

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Общая часть	Error! Bookmark not defined.
1.1. Описание предметной области	4
1.2. Анализ методов решений	6
1.3. Функции проектируемой системы	7
1.4. Обоснование (выбор) методики, архитектуры, технологии и инструментальных средств проектирования и разработки	8
2 Специальная часть.....	Error! Bookmark not defined.
2.1. Контекстная диаграмма работы ИС.	17
2.2. UML диаграммы.....	18
2.3. Инфологическая модель	20
2.4. Описание интерфейсов	24
2.5. Описание работы.....	26
2.6. Тестирование	29
3 Организационно-экономическая часть	Error! Bookmark not defined.
3.1. Техничко-экономическое обоснование.	33
3.2. Организационная часть.....	34
3.3. Экономическая часть	36
3.4. Определение цены ИС для ИП "Гафуров Наиль Рахимьянович"	38
4 Охрана труда и техника безопасности	Error! Bookmark not defined.
4.1 Описание предметной области	41
4.2 Расчет искусственного освещения в помещении в ИП «Гафуров Наиль Рахимьянович», где находится рабочее место программиста.....	43
4.3 Электробезопасность в ИП «Гафуров Наиль Рахимьянович»	47
4.4 Пожарная безопасность в ИП «Гафуров Наиль Рахимьянович»	49
ЗАКЛЮЧЕНИЕ	51
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	52

ВВЕДЕНИЕ

Так как из за дня в день Российский рынок в сфере ИТ растёт стремительно, многие большие компании пытаются ускорить темп и сфокусировать внимание на разработке новых продуктов, поэтому они часто пользуются услугами студий, чтобы сократить объём работы и проблем. Чаще всего такие студии, мало персонала, а новых проектов на разработку информационных систем поступает много, поэтому много времени и внимание уделяться проектирование, что ни есть хорошо.

Введении автоматизации в этап проектирование не только ускорит сам процесс, но еще уменьшит ошибки в подсчетах сроках, цен, и сотрудники будут больше фокусировать внимание на разработку решений. Также если сам процесс проектирования станет удобным, станет больше проектов которые можно вести параллельную разработку , что повысит доход студии.

Цель проекта – разработка информационной системы которая автоматизирует этап проектирование разработки сервисов и программных продуктов.

Задачи, которые необходимо решить для достижения цели:

1. Провести анализ предметной области;
2. Определить функциональные требования;
3. Выполнить анализ методов решений;
4. Спроектировать автоматизированную систему;
5. Разработать систему;
6. Запустить прототип и провести тестирование.
7. Рассчитать экономическую эффективность, а также затрат средств во время разработки;
8. Провести анализ техники труда и безопасности.

1 Общая часть

1.1 Описание предметной области

1.1.1 Характеристика студии

Гафуров Наиль Рахимьянович является индивидуальным предпринимателем который основал веб студию " Gafurov Studio " (далее студия) и имеет там должность CEO. Данная студия занимается Разработками в области цифролизации и автоматизации: автоматизация медицинских учреждений, государственного сектора, интеллектуальные системы общения с пользователем (искусственный интеллект, нейронные сети), проекты в сфере финансовых технологий (финтех) - платежные системы, системы оплаты, системы билетирования.

Студия занимается каждый день разработкой новых и поддержкой старых информационных систем на заказ или как индивидуальные разработки.

Например, сервис "SALETEX" - онлайн продавец для сайта с функцией автоматического звонка клиенту на мобильный телефон[18].



Рисунок 1 – Главная страница сервиса " SALETEX "

Сервис автоматизации малого и среднего бизнеса "UNINSY" [19].

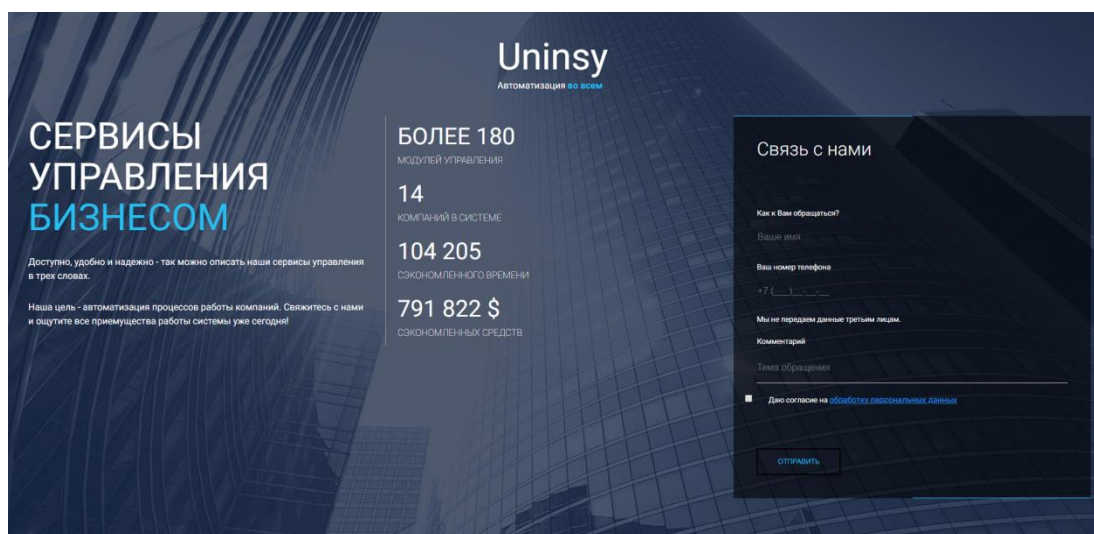


Рисунок 2 – Главная страница сервиса "UNINSY"

1.1.2 Описание алгоритма от заказа информационной системы до проектирование

Заказчик оформляет телефонный звонок через форму на сайте студии или договаривается через одну из зарегистрированных профилей студии в социальных сетях (Facebook, VK или Instagram) [20].

