администрирование системы.

Система управления проектами повышает эффективность командной работы, качество проектного менеджмента руководителями проекта. Предоставляет возможность планировать, выставлять приоритеты задач, обсуждать задачи по средствам комментариев между командой, просматривать историю выполненных задач.

Задачи систем управления проектами:

1. Обеспечение проект-менеджера необходимыми инструментами планирования и контроля процесса реализации проекта;
2. Предоставление участникам проекта понятных инструментов для решения задач и доступа к соответствующей информации;
3. Предоставление руководителям подразделений инструментария по контролю загрузки исполнителей проектных и непроектных задач и информации для принятия решений о назначении сотрудников на новые проекты и перераспределении нагрузки;
4. Снабжение директора удобными инструментами, позволяющими автоматизировать рутинные операции и устанавливать контроль состояния всего портфеля проектов и качества работы руководителей каждого проекта;
5. Обеспечение руководителя проекта целостной моделью мониторинга портфеля проектов и анализа принимаемых решений и сопутствующих отклонений;
6. Обеспечение акционеров компании инструментом для мониторинга соответствия проектного портфеля стратегическим целям компании.

На мировом рынке существует несколько программных решений по управлению проектами:

Jira – это коммерческая система отслеживания ошибок, предназначенная для организации взаимодействия с пользователями и управления проектами. Система разработана компанией Atlassian в 2002 году и является одним из двух ее основных продуктов (наряду с вики-системой Confluence). Веб-интерфейс предоставляет возможности по созданию задач, отслеживанию процесса работы над проектом и многое другое.

Преимущества:

1. широкий функционал;
2. стабильность работы;
3. наличие мобильного приложения.

Недостатки:

1. широкий функционал и сложный интерфейс так же делает систему сложной по порогу вхождения;
2. высокая стоимость;
3. необходимость иметь специального администратора, который должен обслуживать пользователей в случае возникновения проблем;
4. проблемы, связанные с интеграцией с другими сервисами, например, Битрикс.

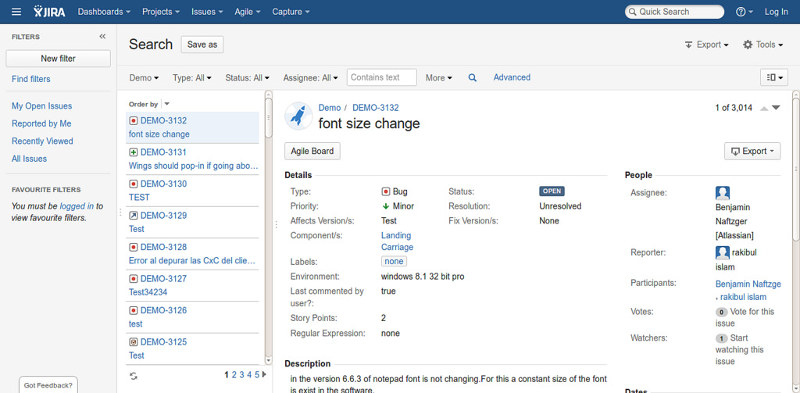
На рисунке 1.1 приведен пользовательский интерфейс, который отражает список задач и детальное описание выбранной задачи.

Рисунок 1.1 – Пользовательский интерфейс системы управления проектами Jira

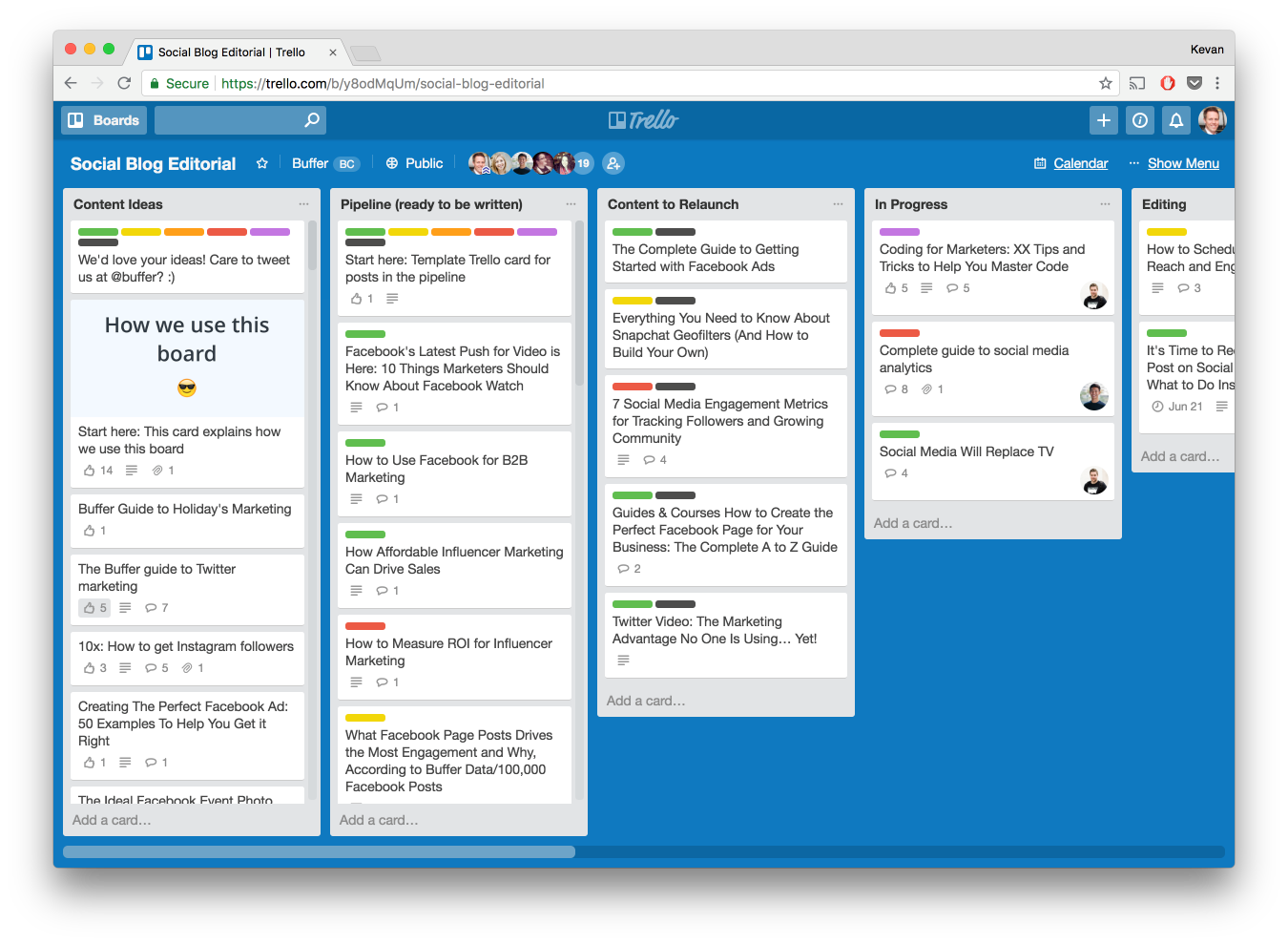
Trello – программа для управления проектами небольших групп. Система построена по японскому принципу управления «канбан», который предполагает последовательный контроль за этапами производства. В Trello систематизация осуществляется с помощью досок, которые можно увидеть на рисунке 1.2. Это могут быть различные проекты или процессы в компании. Доска состоит из списков – вертикальных колонок, расположенных слева направо, которые показывают этапы производства, от идей до реализации.

Рисунок 1.2 – Доски со списками задач в пользовательском интерфейсе менеджере задач Trello

В силу своей простоты Trello имеет ряд преимуществ:

1. интуитивно понятный и привлекательный интерфейс;
2. удобен для небольших команд и ежедневного планирования бытовых задач;
3. сервис условно бесплатен – основной функционал позволяет работать почти без ограничений.

Недостатки:

1. не подходит для глобальных задач и работы над крупными проектами, из-за упрощенного функционала;
2. нельзя отслеживать аналитику и формировать отчеты;
3. административный контроль и другие улучшения доступны в платной версии программы;
4. низкий уровень безопасности.

# 2 ПОСТАНОВКА ЗАДАЧИ

### 2.1 Общие определения

Пользователь – информация, связанная с реальным человеком, имеющим доступ к использованию данной системы. Набор функционала, которым может воспользоваться пользователь, зависит от назначенных ему прав доступа. Для различных проектов пользователь может иметь различные права доступа.

Права доступа – разрешение на определенные действия в рамках проекта для конкретного пользователя. В отношении проектов существуют следующие виды прав доступа:

1. просмотр задач – возможность просматривать основной информации по проектным задачам;
2. управление задачами – возможность создания и редактирования задач;
3. управление проектом – возможность изменять имя, описание и другие параметры проекта.

Администратор – специальный пользователь, который отвечает за управление пользовательскими аккаунтами в системы, выполняет конфигурацию системы на глобальном уровне (в отношении всех проектов) и имеющий непосредственный доступ к базе данных.

Проект – может означать идею, описание цели или чего-либо другого, по отношению к чему могут выполняться какие-либо задачи.

Задача – означает конкретную единицу работы, которую нужно выполнить конкретным исполнителем по отношению к определенному проекту.

Подзадача – дочерняя задача, которая необходима для детализации или разбиения некоторой задачи на несколько подзадач (подзадачи могут быть созданы для любого типа задачи кроме подзадачи).

Нововведение – новая задача, новая функция или новое свойство системы – задача, связанная с реализацией чего-то нового.

Дефект – задача, связанная с исправлением определенных проблем.

Иная задача – абстрактный тип задачи, обычно используется для тех случаев, когда другие типы задач не подходят.

Исполнитель – конкретный пользователь, ответственный за выполнение задачи.

Приоритет – приоритет задачи. Означает важность этой задачи по отношению к другим. Виды приоритетов:

1. тривиальный – означает минимальный приоритет задачи (например, изменение цвета кнопки, сортировка документов и т.п.);
2. низкий – более высокий приоритет, но все еще не критичный к исполнению (результат работы над проектом может отдаваться даже в случаях, когда не выполнены некоторые задачи этого приоритета);
3. высокий – означает высокий приоритет и важность выполнения этой задачи для конечного клиента/заказчика;
4. критический – означает высокий приоритет и первоочередность выполнения такой задачи;
5. блокирующий – также высокий приоритет, отличающийся от других подобных тем, что задача с этим приоритетом будет препятствовать работе других участников команды.

Предполагаемое время – запланированное время необходимое для выполнения задачи.

Затраченное время – фактически потраченное время на выполнение задачи.

Статья – документ, содержащий некоторую информацию о проекте в целом или о какой-либо задаче.

Статус – означает стадию, на котором находится задача. Статус задачи может быть следующих видов:

1. открыта – означает, задача создана и ей присвоен исполнитель;
2. выполняется – исполнитель принял задачу и находится в процессе ее выполнения;
3. тестирование – задача была выполнена исполнителем и сейчас находится в процессе тестирования;
4. завершена – задача была протестирована, ошибок не выявлено;
5. верификация – выполненная и протестированная задача находится на стадии проверки более квалифицированными работниками, для определения правильности ее решения, с точки зрения логичности применения тех или иных методов для ее исполнения.

### 2.2 Требования к программному продукту

Необходимо разработать систему для реализации удобного и эффективного управления проектами. Основные возможности:

1. планирование различных событий, зависящих друг от друга;
2. идентификация крупных составных частей проекта (вехи проекта) и их декомпозиция, посредством которой создается структура декомпозиции работ, также называемая иерархической структурой работ (англ. work break-down structure — WBS);
3. планирование расписания работы сотрудников и назначение ресурсов на конкретные задачи;
4. расчет времени, необходимого на решение каждой из задач;
5. сортировка задач в зависимости от сроков их завершения;
6. презентация графика работ по проекту в виде диаграммы Гантта[2];
7. управление несколькими проектами одновременно.
8. управление списком задач для сотрудников и предоставление информации по распределения ресурсов;
9. обзор информации о сроках выполнения задач;
10. возможность раннего предупреждения о возможных рисках, связанных с проектом;
11. обзор информации о рабочей нагрузке;
12. предоставление информации о ходе проекта, показатели и их прогнозирование.
13. обсуждение и согласование рабочих вопросов проекта;
14. фиксация проблем проекта и запросов на изменения, их обработка;
15. ведение рисков проекта и управление ими.