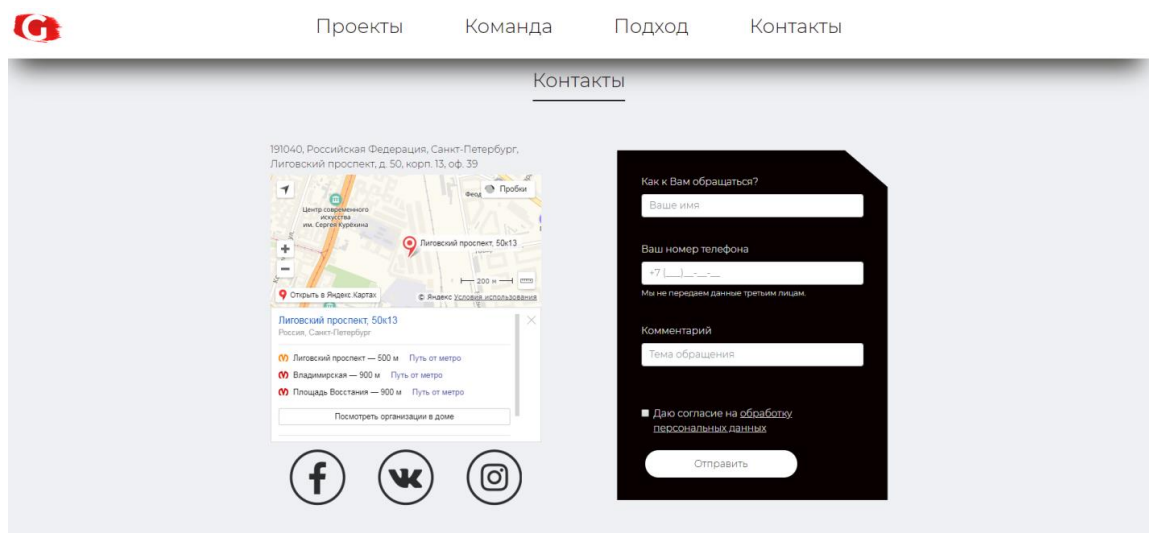


Рисунок 3 – Форма оформления звонка на главной страницы "Gafurov Studio", а также другая контактная информация

Маркетолог звонит заказчику предоставляет всю необходимую и интересующую информацию об услугах студии. Если заказчика удовлетворяют условия, то либо оформляется техническое задание на основе опроса заказчика, либо представляется техническое задание самим заказчиком. Далее на основе информации ТЗ создается архитектура системы в графическом редакторе, документации и идёт оценка, анализ ресурсов и методов решение проекта. После анализа проекта уточняются нюансы с заказчиком и идёт согласование на разработку проекта.

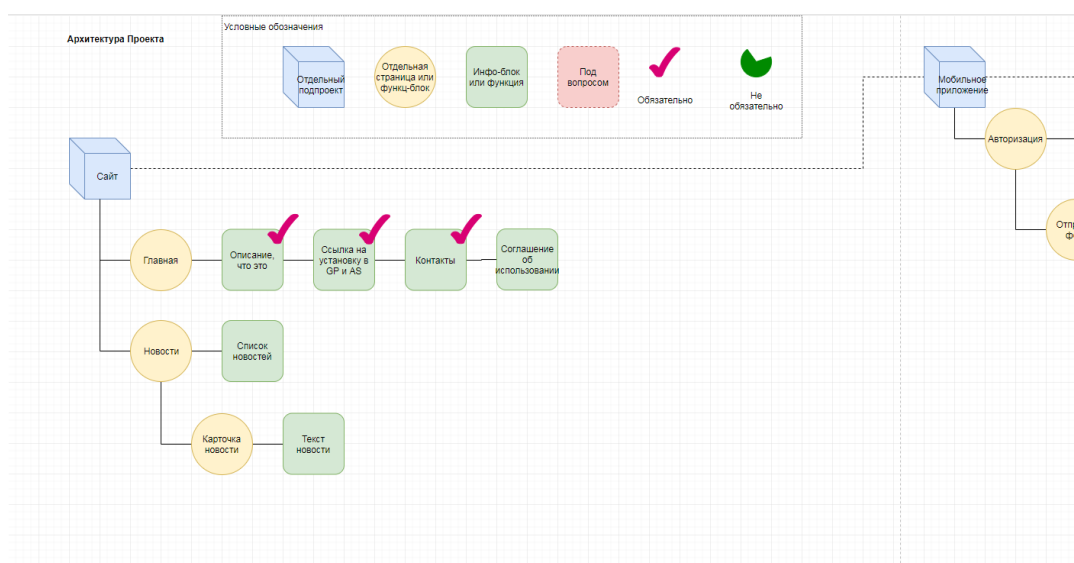


Рисунок 4 – Пример как выглядит архитектура проекта

1.2 Анализ методов решений

Оформлением заказа, проектированием и подсчётом сметы занимается в студии один человек. Из за этого скорость оформления новых заказов низкая, вероятность ошибиться в оформлении бумаг высокая, а также сама информация не структурирована и ориентироваться в несколько проектов одновременно сложно так как информация не централизована в одном месте.

Поэтому было решено найти решение, которое автоматизирует этот процесс.

Студия для решения этих использует следующие ПО:

- Веб сервис Draw.io для составления контекстных диаграмм архитектуры проекта;

- Google Таблицы для подсчёты сметы;

- Google Документы для составления тз и иных документов;

При анализе Case - средств на рынке не было найдено системы которые объединяло все этапы и автоматизировали их. Поэтому было решено разработать информационную систему.

1.3 Функции проектируемой системы.

Информационная система будет разработана как модульная. То есть будет делиться на подсистемы, для возможности дальнейшего расширения функций.

В информационной системе будут следующие модули:

- Подсистема профиль пользователя;

- Подсистема менеджер проектов;

- Подсистема редактор архитектуры.

Подсистема профиль пользователя представляет собой возможность авторизоваться в системе и сделать информацию который создаёт пользователей конфиденциальной. В системе будут следующие функции:

- Регистрация в системе;

- Авторизация;

- Настройка параметров.

Подсистема менеджер проектов представляет собой подсистему, которая хранит, а также позволяет редактировать информацию проектов. В этой системе будут следующие функции:

- Просмотр проектов;

- Редактирование (добавление новых/ редактирование созданных/ удаление) проектов;

- Редактирование информации проектов;

- Просмотр зависимых проектов (подпроекты/ сервисы);
- Редактирование (добавление новых/ редактирование созданных/ удаление) зависимых проектов (подпроекты/ сервисы);
- Редактирование информации зависимых проектов (подпроекты/ сервисы);

Подсистема редактор архитектуры будет отвечать за хранение информации об зависимости функциональных блоков и их функций, а также должна предоставлять специальный редактор для редактирование и просмотра этой архитектуры. Это система будет иметь следующие функции:

- Просмотр архитектуры;
- Редактирование (добавление новых/ редактирование созданных/ удаление) функциональных блоков;
- Редактирование информации функциональных блоков;
- Изменение зависимости функциональных блоков;
- Редактирование (добавление новых/ редактирование созданных/ удаление) функции для функциональных блоков;
- Редактирование информации функции;
- Создании сложно структурированной функции. То есть создания, редактирование и возможность просмотра архитектуры для функции;
- Сохранения архитектуры;

1.4 Обоснование (выбор) методики, архитектуры, технологии и инструментальных средств проектирования и разработки

1.4.1 Обоснование (выбор) методики

Информационная система будет разработана как веб сервис[1]. Причинной выбора этой технологии разработки стали следующими:

- В дальнейшем проектированием будут заниматься несколько сотрудников;

- Доступ к информации к проектам должна осуществляться не только с стационарного компьютера, но и также с мобильного устройства;
- В будущем планируется использовать систему не только как средство для локального использования внутри студии, но также для продажи услуг этих средств любому желающему.

1.4.2 Обоснование (выбор) архитектуры

Была выбрана архитектура MVC. MVC состоит из объектов трех видов. Модель - это объект приложения, а вид - экранное представление. Контроллер описывает, как интерфейс реагирует на управляющие воздействия пользователя[1].

Причиной выбора этой архитектуры стали следующими:

- Архитектура MVC разделяет код на различные области, обеспечивая лучшую организацию[2]
- MVC отделяет их друг от друга, за счет чего повышается гибкость и улучшаются возможности повторного использования[1].

1.4.3 Обоснование (выбор) технологии

Для разработки веб интерфейса были выбраны стандартные технологии.

JavaScript - это сценарный язык программирования, наиболее применяемый для создания интерактивных веб страниц. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений[3].

HTML — язык разметки, определяемая структура веб страницы[4]. HTML5 — улучшение уровня поддержки мультимедиа-технологий с одновременным сохранением обратной совместимости, удобочитаемости кода для человека и простоты анализа для парсеров.

CSS язык описания, который описывает как должен обрабатывать браузер объекты веб-документа, написанного с использованием языка разметки[5].

Этот язык используется создателями веб-страниц для определения цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы

Чтобы разработать интерфейс и ускорить время разработки было решено использовать готовые решения.

Для верстки был выбран фреймворк для интерфейсов Bootstrap.

Bootstrap предоставляет шаблоны дизайна, основанные на HTML и CSS для разметки, форм, кнопок, таблиц и много другое, а также дополнительные плагины JavaScript [6].

Для разработки работы интерфейса была выбрана JavaScript библиотека JQuery.

jQuery - бесплатная (под лицензией МГТ) библиотека JavaScript, созданная Джоном Резигом в 2006 году, разработанная для упрощения написания HTML-кода для клиентской стороны. Она обеспечивает быстрый обход и обработку HTML- документа, обработка событий, анимации и Ajax, благодаря API, работающему с множеством браузеров[7].

При выборе разработки сервисного приложения был провиден анализ программных решений. При анализе типичных программных продуктов, как планируемая разрабатываемая система, на рынке чаще всего использую следующие языки разработки сервисного приложения это PHP, Java и технологию NodeJs.

PHP (рекурсивный акроним словосочетания PHP: Hypertext Preprocessor) - это язык программирования общего назначения с открытым

исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML[8].

После анализа были выведены следующие преимущества:

– Много специалистов владеют этим языком. Согласно поиску кандидатов с открытой резюме на сервисе HeadHunter[17] (один из популярных русских сервисов интернет-рекрутмента), которые указали один из ключевых навыков знания PHP составляет 2950 (на момент написания отчета) кандидатов проживающих (готовы работать) в городе Санкт - Петербург. На рисунке 5 изображен результат поиска сотрудников.

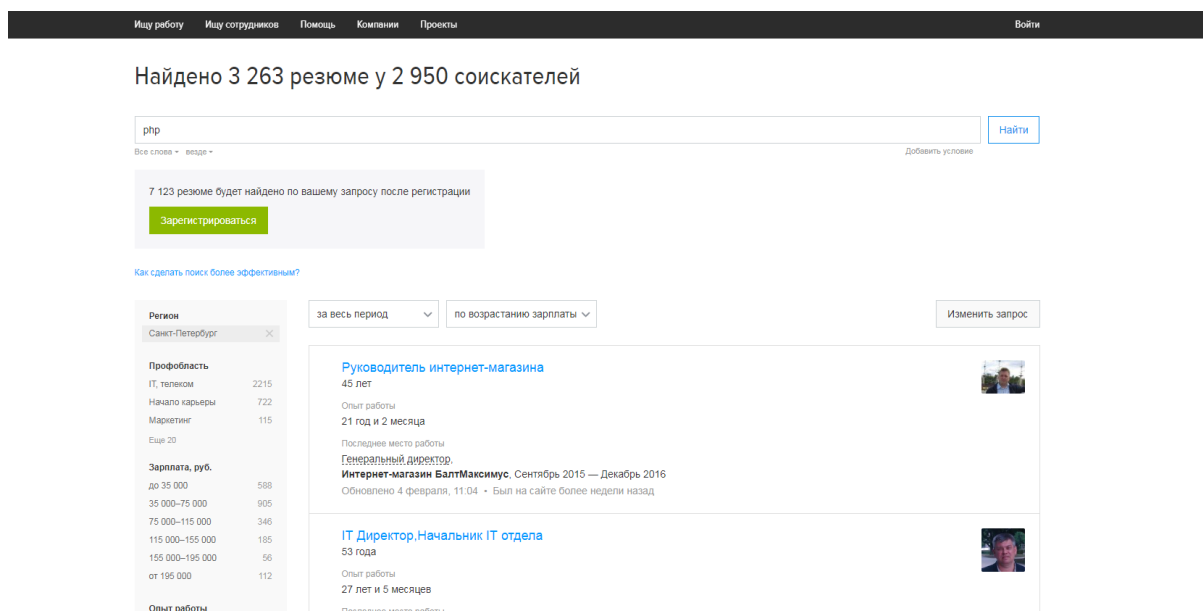


Рисунок 5 – Результат поиска свободных сотрудников, которые указали в списке ключевых навыков PHP, а также возможность работать в городе Санкт - Петербурге

– Исходя из простоты в изучении этого языка, специалист с данным ключевым навыком дешевые. Согласно полученным данным выше 390 кандидатов с опытом работы более одного года готов работать с зарплатой до 35 000 рублей в месяц.

– Много проверенных фреймворков и готовых решений. Так как php появился с 1995 году[8].

При анализе были выведены следующие недостатки.

- На этом языке не возможно писать асинхронный код. Так как php был разработан с 1995 году, и в этом времени не использовали асинхронные методы разработки, поэтому в движке и сам синтаксис не предполагал возможность данных функций. Без асинхронных методов нельзя гарантировать быструю скорость ответов на параллельные запросы от одного и того же пользователя.

- Не однозначная архитектура языка, синтаксиса и итоговое размещение готовых программ. Пример не однозначной архитектуры: в документации написаны пути решения исправления работы стандартной функции, то есть функции могут иметь не предполагаемый работу и обработку данных. Пример не однозначного синтаксиса будет сравнения переменных с помощью "==" . Движок не приводит к одному типу сравниваемых переменных, иногда переводит одну переменную в число, если вторая сравниваемая переменная имеет тип число. Пример не однозначного размещения итого приложения является в том что нет четкой структуры построения вспомогательных файлов для размещения на сервер приложения[9].

Технология NodeJs. Node js не язык программирование, это серверная JavaScript-платформа, предназначенная для создания масштабируемых распределенных сетевых приложений, использующая событийно-ориентированную архитектуру и не блокирующее асинхронное взаимодействие. Она основана на JavaScript движке V8 и использует этот же JavaScript для создания приложений[10].

После анализа были выведены следующие преимущества:

- Меньше сотрудников надо нанимать для поддержки данного проекта. Так как если сервисное приложение будет разработано на языке JavaScript, то нанятые сотрудники со знанием данного языка могут не только поддерживать и разрабатывать функции на сервисном приложении, но также и на клиентском приложении (интерфейсе)

– Данная технология разработана для использования методов асинхронного кода. Что увеличит скорость ответа на параллельные запросы от клиента и сократит потребляемой мощности сервера.