Рабочее название системы - Ziro. Система должна быть реализована в виде веб-системы в виду того, что это наиболее предпочтительный вариант для совместного использования в рамках большого количества людей (работники, менеджер, заказчики и т.п.).

# 3 ПРОЕКТИРОВАНИЕ И МОДЕЛИРОВАНИЕ

### 3.1 Логическая модель

### 3.1.1 Основные роли в системе

В системе должны быть определены некоторые логические группы пользователей, при попадании в которые пользователь будет иметь возможность пользоваться определенным набором функционалом. Отношение пользователя к той или иной группе будет означать то, что пользователь имеет определенную роль в системе.

В системе должны быть определены две основные роли пользователей:

1. пользователь (User) – наделенные данной ролью пользователи системы могут пользоваться основным функционалом системы достаточным для осуществления деятельности по управлению проектами;
2. администратор (Administrator) – данная роль присваивается пользователем, ответственным за создание новых пользователей в системе, назначение им ролей, управление доступом пользователей к определенным проектам, а также создание проектов в системе.

Между данными двумя ролями существует большая разница как в контексте использования функционала системы, так и в целом отношения к главной деятельности – управлению проектами.

### 3.1.2 Варианты использования системы

Пользователи, имеющие роль администратор по своей сути не имеют никакого отношения к управлению проектами. Их главная обязанность – введение новых пользователей и проектов в систему, а также разграничение прав доступа. Также они обязаны разрешать любые вопросы, касающиеся работоспособности системы, и восстанавливать учетные записи для пользователей с ролью «User».

Таким образом, для администраторов должен быть предусмотрен и доступен следующий функционал в системе:

1. создание первоначальной учетной записи для пользователя;
2. назначение роли для пользователя
3. создание проекта;
4. назначение для пользователя проектов, с которыми он может работать;
5. лишения доступа к определенным проектам для определенного пользователя.

В свою очередь пользователи, относящиеся к роли «User», не могут никаким образом воздействовать на учетные данные других пользователей либо на данные по проектам, но могут осуществлять основную деятельность в отношении проектов. Т.е. пользователь с ролью «User» этот тот самый человек, который имеет прямое отношение к разработке и развитию программного проекта. Это может быть, как программист, ведущий разработку на проекте, так и проектный руководитель, организующий работу среди подчиненных, или же тестировщик, обеспечивающий качество выполненных работ, и др. При этом данные пользователи могут вести свою деятельность только в отношении тех проектов, к которым им был дан доступ администратором системы, т.е. они ничего не будут знать о других проектах в системе и любой относящейся к ним информации. В свою очередь, когда пользователю назначается какой-то проект, он получает полный доступ ко всему функционалу и информации в отношении проекта и становится частью команды (группа людей, относящихся к одному и тому же проекту).

Таким образом, для осуществления своей деятельности, пользователям с данной ролью необходим следующий функционал в системе:

1. редактирование профиля с заданием всех ключевых персональных данных: должность, контакты, фото и т.д;
2. просмотр информации о членах команды, включая их контакты;
3. получение списка текущих задач, всех задач в проекте, закрытые задачи в прошлом для определенного проекта, а также получение списка задач по множеству другим критериям для формирования статистики, изучения предыдущего опыта и для других целей;
4. создание задачи с указанием всех основных параметров: название, описание, сроки, тип, статус и другие данные;
5. возможность редактирования основных данных по задаче (за исключением некоторых данных, таких как номер задачи, создатель, дата создания и т.д.);
6. назначение исполнителя для определенной задачи;
7. возможность обсуждения задачи с остальными членами команды;
8. ведение и просмотр журнала работ по конкретной задаче;
9. просмотр информации по текущим проектам;
10. возможность загрузки документации, относящейся к определенному проекту.

Варианты использования описанного функционала для каждой роли пользователей отображено в приложении.

***3.2 Общая архитектура системы***

Прежде чем приступить к разработке конечного программного обеспечения, необходимо определиться с его типом и архитектурой.

Принимая во внимание критерии, определенные в постановке задачи, можно определить следующее:

В контексте работы с данными разрабатываемое программное обеспечение должно удовлетворять критериям доступности и согласованности это означает то, что каждое пользовательское приложение не может иметь свое собственное хранилище данных, т.к. данные используются совместно с другими пользователями и пришлось бы тратить множество ресурсов на поддержание согласованности данных во всех хранилищах всех пользователей и всегда. Из этого следует вывод, что должна существовать единая база данных, доступ к которой должны иметь приложения пользователей. На данном этапе архитектура разрабатываемого приложения выглядело бы так, как показано на рисунке 3.1



Рисунок 3.1 – Архитектура приложения, взаимодействующего с базой данных

Но данная архитектура не удовлетворяет другим определенным критериям:

1. простота – приложение не получится использовать сразу же после ее установки, т.к. нужно будет потратить время на ее конфигурацию (работа с базой данных, настройки безопасности, другие настройки необходимые для сторонних сервисов и обработки данных);
2. легковесность – приложение будет тратить большое количество ресурсов пользовательской машины, т.к. все операции по обработки данных будут осуществляться на ней, так же должны будут установлены все компоненты по взаимодействию со сторонними сервисами и базой данных, что значительно увеличит размер приложения;
3. безопасность – текущая архитектура приложения увеличивает риск доступа к базе данных злоумышленником;
4. доступность – пользователи должны иметь возможность работать отовсюду, это значит, что и база данных должна быть доступна всегда.

Для удовлетворения всех вышеперечисленных требований необходимо разбить компонент «Приложение пользователя» на два других программных обеспечения: клиентское приложение и сервер.

Дополнительное звено «Сервер» в архитектуре необходимо для соблюдения вышеописанных критериев системы за счет следующих характеристик:

1. доступ к базе данных – только сервер будет иметь прямой доступ к базе данных, таким образом никаких сведений о базе данных не будет храниться на клиентских приложениях, что повышает безопасность доступа к данным;
2. обработка – сервер отвечает за обработку всех запросов, поступающих с клиентских приложений, таким образом, клиентское приложение больше не будет выполнять никаких затратных действий, а также будет иметь значительно меньший размер и сложность.

Текущая архитектура системы представлена на рисунке 3.2.



Рисунок 3.2 – Архитектура клиент-серверной системы с доступом к базе данных

Существует лишь один способ обеспечения абсолютной доступности сервера для клиентских приложений – осуществления их взаимодействия в сети Интернет. Сервер, работающий в соответствии со стандартами и по протоколам определенными в сети интернет называется веб-сервером. С другой стороны, клиентские приложения должны устанавливать соединение с сервером и посылать запросы в правильном формате**.** Теперь система будет содержать три звена: веб-сервер, клиентское веб-приложение и базу данных. Новая архитектура системы представлена на рисунке 3.3.



Рисунок 3.3 – Архитектуры веб-системы

Следует отметить, что база данных изолирована локальной системой, поэтому к базе данных никто не сможет получить доступ кроме веб-сервера.

На данном этапе все основные критерии соблюдены, остаются некоторые проблемы с разворачивание клиентского приложения на машине пользователя и использования его на различных устройствах.

Предполагается, что пользователь будет использовать приложение на мобильном устройстве и на компьютере, при этом операционная система может быть любой. Для того чтобы удовлетворить пользователей компьютеров и ноутбуков уже существует готовое решение, которое независимо от операционной системы, знакомо каждому практически каждому пользователю и работающее в Интернете – браузер.

Для пользователей мобильных устройств можно было бы тоже использовать браузер, но это не лучшее решение в контексте разработки пользовательского интерфейса сразу для двух категорий пользователей. Также существует и другие причины в пользу создания отдельного мобильного приложения (больше возможностей по созданию графического интерфейса, адаптация под разные размеры экранов самая лучшая и др.). Таким образом, для пользователей мобильных устройств необходимо создать отдельное мобильное веб-приложение.

В результате клиентская часть системы делится на два типа: использование браузера пользователями компьютера и использование мобильного приложения пользователями мобильных устройств. Финальная архитектура разрабатываемой системы представлена на рисунке 3.4.



Рисунок 3.4 – Архитектура веб-системы с несколькими типами клиентских приложений

На данном этапе все требования к архитектуре удовлетворены, а значит она и будет являться основой для разрабатываемой системы.

### 3.3 Моделирование пользовательского интерфейса

Перед разработкой пользовательского интерфейса необходимо спроектировать макет веб-страницы. Учитывая критерии, приведенные в постановке задачи, каждая веб-страница должна содержать следующую структуру:

1. верхнюю часть сайта, в которой будет находиться логотип, меню и аватар (фото) пользователя. Меню должно содержать навигацию по веб-приложению;
2. главная секция должна содержать основную информацию приложения, с которой будет взаимодействовать пользователь. Содержимое главной секции будет изменяться в зависимость от той страницы, на которой находится пользователь;
3. нижняя часть сайта должна носить информативный характер и содержать данные о защите авторских прав.

Веб-приложение должно содержать несколько страниц. Главная страница будет содержать список задач, текущего авторизованного пользователя и кнопку, которая позволяет создать новую задачу. Макет главной страницы представлен на рисунке 3.1.



Рисунок 3.1 – Макет главной страницы

При выборе определенной задачи должно происходить перенаправление на другую страницу, на которой будет детальное описание этой задачи. Так же будет предусмотрена возможность управления состоянием задачей по средствам кнопок. На странице должна быть предусмотрена возможность комментирования, благодаря которой у участников проекта будет возможность задавать вопросы автору задачи относительно текущей задачи или проекта в целом. Так же должен быть предусмотрен вывод информация о ходе работы над задачей, который будет отражаться в журнале работ. Макет детализации задачи приведен на рисунке 3.2.



Рисунок 3.2 – Макет страницы детализации задачи

Страница проектов должна содержать таблицу с проектами и окном поиска для возможности быстрого нахождения нужного проекта. При клике по строке с проектом, пользователь будет переходить на страницу с детальным описанием проекта. На странице проекта можно будет скачать или загрузить текстовые-документы, относящиеся к этому проекту. Страница с таблицей проектов и страница детализации проекта должны быть схожи по визуальному оформлению и логике выполняемых действий со страницами списка задач и детализации задачи.

Страница личного профиля пользователя должна загружаться при клике по фото пользователя в верхней части сайта. На этой странице должна размещаться достаточная информация о работнике: имя, фамилия, должность, контактные данные, навыки и другое. Должна быть предусмотрена возможность редактирования профиля. Так же из личного кабинета будет предоставлена возможность выхода из аккаунта. Макет страницы личного профиля пользователя представлен на рисунке 3.3.



Рисунок 3.3 – Макет страницы личного профиля пользователя

Страница входа должна быть представлена простой формой с полями для ввода email и пароля пользователя, а также кнопкой войти.

В случае если текущий пользователь имеет роль – администратор, в меню скрываются некоторые пункты и добавляются новые, соответствующие роли. Администратору будет доступна страница с таблицей проектов, которая будет схожа с одноименной страницей для роли «Пользователь». Так же будет доступна страница создания нового пользователя. Данная страница будет содержать форму с полями для ввода информации о пользователе: имя, фамилия, должность, проект, email. Типичная форма для ввода информации представлена на рисунке 3.4.



Рисунок 3.4 – Макет формы для внесения информации

Таким образом…

# 4 РЕАЛИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

### 4.1 Интегрированная среда разработки Visual Studio Code

Visual Studio Code – редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом. Visual Studio Code поддерживает многие языки программирования, например: С, C++, C#, CSS, HTML, JavaScript, Json, React, PHP и другие. Скриншот интерфейса среды разработки приведен на рисунке 4.1.



Рисунок 4.1 – Интерфейс среды разработки Visual Studio Code

Основными преимуществами для выбора данной среды разработки послужили:

1. бесплатная – полноценная версия программы доступна на официальном сайте;
2. легковесность, что выделяет Visual Studio Code над средой разработки того же семейства – Visual Studio;
3. встроенная консоль, которая позволяет избавиться от переключения между окнами и сконцентрироваться на пространстве редактора кода;
4. возможность кастомизации – позволяет настроить редактор кода под свои нужды, скачивать и устанавливать различные темы и наборы иконок;
5. поддержка различных плагинов, позволяющих ускорить разработку приложений.

### 4.1 Визуальная часть

Пользовательский интерфейс системы управления проектами “ZIRO” выдержан в минималистичном стиле с элементами современного material дизайна и представляет собой одностраничное веб-приложение (SPA) с динамически изменяющимся контентом. Страница состоит из трех основных секций:

1. шапка – верхняя часть сайта, содержащая в себе логотип меню и аватар (фото) пользователя. Эта секция изменяется лишь в зависимости от роли авторизованного пользователя;
2. главная секция включает в себя основной контент сайта, с которым взаимодействует пользователь. Контент главной секции изменяется при переходе по ссылкам или при открытии всплывающих окон;
3. подвал – нижняя часть сайта. Содержит копирайт – защита авторских прав в интернете. Эта секция является статичной, и не изменяется ни при каких условиях.

### 4.2 Программная часть

Пользовательский интерфейс веб-приложения написан с использованием современные технологии веб-разработки:

1. JavaScript (ReactJs)
2. Npm
3. Webpack и Babel
4. Router Dom
5. Material UI
6. HTML
7. CSS 3 (SASS)

### 4.2.1 JavaScript (ReactJs)

JavaScript – это язык программирования, который изначально задумывался, как язык для взаимодействия с HTML-документом, который добавлял бы интерактивность на веб-сайт. Сейчас JavaScript единственный язык программирования для браузеров. Он работает под Windows, macOS, Linux, мобильных платформах, одним словом везде.

JavaScript стремительно развивается и сейчас это уже не только язык интерактивности веб-страниц, с помощью него можно взаимодействовать пользователями, получать и отправлять запросы на сервер, а также писать полноценные приложения благодаря Node.js.

Для написания дипломного проекта была выбрана библиотека ReactJs. Это инструмент с открытым исходным кодом для создания пользовательских интерфейсов, разработанный Facebook в 2013 году и поддерживаемый им в настоящее время. Сейчас ReactJs пользуется большой популярностью благодаря своей лаконичности и легкости интеграции с существующими проектами. ReactJs представляет собой полноценный View модуль (Представление) из шаблона проектирования MVC (Модель-Представление-Контроллер), что подтверждает его независимость от серверной части проекта.

ReactJs использует компонентный подход. Компонент – это отдельная часть интерфейса, которую можно переиспользовать, наследовать, компоновать. Технически компонент строится с помощью функции либо класса. Для формирования веб-страницы происходит чтение и отрисовка компонентов в определенном порядке. Обычно ряд компонентов собирается в одном корневом компоненте и последний передается на отрисовку на странице.

Для решения проблем производительность ReactJs использует виртуальный DOM (Document Object Model) – организацию элементов html, которыми можно манипулировать, изменять, удалять или добавлять новые. Для взаимодействия с DOM применяется язык JavaScript. Манипуляции html-элементами с помощью JavaScript может привести к снижению производительности, особенно при изменении большого количества элементов. А операции над элементами могут занять некоторое время, что может замедлить работу приложения. Работа с объектами JavaScript, вместо реальных DOM-элементов, позволяет решить эту проблему, т.е. производить операции быстрее.

Виртуальный DOM представляет легковесную копию обычного DOM. Если приложению нужно узнать информацию о состоянии элементов, то происходит обращение к виртуальному DOM. Если необходимо изменить элементы веб-страницы, то изменения вначале вносятся в виртуальный DOM. Потом новое состояние виртуального DOM сравнивается с текущим состоянием. Если эти состояния различаются, то ReactJs находит минимальное количество манипуляций, которые необходимы до обновления реального DOM до нового состояния и производит их.

В итоге такая схема взаимодействия с элементами веб-страницы работает гораздо быстрее и эффективнее, в отличие от работы из JavaScript с DOM напрямую.

ReactJs использует JSX для описания визуального кода на JavaScript и разметки XML. JSX – это синтаксический сахар для функции создания элемента и компилируется в обычный синтаксис JavaScript. Использовать JSX рекомендуется для создания интерфейса.

ReactJs был выбран вместо JavaScript в силу следующих преимуществ:

1. компонентный подход – значительно улучшает организацию проекта, благодаря возможности переиспользовать и наследовать компоненты, а также разделять их на отдельные файлы и после собирать воедино;
2. использование виртуального DOM вместо обычного – ускоряющего отрисовку веб-страницы;
3. JSX синтаксис, благодаря которому разметка и логика размещаются вместе ускоряет и упрощает запись шаблонов. Так же JSX предлагает безопасность типов, большинство ошибок могут быть обнаружены во время компиляции.

### 4.2.2 NPM

В 2009 году появился Node.js, который вывел JavaScript за пределы браузеров. Если JavaScript выполняет действия на стороне клиента, то Node.js – на сервере. С помощью Node можно писать полноценные приложения.

Для написания дипломного проекта Node.js использовался в качестве легковесного сервера для фронтенд разработки, который помогал в запуске и тестировании приложения.

NPM (Node Package Manager) — это стандартный менеджер пакетов, автоматически устанавливающийся вместе с Node.js. Он используется для скачивания пакетов из облачного сервера npm, либо для загрузки пакетов на эти сервера. Все npm-команды прописываются в консоли (терминале). С помощью npm-скриптов можно создавать небольшие утилиты или описывать сложные системы сборки. Npm-скрипты позволяют запускать и тестировать приложения, настраивать слежение за изменением файлов.

### 4.2.3 Webpack

Приложения, написанные на JavaScript, постоянно усложняются, поэтому хорошим выходом является использование сборщика. Webpack – это модульный сборщик проектов, который позволяет объединять все ресурсы необходимые для проекта, в один файл. Это могут быть не только сторонние библиотеки, но и собственные файлы. Такая модульная система позволяет добиться лучшей организации проект, поскольку он разбит на небольшие модули.

Для того чтобы React работал, необходимо вместе с ним установить Babel. Он предоставляет транспайлинг ES6 и JSX в ES5. Транспайлер – это тип компилятора, который использует исходный код программы, написанной на одном языке программирования, в качестве исходных данных и производит эквивалентный исходный код на другом языке программирования. Транспайлер переводит между языками программирования, которые работают примерно на одном и том же уровне абстракции, в то время как традиционный компилятор переводит с более высокого уровня языка программирования на язык более низкого уровня.

### 4.2.4 React Router

React – это не полноценный фреймворк, а библиотека, т.е. он не решает все потребности приложения. Он позволяет создавать компоненты и управлять состоянием приложения. Для разработки полноценного одностраничного приложения понадобился React Router – это маршрутизатор. В отличие от других подобных он использует JSX. React Router предоставляет несколько собственных компонентов:

1. Router – определяет набор маршрутов, и когда к приложению приходит запрос, Router выполняет сопоставление запроса с маршрутами. И если какой-то маршрут совпадает с URL запроса, то этот маршрут выбирается для обработки запроса;
2. Route – этот компонент не создает DOM элементы, а координирует работу внутренних компонентов, т.е. определяет правила, по которым работает приложение. В атрибутах сюда передается адрес ссылки, которая будет доступна в адресной строке браузера и название компонента, который будет отрисовываться по этому адресу. В Route можно передавать параметры для url-адресов, например, чтобы отобразить конкретную задачу из списка задач передается id;
3. Link – используется для создания ссылок и транслируется в обычный <a href=”> в DOM. В Link передается соответствующий адрес компонента, указанный в Route и задается название ссылки.

### 4.2.5 Material UI

Material UI – это набор компонентов React, который реализует Google Material Design. Эти компоненты работают изолированно, это означает, что они являются само-поддерживающими и вводят только те стили, которые они должны отображать.

Данная библиотека доступна не только для веб-разработки, но и для мобильной разработки под iOS и Android. Material UI облегчает разработку и предоставляет множество готовых решений для дизайна интерфейсов. Основные компоненты, использовавшиеся в написании интерфейса:

1. paper – определяет блок, который напоминает лист белой бумаги и отдающий небольшую тень под собой. Имеет возможность изменять размер, компоноваться с другими подобными и быть вложенным. Обычно используется в качестве фона;
2. table – представляет таблицы, которые помогают отображать данные в удобочитаемом виде для пользователя, упрощают поиск нужной информации;
3. dialog – это всплывающие окна, могут использоваться как для простого вывода информации об успешном действии на странице, так и содержать в себе формы для обратной связи, либо поля для заполнения данных;
4. icon – определяет системный значок или значок пользовательского интерфейса, который символизирует команду, файл, действие. Google предоставляет широкий набор значков, которые можно использовать при создании интерфейса; Material-UI предоставляет два компонента для отображения системных значков: SvgIcon для отображения путей SVG и Icon для отображения значков шрифтов.
5. textField – текстовые поля, которые позволяю пользователю вводить текст в пользовательском интерфейсе. Обычно они используются в формах и диалогах, поисковых полях. TextField транслируется в input в HTML разметке;
6. tabs – это вкладки, похожие на те, что есть в браузере. Tabs помогают организовывать и перемещаться между группами контента, связанными на том же уровне иерархии;
7. buttons – компонент, отвечающий за отображение кнопок. Кнопки, обычно служат для сообщения о действиях, которые может выполнять пользователь. Библиотека Material UI дает широчайший выбор дизайна кнопок и анимации при наведении мышкой и клике. Внутрь кнопки можно вставлять текст, а также иконки, отражающие действия. Есть несколько размеров кнопок, предоставленных библиотекой по умолчанию. Кнопки можно изменять под нужды собственного дизайна, а именно:

* менять цвет;
* задавать размер;
* задавать радиус закругления углов;
* делать кнопку прозрачной с рамкой или полностью окрашенной и другое.

На рисунке 4.2 приведен ряд стандартных кнопок, предоставленных библиотекой Material UI.



Рисунок 4.2 – Набор кнопок от библиотеки Material UI

1. Typography отвечает за размеры шрифтов и их начертания. Помогает без использования дизайнера в команде, хорошо управлять шрифтами на веб-страницах благодаря типографской шкале, которая имеет ограниченный набор размеров шрифтов. Пример шкалы приведен на рисунке 4.3.



Рисунок 4.3 – Типографская шкала шрифтов Material UI

При реализации пользовательского интерфейса были так же использованы многие другие компоненты, предоставленные библиотекой Material UI. Ряд перечисленных выше компонентов и других, использовавшихся в разработанном приложении приведены на рисунках 4.4 и 4.5.



Рисунок 4.4 – Всплывающее окно с формой для создания новой задачи



Рисунок 4.5 – Вкладки, содержащие журнал работ и комментарии

Как уже упоминалось выше, предоставленные по умолчанию библиотекой Material UI стили, можно изменять под собственные при помощи CSS (Cascading Style Sheets) – каскадных таблиц стилей. Стили можно писать, как в специальных файлах с разрешением css, так и непосредственно в самих react-компонентах в фармате объекта.

### 4.2.6 HTML

Несмотря на то, что для написания проекта HTML в чистом виде не использовался, а был заменен JSX, на выходе получилась полноценная HTML-страница. HTML – это язык гипертекстовой разметки, с помощью которого создается структура web-страниц и email-писем. Язык гипертекстовой разметки благодаря тегам позволяет сообщить браузеру смысловую нагрузку элементов на странице.