При анализе были выведены следующие недостатки.

– Нет много разработчиков со знанием Node технологии. Согласно поиску кандидатов с на сервисе HeadHunter, пользователи которые указали один из ключевых навыков знания NodeJs составляет 593 проживающих (готовы работать) в городе Санкт - Петербург. На рисунке 6 изображен результат поиска сотрудников.

Найдено 633 резюме у 593 соискателей

node

Найти

Все слова - везде

Добавить условие

1 317 резюме будет найдено по вашему запросу после регистрации

Зарегистрироваться

Как сделать поиск более эффективным?

Регион

Санкт-Петербург

Профобласть

IT, телеком 504

Начало карьеры 121

Производство 3

Еще 5

Зарплата, руб.

до 60 000 129

60 000–120 000 130

120 000–180 000 76

180 000–240 000 44

240 000–305 000 23

от 305 000 12

Опыт работы

Более 6 лет 266

От 3 до 6 лет 130

От 1 года до 3 лет 123

Нет опыта 116

за весь период

по соответствию

Изменить запрос

Программист

34 года

150000 руб.

Опыт работы

9 лет и 7 месяцев

Последнее место работы

Разработчик Node.js.

ООО "Центр Технологий и Медиа Групп", Январь 2018 — по настоящее время

Обновлено 8 июня, 22:44 • Был на сайте сегодня

Junior Javascript developer / Node.js

30 лет

Опыт работы

2 года и 6 месяцев

Последнее место работы

Junior front-end js developer.

Wintega.com, Сентябрь 2018 — Апрель 2019

Обновлено 7 июня, 10:35 • Был на сайте сегодня

Рисунок 6 – Результат поиска свободных сотрудников, которые указали в списке ключевых навыков NodeJs, а также возможность работать в городе Санкт - Петербурге

– Кандидаты со знанием Node технологии дорогие. Согласно полученным данным выше 90 кандидатов с опытом работы более одного года готовы работать с зарплатой до 65 000 рублей.

Технология Java.

Язык программирования Java, разработанный компанией Sun Microsystems под руководством Джеймса Гослинга и Билла Джоя, был создан как машинно-независимый язык программирования, который достаточно безопасен для сетевых применений и достаточно мощен, чтобы заменить родной исполняемый код[11].

После анализа были выведены следующие преимущества:

– Много специалистов владеют этим языком. Согласно поиску кандидатов на сервисе HeadHunter, которые указали один из ключевых навыков знания PHP составляет 2763 кандидатов проживающих (готовы работать) в городе Санкт - Петербург. На рисунке 7 изображен результат поиска сотрудников.

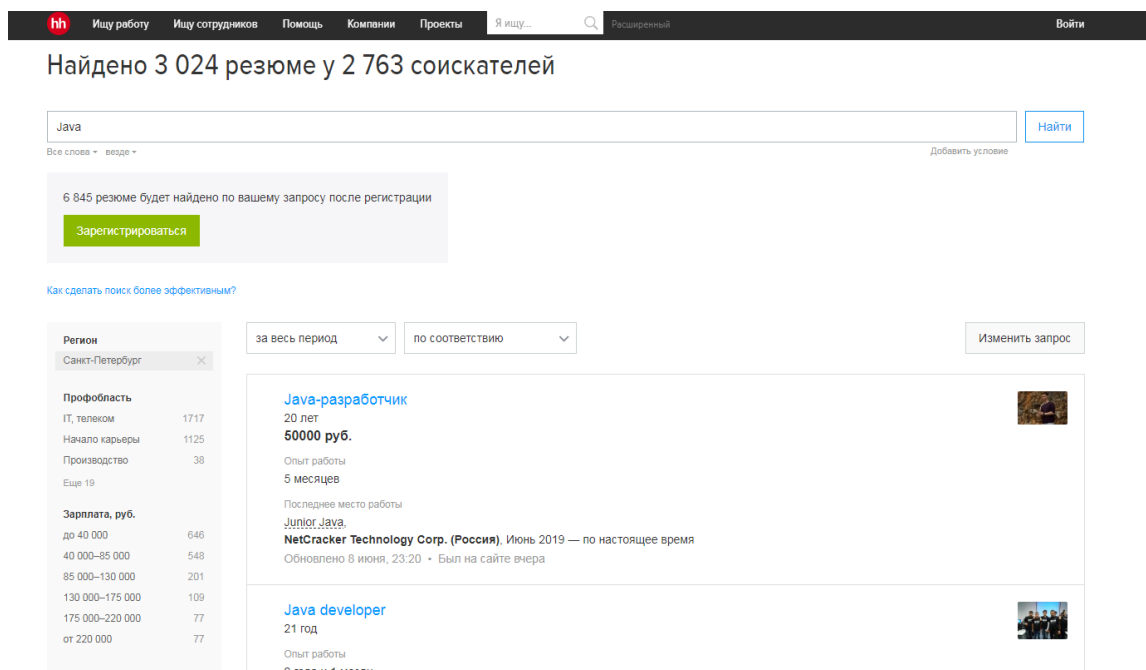


Рисунок 7 – Результат поиска свободных сотрудников, которые указали в списке ключевых навыков Java, а также возможность работать в городе Санкт - Петербурге

При анализе были выведены следующие недостатки.

– Чтобы разрабатывать на Java и использовать связанные технологии в коммерческих целях надо приобрести лицензию. 16 апреля 2019 года изменение вступило в силу. Теперь все опубликованные после этой

даты сборки Java SE можно использовать бесплатно только для личных нужд и с целью разработки. Для использования в коммерческих целях надо оформить платную подписку у Oracle[12].

После проведения сравнения преимуществ и недостатков у сравниваемых языков программирования, был выбран для разработки этого проекта PHP по следующим причинам:

- Много специалистов можно найти в Санкт Петербурге со знанием этого языка, в отличии от NodeJs. Что значит не придется тратить деньги для повышения квалификации нанятых сотрудников;

- Кандидаты со знанием этого языка самые дешевые по сравнению с конкурентами. Так как веб-студия не предоставляет много средств для разработки, это преимущество выделяют среди остальных конкурентов.

Чтобы решить недостатки выбранного языка были предприняты следующие решения:

- Так как на PHP нельзя писать асинхронный код, было решено поднять несколько удаленных серверов для решения проблемы медленной скорости ответа на запросы клиента;

- Чтобы решить проблемы с архитектурой языка, что может намного увеличить время разработки и тестирования, приложение будет разработано на готовом фреймворке. Использование фреймворка в проекте, сократит срок разработки, предоставит готовый каркас программы, что уменьшит возможность ошибиться и появления ошибок в готовой программе, и готовый способ размещения приложения на сервере

Был выбран фреймворк Fat-free. Fat-Free Framework - это веб-инфраструктура с открытым исходным кодом, распространяемая по общедоступной лицензии GNU и размещенная в GitHub и Sourceforge. Программное обеспечение стремится объединить полный набор функций с облегченной кодовой базой, в то же время будучи простым в изучении, использовании и расширении[13].

Для хранения, обработки, проведение других взаимодействий с информацией была выбрана СУБД MySQL. MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с Sun Microsystems. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации[14].

1.4.4 Обоснование (выбор) инструментальных средств разработки и проектировании

Были выбраны следующие инструменты для разработки кода:

– 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[15].

– Git — распределённая система управления версиями. Git поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки[16].

2 Специальная часть

2.1 Контекстная диаграмма работы ИС

На рисунке 8 изображена контекстная диаграмма системы.

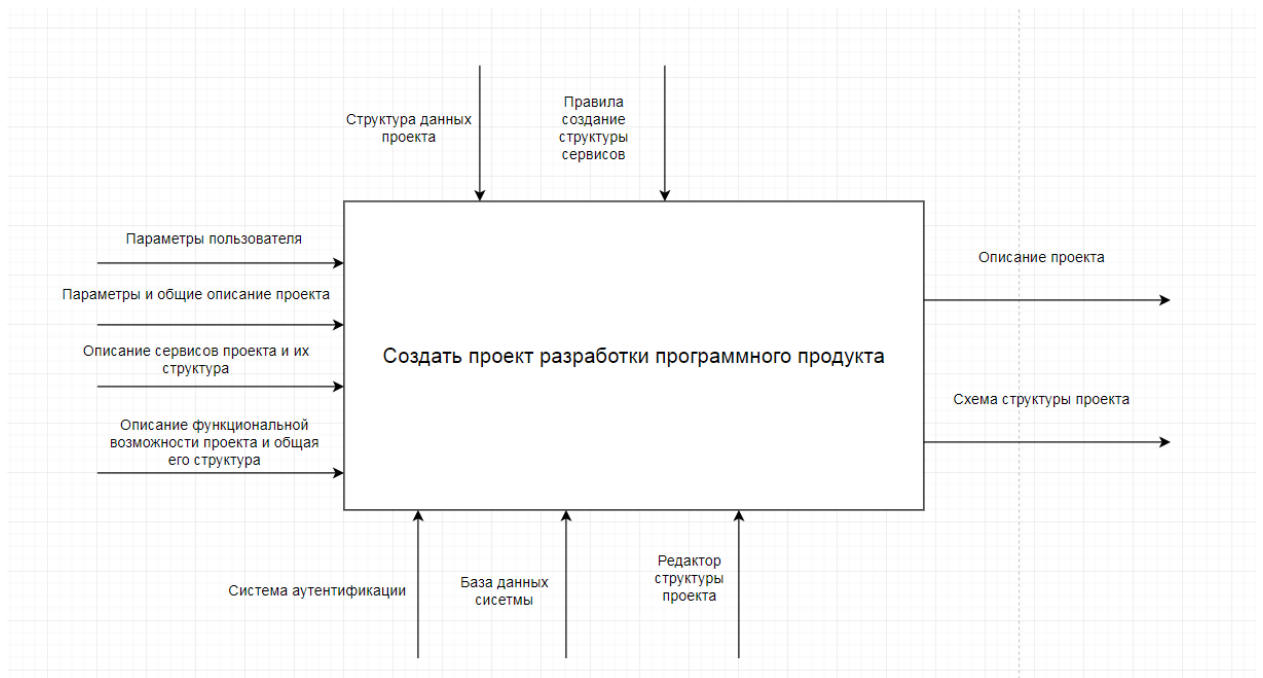


Рисунок 8 - Контекстная диаграмма "Создать проект разработки программного продукта"

2.2 UML-диаграммы

Также разработаны диаграммы в методологии UML. На рисунках 9 и 10 изображена диаграмма использования.

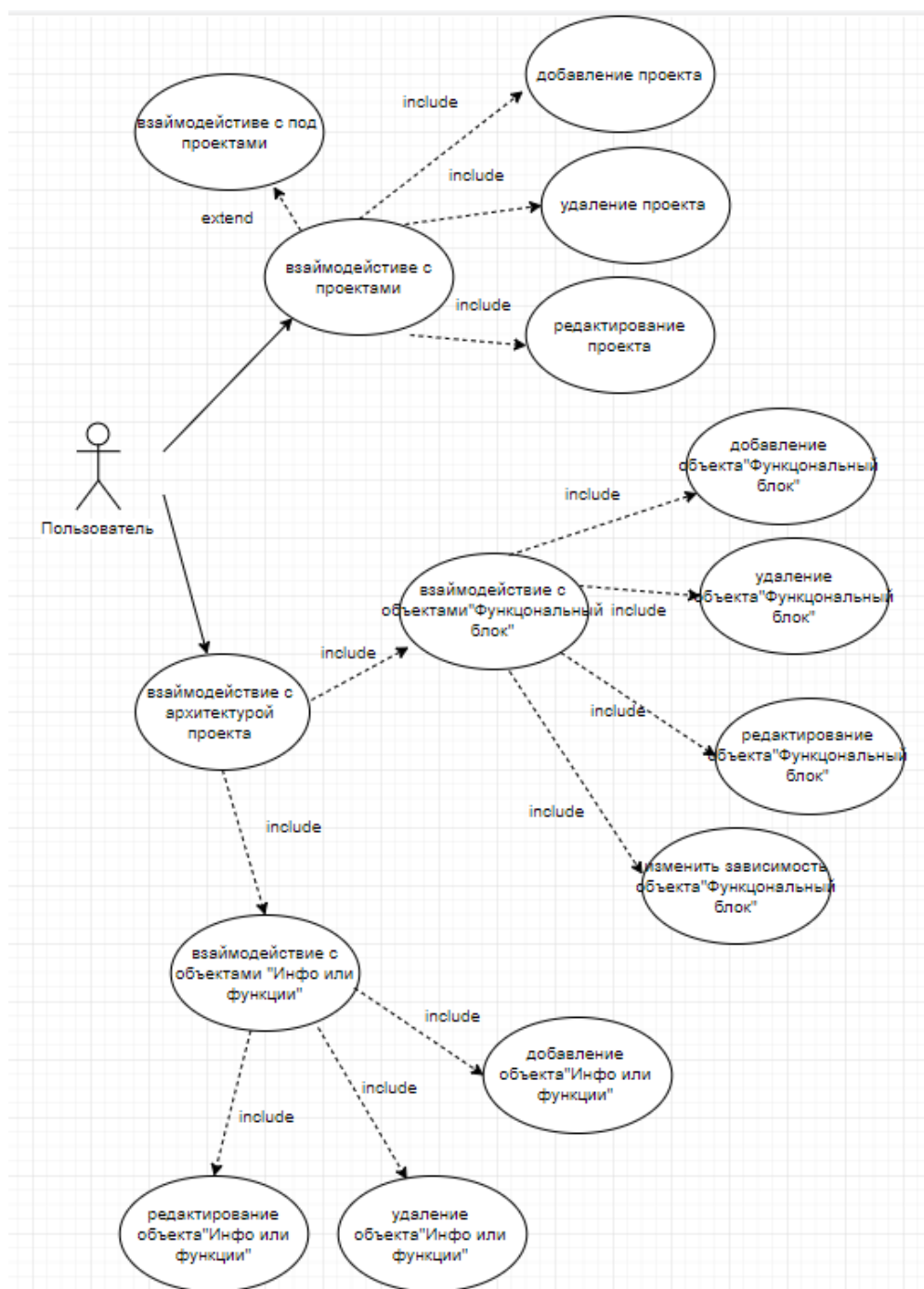


Рисунок 9 - Диаграмма использования для зарегистрированного пользователя в системе