### 4.2.7 CSS 3 (SASS)

CSS – это язык стилей, которые описывают внешний вид HTML-документа. Им задаются цвета, шрифты, расположения отдельных блоков и другие аспектов представления внешнего вида веб-страниц. Стили могут быть написаны как в самом HTML-документе, так и в отдельном файле с разрешением css и подключены в документ.

В наше время для больших и сложных проектов обычно приходится писать очень много стилей, что может затруднить читабельность кода и его сопровождение, многие участки в коде повторяются. Чтобы упростить задачу разработчикам были созданы множество библиотек и CSS-препроцессоров.

Язык написания стилей на препроцессорах очень схож с языком CSS, но предоставляет возможности, которые не доступны в CSS. Препроцессоры напрямую в браузере не работают и нуждаются в компиляции, результатом которой является CSS-файл. Компиляцию производят специальные утилиты, многие из них известны как сборщики проектов. В случае разработанного проекта, использовался сборщик Webpack. Его можно настроить на отслеживание изменений в файлах со стилями при их сохранении. Webpack собирает файлы воедино, если есть импортированные дочерние файлы и компилирует их в CSS-файл. А так же обновляет страницу в браузере при каждом сохранении файла, написанного на препроцессоре.

Для разработки интерфейса приложения был использован CSS-препроцессор SASS. Данный препроцессор предоставляет множество полезных функций, например:

1. переменные – позволяют хранить в себе информацию, которую можно использовать много раз на протяжении написании всех стилей проекта. В переменных можно хранить цвета, шрифты и любые другие значения;
2. вложенности – это подобно вложенностям при написании HTML. Благодаря этой возможность читаемость кода становится проще.
3. импорт – позволяет разбивать большие файлы на более мелкие, таким образом стили становится удобнее сопровождать. Обычный CSS так же имеет эту возможность, но у нее есть существенный недостаток: каждый раз, когда CSS использует @import, то в CSS создается еще один HTTP-запрос. SASS берет идею импорта, но вместо создания отдельного HTTP-запроса, он импортирует указанный файл в тот, где он вызывается. Таким образом, на выходе получается один CSS-файл, скомпилированный из нескольких фрагментов;
4. миксины – позволяют создавать группы деклараций CSS, которые приходится использовать несколько раз. Например, это полезно для вендорных префиксов, которые нужны для поддержки многих браузеров.

# 5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Поскольку разработанный дипломный проект является полноценным веб-приложением для его использования достаточно наличия веб-браузера и подключения к интернету.

При переходе на сайт, не авторизованный пользователь автоматически перенаправляется на страницу авторизации, которая представлена на рисунке 7.1. На этой странице необходимо ввести email и пароль.



Рисунок 7.1 – Страница авторизации

В случае успешного входа, а именно: email и пароль верны, и пользователь существует в системе, происходит перенаправление на главную страницу. Также авторизованному пользователю становится доступно меню с навигацией по сайту, доступ в личный кабинет и другие функции, которые соответствуют его роли. На главной странице, которая приведена на рисунке 7.2, отображаются задачи текущего авторизованного пользователя.



Рисунок 7.2 – Главная страница со списком задач

При клике по блоку с задачей, осуществляется переход на страницу с конкретной задачей. Страница конкретной задачи содержит ее детальное описание. Здесь можно установить статус задачи, написать комментарии и затраченное время на выполнение задачи в журнале работ. Для создателя задачи доступны кнопки редактирования, назначения ответственного за выполнение задачи, а также возможность удалить задачу. На рисунке 7.3 приведен скриншот страницы детального описания задачи.

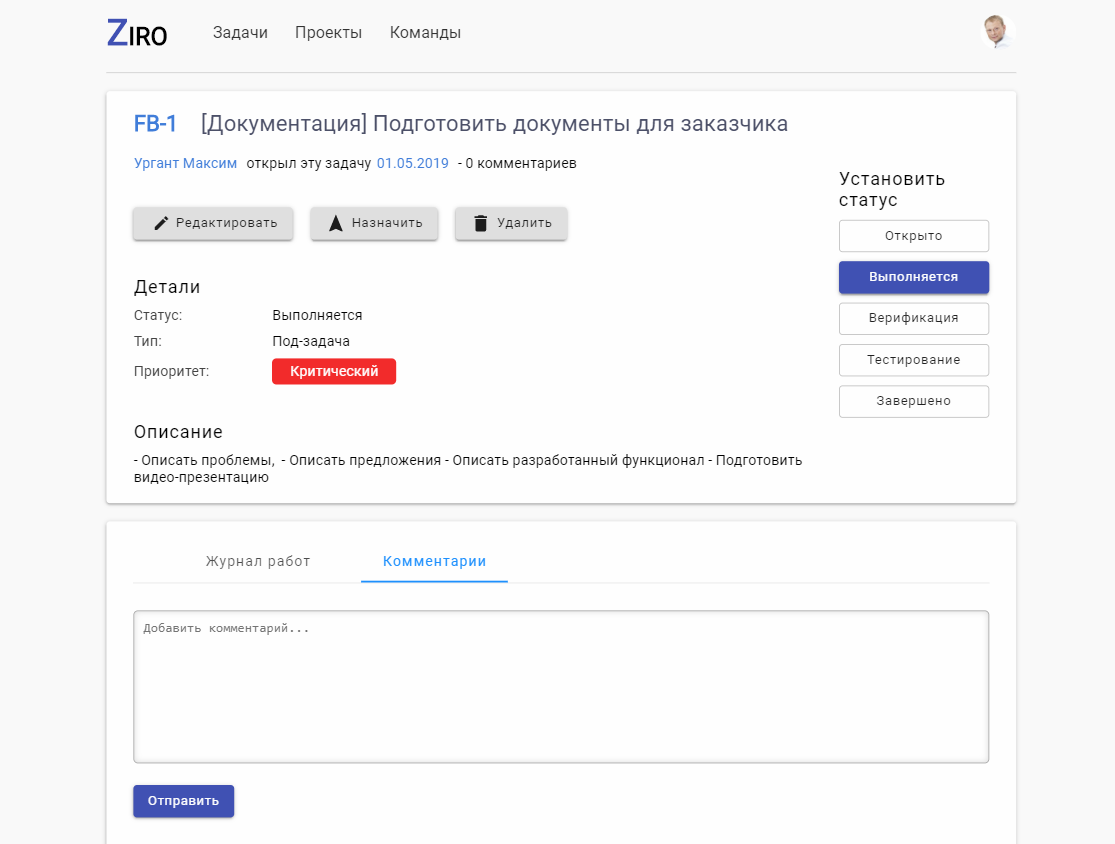


Рисунок 7.3 – Страница детального описания задачи

В профиль пользователя можно перейти, кликнув по фото в правом верхнем углу страницы. На странице профиля размещена информация о пользователе: имя, должность, контактные данные, если были указаны и другое.

Кликнув по кнопке редактировать (отмена) раскрывается блок с формой редактирования, автоматически подставляются данные текущего профиля. Что удобно, в случае, если пользователю необходимо поменять или добавить всего одно поле. При открытии формы редактирования текст кнопки «редактировать» меняется на «отмена». При нажатии на кнопку «сохранить» новые данные перезаписывают старые, форма редактирования закрывается и текст кнопки «отмена» меняется на «редактировать».

При клике по кнопке «выйти» происходит выход из системы, и пользователь перенаправляется на страницу авторизации. Страница профиля пользователя приведена на рисунке 7.4.



Рисунок 7.4 – Страница профиля пользователя

Страница «Проекты» содержит таблицу с проектами к которым относится текущий пользователь системы. В верхней части страницы расположено поле для поиска проекта. При клике по строке с проектом, пользователь переходит на страницу с детальным описанием проекта, она схожа со страницей детального описания задачи. На странице проекта можно скачать или загрузить текстовые-документы, относящиеся к этому проекту. Страница с таблицей проектов приведена на рисунке 7.5.



Рисунок 7.5 – Страница с таблицей проектов

На странице «Команды» отображен список групп пользователей, относящиеся к определенному проекту. Для каждого пользователя указана краткая информация: имя, фамилия, должность и контактная информация. При клике по блоку с конкретным пользователем происходит перенаправление на страницу профиля этого пользователя.

# 6 ТЕСТИРОВАНИЕ ПРОРАММНОГО ОБЕСПЕЧЕНИЯ

Тестирование программного обеспечения – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта.

Тестирование программного обеспечения будет проводится в рамках функционального тестирования, разделив его на критическое и углубленное.

Тестирование программы состоит из разработки тестовых случаев (тестов), их запуска и анализа полученных результатов.

Тестовые случаи – это алгоритмы проверки функциональности программы.

Каждый тестовый случай должен обладать следующими свойствами:

1. точно сформированной целью тестирования;
2. известными начальными условиями тестирования;
3. строго определенной средой тестирования;
4. тестовыми данными и ожидаемым результатом тестирования.

Тестирование было проведено в следующих версиях браузеров и операционных системах, представленных в таблице 6.1.

Таблица 6.1 – Операционные системы и браузеры для тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Google Chrome  v 74.0.3729.157 | Opera v 11.16 | Mozilla Firefox v 56.0 | Microsoft Edge  42.17134.1.0 |
| Windows 10 | + | + | + | + |
| Windows 7 | + | + | + | + |

### 6.1 Критическое тестирование

Критическое тестирование – это процесс поиска ошибок в программе при стандартной ее работе (при правильной последовательности действий, при верном заполнении полей и т. д.).

Тест критического пути является одним из самых распространенных видов функционального тестирования, в частности, для веб-проектов. Частота данного тестирования обусловлена в первую очередь необходимостью периодической проверки всего приложения в сжатые сроки. Плюс ко всему позволяет выявить самые быстро находимые дефекты и исправить приложение в более сжатые сроки.

С помощью данного вида тестирования покрываются все сценарии стандартного использования приложения, исключая негативные сценарии. Данный тип также характеризует тестирование по глубине его проведения.

Это и есть критическое тестирование, благодаря которому можно проверить правильность работы часто используемых функций.

Тестовые случаи для критического тестирования представлены в таблице 6.2.

Таблица 6.2 – Тестовые случаи для критического тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название модуля/экрана | Описание тестового случая | Ожидаемые результаты | Тестовый случай пройден? Да/Нет | Комментарии |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Запуск приложения | Запуск приложения на различных браузерах  1. Запускаем браузер  2. Вводим в адресной строке адрес приложения и нажимаем ввод. | 1. Открыт браузер  2. Загрузка первой страницы приложения | Да |  |
| 2 | Авторизация | Авторизация 1.Открываем веб-приложение.  2. В поле имя пользователя вводим «user1@test.ru».  3. В поле пароль вводим «user1». 4. Нажимаем кнопку «Войти». | 1. Открыта стартовая страница  2. Отображение введенного имени в поле имя пользователя  3. Отображение введенного пароля в поле пароль  4. Переход на персональную страницу пользователя | Да |  |

Продолжение таблицы 6.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 3 | Группа | Группа  1. Выбираем группу из списка групп.  2. Нажимаем кнопку просмотреть информацию о группе. | 1. Отображение информации о группе  2. Отображение информации посещаемости  3. Отображение информации о преподавателе  4. Отображение сообщений чата  5. Отображение списка слушателей | Да |  |
| 4 | Редактирование персональных данных | Редактирование персональных данных   1. Авторизация в приложении 2. Перейти на страницу пользователя 3. Выбрать пункт меню «Редактировать» | 1. В поле «Фамилия» отображение фамилии 2. В поле «Имя» отображение имени 3. В поле «Отчество» отображение отчества 4. В поле «Дата рождения» отображение даты 5. В поле «Дополнительная информация» отображение дополнительной информации | Да |  |
| 5 | Внутренняя почта | Внутренняя почта   1. Авторизация в приложении 2. Перейти на страницу пользователя 3. В пункте меню выбрать «Почта» | 1. Отображение списка входящих сообщений | Да |  |

Продолжение таблицы 6.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 6 | Отправка личного сообщения | Отправка личного сообщения   1. Авторизация в приложении 2. Перейти на страницу пользователя 3. В пункте меню выбрать «Почта» 4. Нажать кнопку «Новое сообщение» 5. Заполнить все поля 6. Нажать кнопку «Отправить» | 1. Закрытие модального окна 2. Отображение сообщения об успешной отправке личного сообщения | Да |  |
| 7 | Отправка сообщения в чат группы | Отправка сообщения в чат группы   1. Авторизация в приложении 2. Перейти на страницу пользователя 3. Нажимаем кнопку просмотреть информацию о группе 4. На странице группы написать сообщение в чат | 1. Введенное сообщение отобразится в списке сообщений чата | Да |  |

### 6.2 Углубленное тестирование

Углубленное (расширенное) тестирование – это процесс поиска ошибок в программе в нестандартных, непредвиденных ситуациях (например, при некорректно вводимых данных).

Примеры тестовых случаев для углубленного тестирования представлены ниже в таблице 6.3.

Таблица 6.3 - Тестовые случаи для углубленного тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название модуля/экрана | Описание тестового случая | Ожидаемые результаты | Тестовый случай пройден? Да/Нет | Комментарии |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Авторизация пользователя | Авторизация пользователя с Незаполненным полем «Пароль»  1. Запускаем веб-приложение  2. Поле «Email» заполняем  3. Поле «Пароль» оставляем пустым. | 1. Открыт браузер  2. Открыто веб-приложение  3. Отображения сообщения с текстом «Ошибка! Неверный логин или пароль» | Да |  |
| 2 | Редактирование персональных данных | Редактирование персональных данных  1. Запускаем веб-приложение  2. Выбираем пункт меню «Редактировать»  3. Заполняем все поля  4. Поле «Фамилия» оставляем пустым | 1. Открыта страница редактирования персональных данных  2. Отображение в полях ввода всех корректных данных  3. Отображение сообщения с текстом «Заполните это поле.» | Да |  |
| 3 | Отправка сообщения в чат группы | 1. Запуск веб-приложение 2. Авторизация 3. Выбираем группу из списка групп 4. Нажимаем кнопку просмотреть информацию о группе. 5. Поле «Сообщение» оставляем пустым 6. Нажать кнопку «Отправить» | 1. Открыта страница группы 2. Отображение сообщения с текстом «Заполните это поле.» | Да |  |

Продолжение таблицы 6.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 4 | Отправка личного сообщения | 1. Запуск веб-приложение 2. Авторизация 3. В пункте меню выбрать «Почта» 4. Нажать кнопку «Новое сообщение» 5. Все поля оставить пустыми | 1. Открыто окно ввода сообщения 2. Отображение сообщение с текстом «Заполните это поле» | Да |  |

# 7 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО СРЕДСТВА

### 7.1 Расчет сметы затрат, цены и прибыли на программное средство

В современных рыночных экономических условиях программное средство выступает преимущественно в виде продукции научно-технических организаций, представляющей собой функционально завершенные и имеющие товарный вид программные средства, реализуемые покупателям по рыночным отпускным ценам. Все завершенные разработки программных средств являются научно-технической продукцией.

Выбор эффективных проектов связан с их экономической оценкой и расчетом экономического эффекта.

У разработчика экономический эффект выступает в виде чистой прибыли, а у пользователя – в виде экономии различных ресурсов, получаемой за счет:

1. снижения трудоемкости расчетов и алгоритмизации программирования и отладки программ за счет использования программного средства в процессе разработки автоматизированных систем обработки данных;
2. снижения расходов на материалы (магнитные ленты, магнитные диски и прочие материалы);
3. улучшения показателей основной деятельности предприятий в результате использования программного средства.

Стоимостная оценка программного средства у разработчиков предполагает составление сметы затрат, которая включает следующие статьи:

1. заработная плата исполнителей основная (ЗО) и дополнительная (ЗД);
2. отчисления в фонд социальной защиты населения (ЗСЗ);
3. отчисления на развитие здравоохранения и охрану здоровья (ЗОЗ);
4. налоги, от фонда оплаты труда (Не);
5. материалы и комплектующие (М);
6. спецоборудование (РС);
7. машинное время (РМ);
8. расходы на научные командировки (РНК);
9. прочие прямые затраты (Пз);
10. накладные расходы (РН).

На основании сметы затрат рассчитывается себестоимость и отпускная цена программного средства.

### 6.1.1 Исходные данные

Исходные данные для расчета заносятся в таблицу (таблица 6.1).

Таблица 6.1 – Исходные данные для расчета

| № пп | Наименование показателя | Единица измерения | Условные обозначения | Норматив |
| --- | --- | --- | --- | --- |
| 1. | Коэффициент изменения скорости обработки информации | ед. | Кск | 0,5 |
| 2. | Численность разработчиков | чел. | Чр | 2 |
| 3. | Тарифная ставка 1-го разряда в организации | руб. | Тм1 | 36,4 |
| 4. | Тарифный коэффициент | ед. | Тк | 2,48 |
| 5. | Норматив дополнительной заработной платы рассчитывается по формуле | % | Нд | 16 |
| 6. | Норматив отчислений в фонд социальной защиты населения | ед. | Нсз | 35 |
| 7. | Норматив налога, уплачиваемого единым платежом | % | Нне | 4 |
| 8. | Норматив расходов на командировки в целом по научной организации | % | Нрнк | 30 |
| 9. | Норматив прочих затрат в целом по научной организации | % | Нпз | 20 |
| 10. | Норматив накладных расходов в целом по научной организации | % | Нрн | 100 |
| 11. | Уровень рентабельности программного средства | % | Урн | 53 |
| 12. | Норматив расходов на сопровождение и адаптацию | % | Нрса | 4 |
| 13. | Норматив отчислений в местный и республиканский бюджеты | % | Нмр | 3,9 |
| 14. | Норматив НДС | % | Ндс | 20 |
| 15.  Продолжение таблицы | Налог на прибыль при отсутствии льгот | % | Нn | 24 |
| № пп | Наименование статей | Единица измерения | Условные обозначения | Норматив |
| 16. | Норматив расхода машинного времени на отладку 100 строк исходного кода | час | Нмв | 12 |
| 17. | Количество дней в году | день | Дг | 365 |
| 18. | Количество праздничных дней в году | день | Дп | 9 |
| 19. | Количество выходных дней в году | день | Дв | 104 |
| 20. | Количество дней отпуска | день | До | 24 |

### 6.1.2 Общая характеристика разрабатываемого программного средства

Данное программное средство является программным средством общего назначения, основные функции:

1. организация ввода информации;
2. контроль, предварительная обработка и ввод информации;
3. управление вводом/выводом;
4. генерация структуры базы данных;
5. манипулирование данными;
6. организация поиска и поиск в базе данных;
7. обслуживание файлов;
8. обработка ошибочных и сбойных ситуаций;
9. справка и обучение.

Разрабатываемое программное средство входит в третью группу сложности.

***6***.1.3 Определение объема программного средства

Объем программного средства определяется путем подбора аналогов на основании классификации типов программных средств, каталога функции программного средства и аналогов программного средства в разрезе функций, которые постоянно обновляются и утверждаются в установленном порядке. На основании информации о функциях разрабатываемого программного средства по каталогу функций определяется объем функций. Общий объем программного средства рассчитывается по формуле:

 (6.1)

где VO – общий объем программного средства, строка исходного кода;

Vi – объем функций программного средства, строка исходного кода;

n – общее число функций.

Содержание и объем функций разрабатываемого программного средства приведено в таблице 6.2.

Таблица 6.2 – Содержание и объем функций разрабатываемого программного средства

|  |  |  |
| --- | --- | --- |
| № функции | Содержание функции | Объем функций (строки исходного кода) |
| 101 | Организация ввода информации | 150 |
| 102 | Контроль, предварительная обработка и ввод информации | 450 |
| 111 | Управление вводом/выводом | 2 400 |
| 201 | Генерация структуры базы данных | 4 300 |
| 207 | Манипулирование данными | 9 550 |
| 208 | Организация поиска и поиск в базе данных | 5 480 |
| 304 | Обслуживание файлов | 420 |
| 506 | Обработка ошибочных и сбойных ситуаций | 410 |
| 604 | Справка и обучение | 720 |
|  | ИТОГО | 23 880 |

### 6.1.4 Расчет трудоемкости выполняемой работы

На основании общего объема программного средства определяется нормативная трудоемкость (ТН). Нормативная трудоемкость устанавливается с учетом сложности программного средства. Выделяется три группы сложности, в которых учтены следующие составляющие программного средства; языковой интерфейса, ввод-вывод, организация данных, режим работы, операционная и техническая среда. Кроме того, устанавливаются дополнительные коэффициенты сложности программного средства.

С учетом дополнительного коэффициента сложности КСЛ рассчитывается общая трудоемкость программного средства:

 (6.2)

где ТО – общая трудоемкость ПС, человеко-день;

ТН – нормативная трудоемкость ПС, человеко-день;

КСЛ – дополнительный коэффициент сложности ПС.

Поскольку разрабатываемое ПС относится к группе сложности 1, то нормативная трудоемкость Tн равна 442. ПС обладает характеристикой функционирования в расширенной операционной среде, поэтому коэффициент сложности Kсл равен 0.08.

При решении сложных задач с длительным периодом разработки ПС трудоемкость определяется по стадиям разработки (техническое задание – ТЗ, эскизный проект – ЭП, технический проект – ТП, рабочий проект – РП и внедрение – ВН) с учетом новизны, степени использования типовых программ и удельного веса трудоемкости стадий разработки ПС и общей трудоемкости разработки ПС. При этом на основании общей трудоемкости рассчитывается уточненная трудоемкость с учетом распределения по стадиям (ТУ).

 (6.3)

где Тi – трудоемкость разработки ПС на i-й стадии, человеко-день;

m – количество стадий разработки.

Трудоемкость ПС по стадиям определяется с учетом новизны и степени использования в разработке типовых программ и ПС.

 (6.4)

где ТСТi – трудоемкость разработки ПС на i-й стадии (технического задания, эскизного проекта, технического проекта, рабочего проекта и внедрения), человеко-день;

КН – поправочный коэффициент, учитывающий степень новизны ПС;

КТ – поправочный коэффициент, учитывающий степень использования в разработке типовых программ и ПС;

dСТi – удельный вес трудоемкости i-й стадии разработки ПС в общей трудоемкости разработки ПС.

На основании данных в приложениях разрабатываемое ПС относится к 3-ей группе сложности и имеет степень новизны B, при этом Ксл=0,08, а Кн=0,7. Коэффициенты удельных весов трудоемкостей для каждой стадии: ТЗ=0,09; ЭП=0,07; ТП=0,07; РП=0,61; ВН=0,16.

Исходя из этих данных, можно рассчитать трудоемкость ПС на каждой стадии:

Тогда уточненная трудоемкость с учетом распределения по стадиям равна:

### 6.1.5 Расчет основной заработной платы исполнителей

Эффективный фонд времени работы одного работника (ФЭФ) рассчитывается по формуле:

(6.5)

где ДГ – количество дней в году, день;

ДП – количество праздничных дней в году, день;

ДВ – количество выходных дней в году, день;

ДО – количество дней отпуска, день.

При утверждении плановой численности разработчиков продолжительность разработки определяется по формуле:

 (6.6)

где ТР – срок разработки ПС (лет);

Тi – трудоемкость разработки ПС на i-й стадии (человеко-дней);

ЧРi – численность разработчиков ПС на i-й стадии (чел.);

m – число стадий.

Уточненная трудоемкость и общая плановая численность разработчиков служат базой для расчета основной заработной платы. По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов-исполнителей, участвующих в разработке ПС, с определением образования, специальности, квалификации и должности.

Таким образом, можно рассчитать эффективный фонд времени:

А также срок разработки ПС при количестве разработчиков на всех стадиях разработки ПС равным 2.

В соответствии с «Рекомендациями по применению «Единой тарифной сетки» рабочих и служащих народного хозяйства» и тарифными разрядами и коэффициентами должностей руководителей научных организаций и вычислительных центров, бюджетных учреждений науки непроизводственных отраслей народного хозяйства каждому исполнителю устанавливается разряд и тарифный коэффициент.