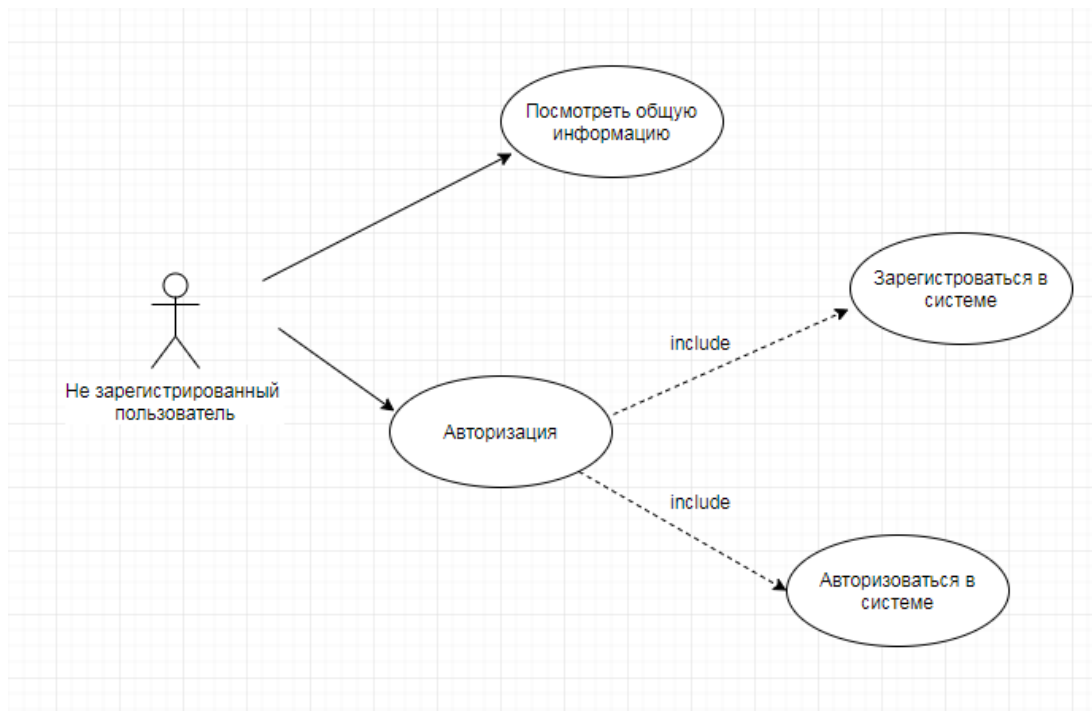


Рисунок 10 - Диаграмма использования для не зарегистрированного пользователя в системе

На рисунках 11 и 12 диаграмма последовательности.

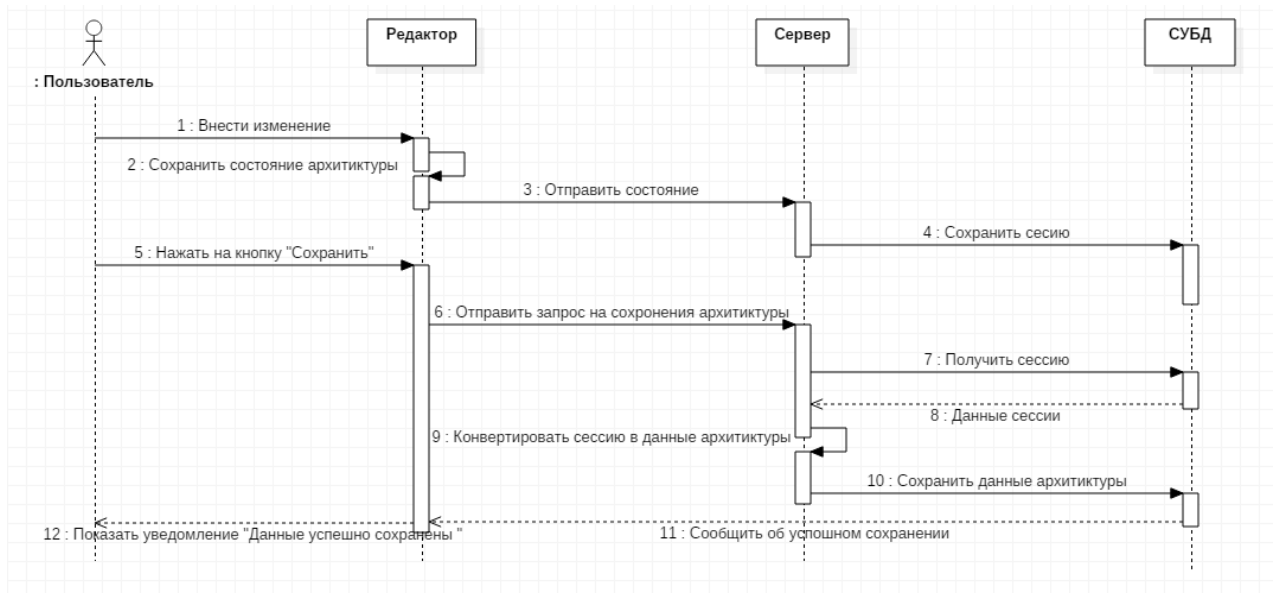


Рисунок 11 - Диаграмма последовательности изображающая принцип работы менеджера проекта