Месячная тарифная ставка каждого исполнителя (ТМ) определяется путем умножения действующей месячной тарифной ставки 1-го разряда (ТМ1) на тарифный коэффициент (ТК), соответствующий установленному тарифному разряду:

. (6.7)

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленный при семичасовом рабочем дне фонд рабочего времени (Фр):

 (6.8)

где ТЧ – часовая тарифная ставка, руб.;

ТМ – месячная тарифная ставка, руб.

Оба работника имеют одинаковую квалификацию и принадлежат 10-му разряду. Тарифная месячная и часовая ставки для них будут равны:

Основная заработная плата исполнителей на конкретное ПС рассчитывается по формуле

 (6.9)

где n – количество исполнителей, занятых разработкой конкретного ПС;

TЧi – часовая тарифная ставка i-го исполнителя (руб.);

ФЭi – эффективный фонд рабочего времени i-го исполнителя (дней);

ТЧ – количество часов работы в день (ч);

К – коэффициент премирования.

### 6.1.6 Расчет дополнительной заработной платы исполнителей

Дополнительная заработная плата на конкретное ПС (ЗДi) включает выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяется по нормативу в процентах к основной заработной плате:

 (6.10)

где ЗДi – дополнительная заработная плата исполнителей на конкретное ПС, руб.;

НД – норматив дополнительной заработной платы, %.

### 6.1.7 Расчет отчислений в фонд социальной защиты населения

Отчисления в фонд социальной защиты населения (ЗСзi) определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной зарплаты исполнителей, определенной по нормативу, установленному в целом по организации:

 (6.11)

где НСЗ – норматив отчислений в фонд социальной защиты населения, %.

### 6.1.8 Расчет налога на ликвидацию последствий чернобыльской катастрофы

Налоги рассчитываемые от фонда оплаты труда определяются в соответствии с действующими законодательными актами по нормативам в процентном отношении к сумме всей заработной платы, относимой на ПС (налог, уплачиваемый единым платежом, включая налог на ликвидацию последствий чернобыльской катастрофы и отчисления в фонд занятости (Неi)):

 (6.12)

где Нне – норматив налога, уплачиваемого единым платежом (%).

### 6.1.9 Расчет расходов по статье «Материалы»

Расходы по статье «Материалы» (М) определяются на основании сметы затрат, разрабатываемой на ПС с учетом действующих нормативов. По статье «Материалы» отражаются расходы на магнитную носители, перфокарты, бумагу, красящие ленты и другие материалы, необходимые для разработки ПС. Нормы расхода материалов в суммарном выражении (НМ) определяются в расчете на 100 строк исходного кода. Сумма затрат материалов рассчитывается по формуле:

 (6.13)

где НМi – норма расхода материалов в расчете на 100 строк исходного кода ПС, руб.;

VOi – общий объем ПС на конкретное ПС, строка исходного кода.

### 6.1.10 Расчет расходов по статье «Спецоборудование»

Расходы по статье «Спецоборудование» (РСi) включает затраты средств на приобретение вспомогательных специального назначения технических и программных средств, необходимых для разработки конкретного ПС, включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию. Затраты по этой статье определяются в соответствии со сметой расходов, которая составляется перед разработкой ПС. Данная статья включается в смету расходов на разработку ПС в том случае, когда приобретаются специальное оборудование или специальные программы, предназначенные для разработки и создания только данного ПС:

 (6.14)

где ЦСi – стоимость конкретного специального оборудования, руб.;

n – количество применяемого специального оборудования.

Необходимо учесть следующие расходы по данной статье:

* хостинг-план для размещения веб-сайта в интернете) – 28,9 рублей;
* VS Resharper – 55,23 рублей.

### 6.1.11 Расчет расходов по статье «Спецоборудование»

Расходы по статье «Машинное время» (РМi) включают оплату машинного времени, необходимого для разработки и отладки ПС, которое определяется по нормативам (в машино-часах) на 100 строк исходного кода (НМВ) машинного времени в зависимости от характера решаемых задач и типа ПЭВМ:

 (6.15)

где ЦМi – цена одного машино-часа, руб.;

VОi – общий объем ПС, строка исходного кода;

НМВ – норматив расхода машинного времени на отладку 100 строк исходного кода, машино-час.

Цена одного машино-часа (ЦМi) для обеспечения работы веб-сервера составляет 2 рубля.

### 6.1.12 Расчет расходов по статье «Научные командировки»

Расходы по статье «Научные командировки» (РНкi) на конкретное ПС определяются по нормативу, разрабатываемому в целом по научной организации, в процентах к основной заработной плате:

 (6.16)

где НРНК – норматив расходов на командировки в целом по научной организации, %.

### 6.1.13 Расчет расходов по статье «Прочие затраты»

Расходы по статье «Прочие затраты» (ПЗi) на конкретное ПС включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по научной организации, в процентах к основной заработной плате:

 (6.17)

где НПЗ – норматив прочих затрат в целом по научной организации.

### 6.1.14 Расчет расходов по статье «Накладные расходы»

Затраты по статье «Накладные расходы» (РНi), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды (РНi), относятся на конкретное ПС по нормативу (НРН) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по научной организации:

 (6.18)

где РНi – накладные расходы на конкретное ПС, руб.;

НРН – норматив накладных расходов в целом по научной организации.

### 6.1.15 Расчет общей суммы расходов на разработку

Общая сумма расходов по всем статьям сметы (СРi) на ПС рассчитывается по формуле:

 (6.19)

Кроме того, организация-разработчик осуществляет затраты на сопровождение и адаптацию ПС (РСАi), которые определяются по нормативу (НРСА)

 (6.20)

где Нрca – норматив расходов на сопровождение и адаптацию, %.

Общая сумма расходов на разработку (с затратами на сопровождение и адаптацию) как полная себестоимость ПС (СП) определяется по формуле:

 (6.21)

### 6.1.16 Расчет прибыли и отпускной цены

Рентабельность и прибыль по создаваемому ПС определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость и отчисления на содержание ведомственного жилого фонда. Прибыль рассчитывается по формуле:

 (6.22)

где Ппсi – прибыль от реализации ПС, руб.;

УРпi – уровень рентабельности ПС, %;

СПi – себестоимость ПС, руб.

Прогнозируемая цена ПС без налогов (Цпi):

. (6.23)

Отчисления и налоги в местный и республиканский бюджеты единым платежом (Омрi):



(6.24)

где Нмр – норматив отчислений в местный и республиканский бюджеты (%).

Налог на добавленную стоимость (НДСi):

, (6.25)

где Ндс – норматив НДС, %.

Прогнозируемая отпускная цена (Цоi):

. (6.26)

.

### 6.2 Расчет экономического эффекта от применения ПС у пользователя

### 6.2.1 Исходные данные

Исходные данные для расчета экономического эффекта от применения ПС у пользователя сведены в таблицу 6.3.

Таблица 6.3 – Исходные данные для расчета экономического эффекта

| Наименование показателей | Обозначения | Единицы измерения | Значение показателя | |
| --- | --- | --- | --- | --- |
| в базовом варианте | в новом варианте |
| 1. Средняя трудоемкость работ в расчете на 100 КБ | Тс1  Тс2 | человеко-час на 100 КБ | 1,8 | 1,7 |
| 2.Средний расход машинного времени в расчете на 100 КБ | Мв1  Мв2 | машино-час на 100 КБ | 9,0 | 8,91 |
| 3.Средний расход материалов в расчете на 100 КБ | Мт1  Мт2 | руб. на 100 КБ | 0,032 | 0,028 |
| 4. Количество типовых задач, решаемых за год | Зт2 | задача | 9 | |
| 5. Ставка налога на прибыль | Нп | % | 18 | |

### 6.2.2 Расчет объема работ

Объем работ в зависимости от функциональной группы и назначения ПС можно определить по формуле:

А = Vпс ∙ Кпс, (6.27)

где Vпс – объем ПС в натуральных единицах измерения;

Кпс – коэффициент применения ПС.

### 6.2.3 Расчет капитальных затрат

Общие капитальные вложения (Ко) заказчика (потребителя), связанные с приобретением, внедрением и использованием ПС, рассчитываются по формуле:

(6.28)

где Кпр – затраты пользователя на приобретение ПС по отпускной цене разработчика с учетом стоимости услуг по эксплуатации и сопровождению, руб.;

Кос – затраты пользователя на освоение ПС, руб.;

Ктс – затраты на доукомплектацию ВТ техническими средствами в связи с внедрением нового ПС, руб.;

Коб – затраты на пополнение оборотных средств в связи с использованием нового ПС, руб.

Затраты на освоение ПС и на пополнение оборотных средств рекомендуется рассчитывать по формулам:

, (6.29)

где Нкос - норматив затрат пользователя на освоение ПС, равный 0,01.

, (6.30)

где Нкоб - норматив затрат на пополнение оборотных средств в связи с использованием нового ПС, равный 0,01.

### 6.2.4 Расчет экономии основных видов ресурсов в связи с использованием нового программного средства

Экономия затрат на заработную плату при использовании нового ПС в расчете на объем выполненных работ:

, (6.31)

где Сзе – экономия затрат на заработную плату при решении задач с использованием нового ПС в расчете на 100 КБ, руб.;

А2 – объем выполненных работ с использованием нового ПС (100 КБ).

Экономия затрат на заработную плату в расчете на 100 КБ (Сзе):

, (6.32)

где Зсм – среднемесячная заработная плата одного программиста, руб.;

Тс1, Тс2 – снижение трудоемкости работ в расчете на 100 строк кода (человеко-часов);

Тч – количество часов работы в день (ч);

Др – среднемесячное количество рабочих дней.

руб.

Объем выполненных работ с использованием нового ПС (100 КБ):

, (6.33)

где Ао – объем работ необходимый для решения одной задачи (100 КБ);

Зт2 – количество типовых задач, решаемых за год (задач).

Экономия затрат на оплату машинного времени (См) в расчете на выполненный объем работ в результате применения нового ПС:

, (6.34)

где Сме – экономия затрат на оплату машинного времени при решении задач с использованием нового ПС в расчете на 100 КБ.

Экономия затрат на оплату машинного времени в расчете на 100 КБ (Сме):

, (6.35)

где Цм – цена одного машино-часа работы ЭВМ;

Мв1, Мв2 – средний расход машинного времени в расчете на 100 КБ при применении соответственно базового и нового ПС.

Экономия затрат на материалы (Смт) при использовании нового ПС в расчете на объем выполненных работ:

, (6.36)

где Смте – экономия затрат на материалы в расчете на 100 КБ при использовании нового ПС.

, (6.37)

где См1, См2 – средний расход материалов у пользователя в расчете на 100 КБ при использовании соответственно базового и нового ПС, руб.

Общая годовая экономия текущих затрат, связанных с использованием нового ПС (Со):

. (6.38)

### 6.2.5 Расчет экономического эффекта

Внедрение нового ПС позволит пользователю сэкономить на текущих затратах, т.е. практически получить на эту сумму дополнительную прибыль. Для пользователя в качестве экономического эффекта выступает лишь чистая прибыль – дополнительная прибыль, остающаяся в его распоряжении (ΔПч), которые определяются по формуле:

, (6.39)

где Нп – ставка налога на прибыль (%).

В процессе использования нового ПС чистая прибыль в конечном итоге возмещает капитальные затраты. Однако, полученные при этом суммы результатов (прибыли) и затрат (капитальных вложений) по годам приводят к единому времени – расчетному году (за расчетный год принят 2019 год) путем умножения результатов и затрат за каждый год на коэффициент привидения (ALFAt), который рассчитывается по формуле:

, (6.40)

где Ен – норматив привидения разновременных затрат и результатов;

tp – расчетный год, tp=1;

t – номер года, результаты и затраты которого приводятся к расчетному (2019-1, 2020-2, 2021-3, 2022-4).

Норматив приведения разновременных затрат и результатов (Ен) для программных средств ВТ в существующей практике принимается в пределах 0,2–0,4. Например, при нормативе 0,4 коэффициентам приведения (ALFAt) по годам будут соответствовать следующие значения:

– расчетный год;

 – 2020 год;

– 2021 год;

 – 2022 год.

Результаты расчета экономического эффекта сведены в таблицу 6.4.

Таблица 6.4 – Данные экономического эффекта от использования нового ПС

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Показатели | Ед.  измерения | 2019 | 2020 | 2021 | 2022 |
| Результаты: |  |  |  |  |  |
| Прирост прибыли за счет экономии затрат (Пч) | руб. | 22 194,4 | 22 194,4 | 22 194,4 | 22 194,4 |
| То же с учетом фактора времени | руб. | 22 194,4 | 15 846,8 | 11 319,15 | 8 078,76 |
| Затраты: |  |  |  |  |  |
| Приобретение, адаптация и освоение ПС (Кпр) | руб. | 12 669,89 | - | - | - |
| Освоение ПС (Кос) | руб. | 126,7 | - | - | - |
| Доукомплектование ВТ техническими средствами (Ктс) | руб. | - | - | - | - |
| Пополнение оборотных средств (Коб) | руб. | 126,7 | 126,7 | 126,7 | 126,7 |
| Всего затрат | руб. | 12 923,29 | 126,7 | 126,7 | 126,7 |
| То же с учетом фактора времени | руб. | 12 923,29 | 90,46 | 64,62 | 46,12 |
| Экономический эффект: |  |  |  |  |  |
| Превышение результата над затратами | руб. | 9 271,11 | 15 756,34 | 11 254,53 | 8 032,64 |
| То же с нарастающим итогом | руб. | 9 271,11 | 25 027,45 | 36 281,98 | 44 314,63 |
| Коэффициент приведения | единиц | 1 | 0,714 | 0,510 | 0,364 |

Таким образом, отпускная цена предлагаемой разработки составляет 12 700 рублей, окупаемость ее будет достигнута уже на первом году применения. При этом эффект от использования в течение четырех лет составит 44 315 рублей.

# 8 ОХРАНА ТРУДА

### 5.1 Производственная санитария, техника безопасности и пожарная профилактика

Работающие с ПЭВМ могут подвергаться воздействию различных опасных и вредных производственных факторов, основными из которых являются повышенные уровни: электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; статического электричества; запыленности воздуха рабочей зоны; повышенное или пониженное содержание аэроионов в воздухе рабочей зоны; повышенный или пониженный уровень освещенности рабочей зоны, содержание в воздухе рабочей зоны оксида углерода, озона, аммиака, фенола, формальдегида и полихлорированных фенилов; напряжение зрения, памяти, внимания; длительное статическое напряжение; большой объем информации, обрабатываемой в единицу времени; монотонность труда; нерациональная организация рабочего места; эмоциональные перегрузки.

Работа с ПЭВМ проводится в соответствии с Санитарными нормами и правилами «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами» и Гигиеническим нормативом «Предельно-допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами», утвержденными постановлением Министерства здравоохранения от 28.06.2013 г. № 59 и Типовой инструкцией по охране труда при работе с персональными ЭВМ, утвержденной постановлением Министерства труда и социальной защиты от 24.12.2013 № 130.

Согласно вышеуказанных документов площадь помещения на одного пользователя ПЭВМ на базе плоских дискретных экранов (жидкокристаллические, плазменные) составляет не менее 4,5 м2.

### 5.1.1 Метеоусловия

В производственных помещениях, в которых работа с использованием ПЭВМ является основной (операторские, расчетные, посты управления, залы вычислительной техники), обеспечиваются оптимальные параметры микроклимата для категории работ 1а и 1б (табл. 5.1) согласно вышеуказанных нормативных документов.

Таблица 5.1 – Оптимальные параметры микроклимата для помещений с ВДТ, ЭВМ и ПЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период года | Категория работ | Температура воздуха, оС, не более | Относительная влажность воздуха, % | Скорость движения воздуха, м/с |
| Холодный | легкая-1а | 22-24 | 40-60 | 0,1 |
| легкая-1б | 21-23 | 40-60 | 0,1 |
| Теплый | легкая-la | 23-25 | 40-60 | 0,1 |
| легкая-1б | 22-24 | 40-60 | 0,2 |

Работа с компьютером относится к категории 1а (работы, производимые сидя и сопровождающиеся незначительным физическим напряжением, при которых расход энергии составляет до 120 ккал/ч, т.е. до 139 Вт).

Интенсивность теплового излучения работающих от нагретых поверхностей технологического оборудования, осветительных приборов, инсоляции на постоянных рабочих местах не превышает значений, указанных в табл. 5.2 согласно Санитарных норм и правил.

Таблица 5.2 – Предельно допустимые уровни интенсивности излучения в инфракрасном и видимом диапазоне излучения на расстоянии 0,5 м со стороны экрана ВДТ, ЭВМ и ПЭВМ

|  |  |  |  |
| --- | --- | --- | --- |
| Диапазоны длин волн | 400-760 нм | 760-1050 нм | свыше 1050 нм |
| Предельно допустимые уровни | 0,1 Вт/м2 | 0,05 Вт/м2 | 4,0 Вт/м2 |

Для создания нормальных метеорологических условий наиболее целесообразно уменьшить тепловыделения от самого источника — монитора, что предусматривается при разработке его конструкции.

В производственных помещениях для обеспечения необходимых показателей микроклимата предусмотрены системы отопления, вентиляции и кондиционирования воздуха.

### 5.1.2 Вентиляция и отопление

Воздух рабочей зоны помещения соответствует санитарно-гигиеническим требованиям по содержанию вредных веществ и частиц пыли, приведенным в Санитарных нормах и правилах «Требованию к контролю воздуха рабочей зоны», Гигиеническом нормативе «Предельно допустимые концентрации вредных веществ в воздухе рабочей зоны», утв. пост. Министерства здравоохранения от 10.10.2017 г. № 92.

В помещениях, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы.

Уровни положительных и отрицательных аэроионов, а также коэффициент униполярности в воздухе всех помещений, где расположены ПЭВМ, соответствуют значениям, указанным в табл. 5.3.

Таблица 5.3 – Уровни ионизации и коэффициент униполярности воздуха помещений при работе с ВДТ, ЭВМ и ПЭВМ

|  |  |  |  |
| --- | --- | --- | --- |
| Уровни | Число ионов в 1 см3 воздуха | | Коэффициент униполярности (У) |
| n+ | n- |
| Минимально допустимые | 400 | 600 | 0,4 ≤ У < 1,0 |
| Оптимальные | 1500-3000 | 3000-5000 |
| Максимально допустимые | 50000 | 50000 |

Одним из мероприятий по оздоровлению воздушной среды является устройство вентиляции и отопления. Задачей вентиляции является обеспечение чистоты воздуха и параметров метеорологических условий на рабочих местах. Чистота воздушной среды достигается удалением загрязненного или нагретого воздуха из помещения и подачей в него свежего воздуха. Для поддержания нормального микроклимата необходим достаточный объем вентиляции, для чего в вычислительном центре предусматривается кондиционирование воздуха, осуществляющее поддержание постоянных параметров микроклимата в помещении независимо от наружных условий.

Параметры микроклимата поддерживаются в холодный период года за счет системы водяного отопления с нагревом воды до 100°С, а в теплый - за счет кондиционирования, с параметрами, отвечающими требованиям СНБ 4.02.01-03.

### 5.1.3 Освещение

В помещении для эксплуатации ПЭВМ предусмотрены естественное и искусственное освещение. Естественное освещение на рабочих местах осуществляется через световые проемы, ориентированные преимущественно на север, северо-восток, восток, запад или северо-запад и обеспечивает коэффициент естественной освещенности не ниже 1,5 %. Оконные проемы оборудованы регулируемыми устройствами типа жалюзи, занавесей.

Для внутренней отделки интерьера помещений используются материалы с коэффициентом отражения для потолка – 0,7- 0,8; для стен – 0,5- 0,6; для пола – 0,3- 0,5.

Искусственное освещение в помещениях осуществляется системой общего равномерного освещения. При работе с документами применяется система комбинированного освещения, а освещенность поверхности стола в зоне размещения рабочего документа должна составлять 300-500 люкс. Освещенность поверхности экрана не более 300 люкс. В качестве источников света применяем люминесцентные лампы типа ЛБ. Коэффициент запаса для осветительных установок общего освещения принимается равным 1,4, а коэффициент пульсации – не более 5 %.

***5.1.4 Шум***

Основными источниками шума в помещениях, оборудованных ЭВМ, являются принтеры, множительная техника и оборудование для кондиционирования воздуха, в самих ЭВМ — вентиляторы систем охлаждения и трансформаторы.

В табл. 5.4 приведены нормированные уровни шума согласно Санитарных норм и правил «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами» и Гигиенических нормативов «Предельно-допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами», которые обеспечиваются за счет использования малошумного оборудования, применения звукопоглощающих материалов для облицовки помещений, а также различных звукопоглощающих устройств (перегородки и т. п.).

Таблица 5.4 – Предельно-допустимые уровни звука, эквивалентные уровни звука и уровни звукового давления в октавных полосах частот при работе с ВДТ, ЭВМ и ПЭВМ и периферийными устройствами

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Категория нормы  шума | Уровни звукового давления, дБ, в октавных полосах  со среднегеометрическими частотами, Гц | | | | | | | | | Уровни звука и  эквивалентные уровни звука, дБА |
| 31.5 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 |
| I | 86 | 71 | 61 | 54 | 49 | 45 | 42 | 40 | 38 | 50 |
| II | 93 | 79 | 70 | 63 | 58 | 55 | 52 | 50 | 49 | 60 |
| III | 96 | 83 | 74 | 68 | 63 | 60 | 57 | 55 | 54 | 65 |
| IV | 103 | 91 | 83 | 77 | 73 | 70 | 68 | 66 | 64 | 75 |

### 5.1.5 Электробезопасность

Помещение вычислительного центра по степени опасности поражения электрическим током относится к помещениям без повышенной опасности.

Основные меры защиты от поражения током:

1. изоляция и недоступность токоведущих частей;
2. защитное заземление (R3 = 4 Ом ГОСТ 12.1.030 - 81).

Первая помощь при поражениях электрическим током состоит из двух этапов: освобождение пострадавшего от действия тока и оказание ему доврачебной медицинской помощи. После освобождения пострадавшего от действия электрического тока необходимо оценить его состояние. Во всех случаях поражения электрическим током необходимо вызвать врача независимо от состояния пострадавшего.

***5.1.6 Излучение***

При работе с дисплеем могут возникать следующие опасные факторы: электромагнитные и электростатические поля, ультрафиолетовое и инфракрасное излучение.

Уровни физических факторов на рабочих местах пользователей, создаваемые ПЭВМ и периферийными устройствами, не превышают предельно-допустимые уровни: электромагнитных и электростатических полей (табл. 5.5, 5.6), ультрафиолетового (табл. 5.7), установленных Гигиеническим нормативом «Предельно допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами».