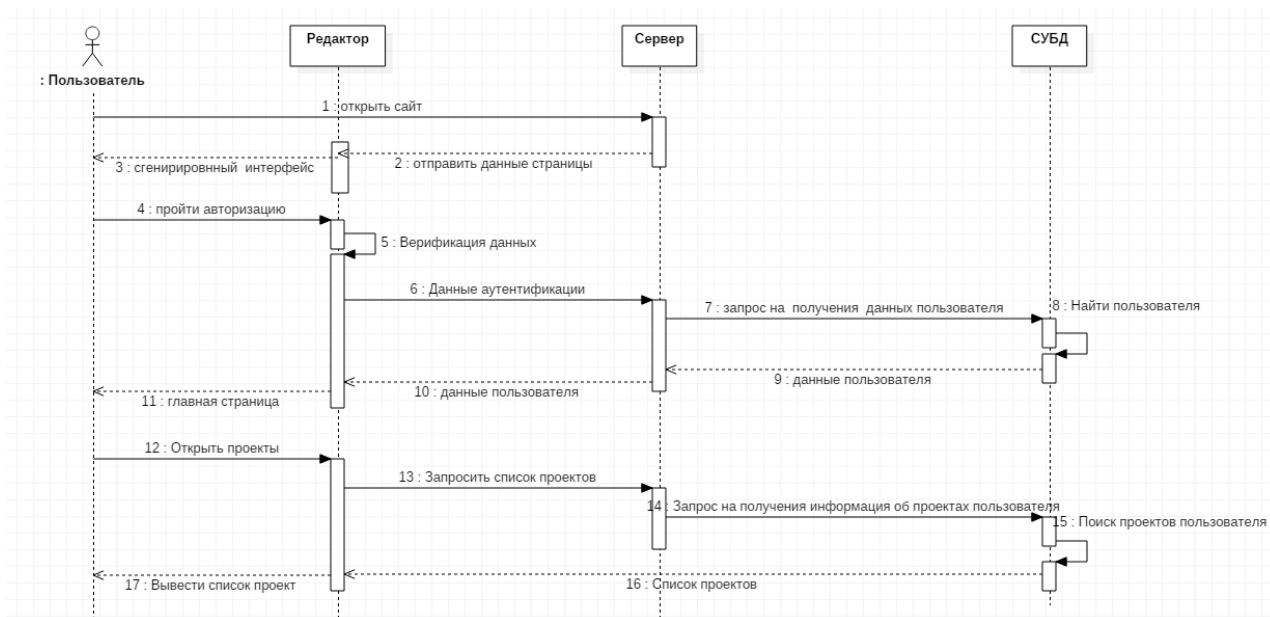


Рисунок 12 - Диаграмма последовательности изображающая принцип работы менеджера проекта

2.3 Инфологическая модель

На рисунке 13 схема структуры базы данных системы.

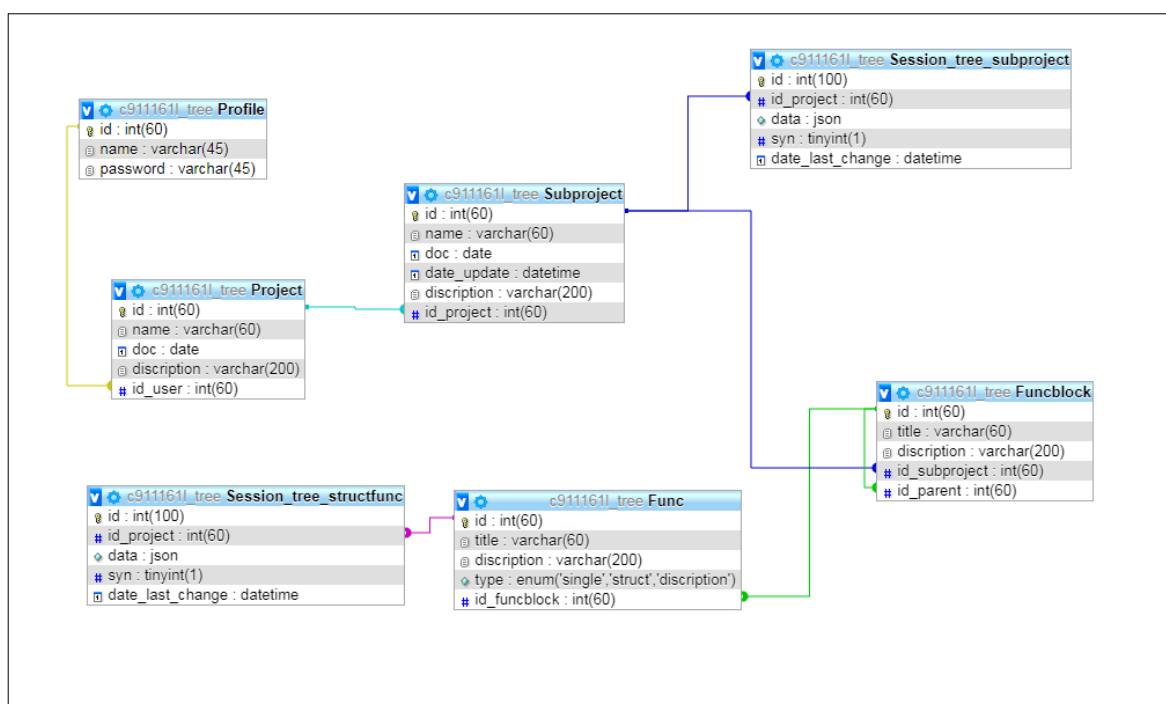


Рисунок 13 - Инфологическая модель информационной системы

На рисунке 14 схема сущности Профиль (оригинальное название "Profile"). В данной сущности хранятся данные пользователя в том числе данные для авторизации.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(60)			Нет	Нет		AUTO_INCREMENT
2	name	varchar(45)	utf8_general_ci		Нет	Нет		
3	password	varchar(45)	utf8_general_ci		Нет	Нет		

Рисунок 14 - Сущность Профиль (оригинальное название "Profile")

На рисунке 15 схема сущности Проект (оригинальное название "Project"). В данной сущности хранятся данные проектов которые пользователь создал.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(60)			Нет	Нет		AUTO_INCREMENT
2	name	varchar(60)	utf8_general_ci		Нет	Нет		
3	doc	date			Нет	Нет		
4	discription	varchar(200)	utf8_general_ci		Да	NULL		
5	id_user	int(60)			Нет	Нет		

Рисунок 15 - Сущность Проект (оригинальное название "Project")

На рисунке 16 схема сущности Сервисы (оригинальное название "Subproject"). В данной сущности записаны данные сервисов (подпроекты).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(60)			Нет	Нет		AUTO_INCREMENT
2	name	varchar(60)	utf8_general_ci		Нет	Нет		
3	doc	date			Нет	Нет		
4	date_update	datetime			Да	NULL		
5	discription	varchar(200)	utf8_general_ci		Да	NULL		
6	id_project	int(60)			Нет	Нет		

Рисунок 16 - Сущность Сервисы (оригинальное название "Subproject")

На рисунке 17 схема сущности Функциональный блок (оригинальное название "Funcblock"). В данной сущности записаны данные функциональных блоков сервисов.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(60)			Нет	Нет		AUTO_INCREMENT
2	title	varchar(60)	utf8_general_ci		Нет	Нет		
3	discription	varchar(200)	utf8_general_ci		Да	NULL		
4	id_subproject	int(60)			Нет	Нет		
5	id_parent	int(60)			Да	NULL		

Рисунок 17 - Сущность Функциональный блок (оригинальное название "Funcblock")

На рисунке 18 схема сущности Функции (оригинальное название "Func"). В данной сущности записаны данные функции функционального блока сервисов.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(60)			Нет	Нет		AUTO_INCREMENT
2	title	varchar(60)	utf8_general_ci		Нет	Нет		
3	discription	varchar(200)	utf8_general_ci		Да	NULL		
4	type	enum('single', 'struct', 'discription')	utf8_general_ci		Нет	Нет		
5	id_funcblock	int(60)			Нет	Нет		

Рисунок 18 - Сущность Функции (оригинальное название "Func")

На рисунке 19 схема сущности Сессия хранения структуры сервиса (оригинальное название "Session_tree_subproject"). В данной сущности записаны данные сессии структуры сервиса.