Таблица 5.5 – Предельно допустимые уровни электромагнитных полей от экранов ВДТ, ЭВМ и ПЭВМ

|  |  |  |
| --- | --- | --- |
| Наименование параметра | | Предельно-допустимые уровни |
| Напряженность электрического поля в диапазоне частот: | |  |
|  | 5 Гц-2 кГц | не более 25,0 В/м |
|  | 2-400 кГц | не более 2,5 В/м |
| Плотность магнитного потока магнитного поля в диапазоне частот: | |  |
|  | 5 Гц-2 кГц | не более 250 нТл |
|  | 2-400 кГц | не более 25 нТл |
| Напряженность электростатического поля | | не более 15 кВ/м |

Таблица 5.6 – Предельно допустимые уровни электромагнитных полей при работе с ВДТ, ЭВМ, ПЭВМ от клавиатуры, системного блока, манипулятора «мышь», беспроводных системам передачи информации и иных периферийных устройств

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Диапазоны частот | 0,3-300  кГц | 0,3-3  МГц | 3-30  МГц | 30-300  МГц | 0,3-300  ГГц |
| Предельно допустимые уровни | 25 В/м | 15 В/м | 10 В/м | 3 В/м | 10 мкВт/см2 |

Таблица 5.7 – Предельно допустимые уровни интенсивности излучения в ультрафиолетовом диапазоне на расстоянии 0,5 м со стороны экрана ВДТ, ЭВМ и ПЭВМ

|  |  |  |  |
| --- | --- | --- | --- |
| Диапазоны длин волн | 200-280 нм | 280-315 нм | 315-400 нм |
| Предельно допустимые уровни | не допускается | 0,0001 Вт/м2 | 0,1 Вт/м2 |

Наиболее эффективным и часто применяемым методом защиты от электромагнитных излучений является установка экранов. Экранируют либо источник излучения, либо рабочее место. Часто экран устанавливают непосредственно на монитор.

При работе монитора на экране кинескопа накапливается электростатический заряд, создающий электростатическое поле. При этом персонал, работающий с монитором, приобретают электростатический потенциал. Заметный вклад в общее электростатическое поле вносят электризующиеся от трения поверхности клавиатуры и мыши.

***5.1.7 Пожарная безопасность***

По взрывопожарной и пожарной опасности помещения и здания для ЭВМ относятся к категории Д согласно ТКП 474-2013. Здания для ВЦ и части зданий другого назначения, в которых предусмотрено размещение ЭВМ, относятся к 2 степени огнестойкости согласно ТКП 45-2.02-315-2018.

Для предотвращения распространения огня во время пожара с одной части здания на другую устраивают противопожарные преграды в виде стен, перегородок, дверей, окон. Особое требование предъявляется к устройству и размещению кабельных коммуникаций.

Нормы первичных средств пожаротушения для вычислительных центров приведены в табл. 5.8.

Таблица 5.8 – Примерные нормы первичных средств пожаротушения для вычислительного центра

|  |  |  |  |
| --- | --- | --- | --- |
| Помещение | Площадь, м2 | Углекислотные огнетушители ручные | Порошковые огнетушители |
| Вычислительный центр | 100 | 1 | 1 |

Для ликвидации пожаров в начальной стадии применяются первичные средства пожаротушения: внутренние пожарные водопроводы, огнетушители типа ОВП-10, ОУ-2, асбестовые одеяла и др.

Эвакуация персонала вычислительного центра осуществляется через эвакуационные выходы. Количество и общая ширина эвакуационных выходов определяются в зависимости от максимального возможного числа эвакуирующихся через них людей и предельно допустимого расстояния от наиболее удаленного места возможного пребывания людей до ближайшего эвакуационного выхода согласно ТКП 45-2.02-315-2018.

Расчетное время эвакуации устанавливается по реальному расчету времени движения одного или нескольких потоков людей через эвакуационные выходы из наиболее удаленных мест размещения людей. Необходимое время эвакуации устанавливается на основе данных о критической продолжительности пожара с учетом степени огнестойкости здания, категории производства по взрывной и пожарной опасности. Для успешной эвакуации необходимо, чтобы расчетное время было меньше необходимого.

***5.2 Требования к ВДТ, ЭВМ, ПЭВМ и периферийным устройствам***

Уровни физических факторов (уровни электромагнитных и электростатических полей, уровни вибрации, уровни ультрафиолетового, инфракрасного, видимого и мягкого рентгеновского излучений), создаваемые ПЭВМ и периферийными устройствами, не превышают предельно-допустимые уровни, установленных Гигиеническим нормативом «Предельно-допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами».

Уровни шума, создаваемые ПЭВМ и периферийными устройствами, не превышают предельно допустимых уровней, предусмотренных Гигиеническим нормативом, и устанавливаются в зависимости от следующих категорий производимых работ (табл. 5.4):

1. категория I – выполнение основной работы в залах вычислительной техники, а также в помещениях с ПЭВМ всех типов учреждений образования;
2. категория II – выполнение работы с ПЭВМ в помещениях, где работают инженерно-технические работники, осуществляющие лабораторный, аналитический или измерительный контроль;
3. категория III – выполнение работы в помещениях операторов ЭВМ (без дисплеев);
4. категория IV – выполнение работы с ПЭВМ в помещениях для размещения шумных агрегатов.

В производственных помещениях, в которых работа с ПЭВМ является вспомогательной, уровни шума на рабочих местах не превышают значений, установленных для видов трудовой деятельности, осуществляемых в этих помещениях, в соответствии с Санитарными нормами и правилами, устанавливающими ПДУ шума на рабочих местах, в помещениях жилых и общественных зданий.

Допустимые визуальные эргономические параметры устройств отображения ВДТ, ЭВМ и ПЭВМ соответствуют допустимым значениям, установленным Гигиеническим нормативом.

Конструкция ЭВМ и ПЭВМ, дизайн и совокупность эргономических параметров обеспечивают надежное и комфортное считывание отображаемой информации в условиях эксплуатации. Конструкция оборудования ЭВМ и ПЭВМ обеспечивает возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении для обеспечения фронтального наблюдения экрана.

Дизайн ВДТ, ЭВМ, ПЭВМ и периферийных устройств должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ВДТ, ЭВМ, ПЭВМ и периферийных устройств имеет матовую поверхность с коэффициентом отражения 0,4-0,6 и не имеет блестящих деталей, способных создавать блики.

# ЗАКЛЮЧЕНИЕ

В результате выполненного дипломного проекта был разработан пользовательский интерфейс веб-приложение, соответствующий предметной область, решающий все поставленные цели и задачи.

В ходе проектирования интерфейса приложения были проанализированы существующие аналоги, и с учетом их достоинств и недостатков, а также с учетом специфики предметной области, была разработана логическая модель функционирования приложения.

Преимущества созданного пользовательского интерфейса веб-приложения:

1. доступность на любом устройстве и любой операционной системе
2. лаконичный и интуитивно понятный дизайн;
3. охватывает все необходимые потребности для управления проектами;
4. возможность коммуникации между участниками команды;
5. безопасность – неавторизованный пользователь не сможет просматривать страницы веб-приложения.

Пользовательский интерфейс разработан с высоким уровнем удобства использования, простоты вхождения для новых пользователей, а также покрывает все необходимое требования, касающиеся данного типа приложений.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

уоруркп

шуопкшуо

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Конвейер обработки запросов | | | | | | | | | | | 65 |
|  | | | | | | | | | | | |
|  |  |  |  |  | ДП–3070111406–2019-03 | | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № документа | Подпись | Дата |
| Разраб. | | Вихарев |  |  | «Серверная часть с уровнем доступа к данным системы управления программными проектами Ziro на базе технологий ASP.NET Core» | Лит | | | Лист | Листов | |
| Руковод. | | Иванченко |  |  |  | У |  | 3 | 10 | |
| Консульт. | | Иванченко |  |  | 1-40 01 01  БНТУ, г. Минск | | | | | |
| Н.контр. | | Домаренко |  |  |
| Зав.каф. | | Полозков |  |  |
| Конвейер обработки запросов | | | | | | | | | | | 65 |
|  | | | | | | | | | | | |
|  |  |  |  |  | ДП–3070111406–2019-03 | | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № документа | Подпись | Дата |
| Разраб. | | Вихарев |  |  | «Серверная часть с уровнем доступа к данным системы управления программными проектами Ziro на базе технологий ASP.NET Core» | Лит | | | Лист | Листов | |
| Руковод. | | Иванченко |  |  |  | У |  | 3 | 10 | |
| Консульт. | | Иванченко |  |  | 1-40 01 01  БНТУ, г. Минск | | | | | |
| Н.контр. | | Домаренко |  |  |
| Зав.каф. | | Полозков |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Конвейер обработки запросов | | | | | | | | | | | 65 |
|  | | | | | | | | | | | |
|  |  |  |  |  | ДП–3070111406–2019-03 | | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № документа | Подпись | Дата |
| Разраб. | | Вихарев |  |  | «Серверная часть с уровнем доступа к данным системы управления программными проектами Ziro на базе технологий ASP.NET Core» | Лит | | | Лист | Листов | |
| Руковод. | | Иванченко |  |  |  | У |  | 3 | 10 | |
| Консульт. | | Иванченко |  |  | 1-40 01 01  БНТУ, г. Минск | | | | | |
| Н.контр. | | Домаренко |  |  |
| Зав.каф. | | Полозков |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Конвейер обработки запросов | | | | | | | | | | | 65 |
|  | | | | | | | | | | | |
|  |  |  |  |  | ДП–3070111406–2019-03 | | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № документа | Подпись | Дата |
| Разраб. | | Вихарев |  |  | «Серверная часть с уровнем доступа к данным системы управления программными проектами Ziro на базе технологий ASP.NET Core» | Лит | | | Лист | Листов | |
| Руковод. | | Иванченко |  |  |  | У |  | 3 | 10 | |
| Консульт. | | Иванченко |  |  | 1-40 01 01  БНТУ, г. Минск | | | | | |
| Н.контр. | | Домаренко |  |  |
| Зав.каф. | | Полозков |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Конвейер обработки запросов | | | | | | | | | | | 65 |
|  | | | | | | | | | | | |
|  |  |  |  |  | ДП–3070111406–2019-03 | | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № документа | Подпись | Дата |
| Разраб. | | Вихарев |  |  | «Серверная часть с уровнем доступа к данным системы управления программными проектами Ziro на базе технологий ASP.NET Core» | Лит | | | Лист | Листов | |
| Руковод. | | Иванченко |  |  |  | У |  | 3 | 10 | |
| Консульт. | | Иванченко |  |  | 1-40 01 01  БНТУ, г. Минск | | | | | |
| Н.контр. | | Домаренко |  |  |
| Зав.каф. | | Полозков |  |  |
| Конвейер обработки запросов | | | | | | | | | | | 65 |
|  | | | | | | | | | | | |
|  |  |  |  |  | ДП–3070111406–2019-03 | | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № документа | Подпись | Дата |
| Разраб. | | Вихарев |  |  | «Серверная часть с уровнем доступа к данным системы управления программными проектами Ziro на базе технологий ASP.NET Core» | Лит | | | Лист | Листов | |
| Руковод. | | Иванченко |  |  |  | У |  | 3 | 10 | |
| Консульт. | | Иванченко |  |  | 1-40 01 01  БНТУ, г. Минск | | | | | |
| Н.контр. | | Домаренко |  |  |
| Зав.каф. | | Полозков |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Конвейер обработки запросов | | | | | | | | | | | 65 |
|  | | | | | | | | | | | |
|  |  |  |  |  | ДП–3070111406–2019-03 | | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № документа | Подпись | Дата |
| Разраб. | | Вихарев |  |  | «Серверная часть с уровнем доступа к данным системы управления программными проектами Ziro на базе технологий ASP.NET Core» | Лит | | | Лист | Листов | |
| Руковод. | | Иванченко |  |  |  | У |  | 3 | 10 | |
| Консульт. | | Иванченко |  |  | 1-40 01 01  БНТУ, г. Минск | | | | | |
| Н.контр. | | Домаренко |  |  |
| Зав.каф. | | Полозков |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Конвейер обработки запросов | | | | | | | | | | | 65 |
|  | | | | | | | | | | | |
|  |  |  |  |  | ДП–3070111406–2019-03 | | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № документа | Подпись | Дата |
| Разраб. | | Вихарев |  |  | «Серверная часть с уровнем доступа к данным системы управления программными проектами Ziro на базе технологий ASP.NET Core» | Лит | | | Лист | Листов | |
| Руковод. | | Иванченко |  |  |  | У |  | 3 | 10 | |
| Консульт. | | Иванченко |  |  | 1-40 01 01  БНТУ, г. Минск | | | | | |
| Н.контр. | | Домаренко |  |  |
| Зав.каф. | | Полозков |  |  |