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 	int(100)			Нет	Нет		AUTO_INCREMENT
2	id_project 	int(60)			Нет	Нет		
3	data	json			Нет	Нет		
4	syn	tinyint(1)			Нет	0		
5	date_last_change	datetime			Нет	Нет		

Рисунок 19 - Сущность Сессия хранения структуры сервиса
(оригинальное название "Session_tree_subproject")

На рисунке 20 схема сущности Сессия хранения структуры сложноструктурной функции (оригинальное название "Session_tree_structfunc"). В данной сущности записаны данные сессии структуры сложноструктурной функции.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 	int(100)			Нет	Нет		AUTO_INCREMENT
2	id_project 	int(60)			Нет	Нет		
3	data	json			Нет	Нет		
4	syn	tinyint(1)			Нет	0		
5	date_last_change	datetime			Нет	Нет		

Рисунок 20 - Сессия хранения структуры сложноструктурной функции
(оригинальное название "Session_tree_structfunc")

2.4 Описание интерфейсов

На рисунке 21 интерфейс "Мои проекты". В интерфейсе пользователь может создавать новые проекты, удалять существующие или открыть проект для дальнейшего редактирования и просмотра информации

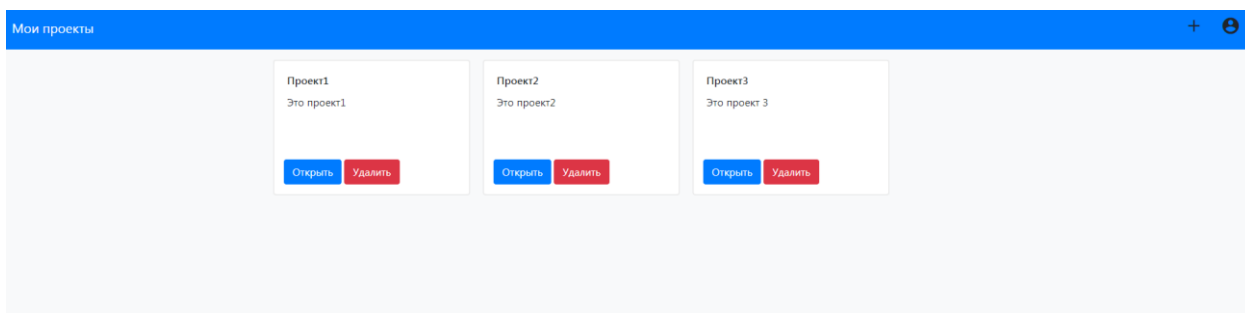


Рисунок 21 - Интерфейс "Мои проекты"

После нажатии кнопки "Открыть" на одном из проектов, откроется интерфейс "Проект" как показано рисунке 22. По умолчанию открывается вкладка "Описание". В этом окне пользователь может увидеть основную информацию об выбранном проекте.

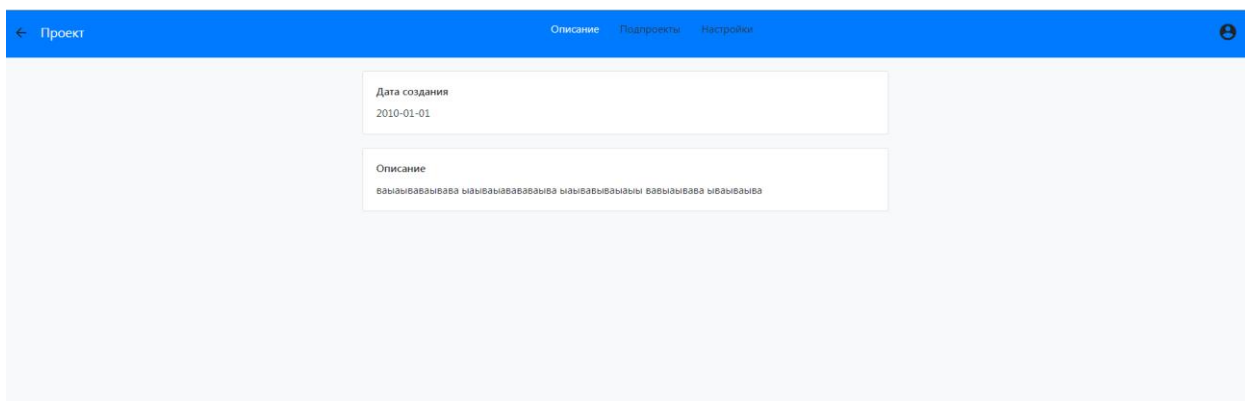


Рисунок 22 - Интерфейс "Проект" вкладка "Описание"

После нажатии вкладки "Подпроекты" сверху по середине меню, интерфейс "Проект" будет выглядеть как показано рисунке 23. Данный под интерфейс имеет принцип работы как у "Мои проекты", но вместо проектов в нем можно редактировать список зависимых подпроектами.

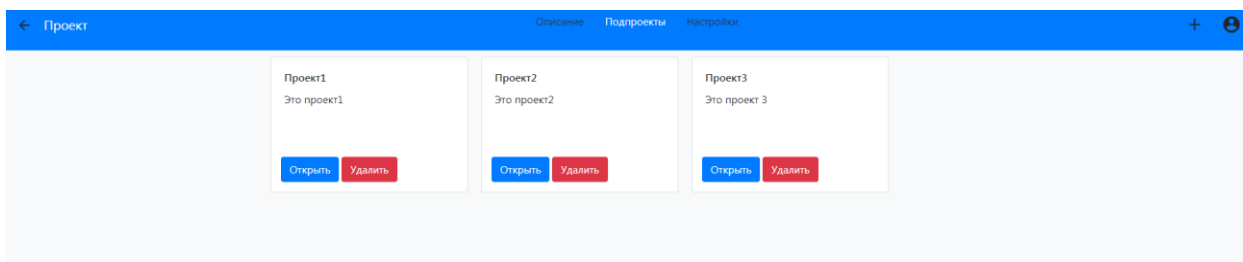


Рисунок 23 - Интерфейс "Проект" вкладка "Подпроект"

После нажатии вкладки "Настройки" откроется интерфейс где можно настроить параметры проекта.

При открытии подпроекта откроется интерфейс с точно таким же функционалом как у интерфейса "Проект", но отличие будет заключаться не только вывода информации подпроекта, но также будет отсутствовать вкладка "Подпроекты" и во вкладке "Настройки" будет функция изменить структуру. При нажатии кнопки изменить структуру откроется интерфейс как на рисунке 24.

В интерфейс "Редактор структуры подпроекта (сервиса)" можно изменить структур сервиса. Редактировать функциональные блоки, изменить зависимость друг от друга функциональных блоков, редактировать функции у функциональных блоков.

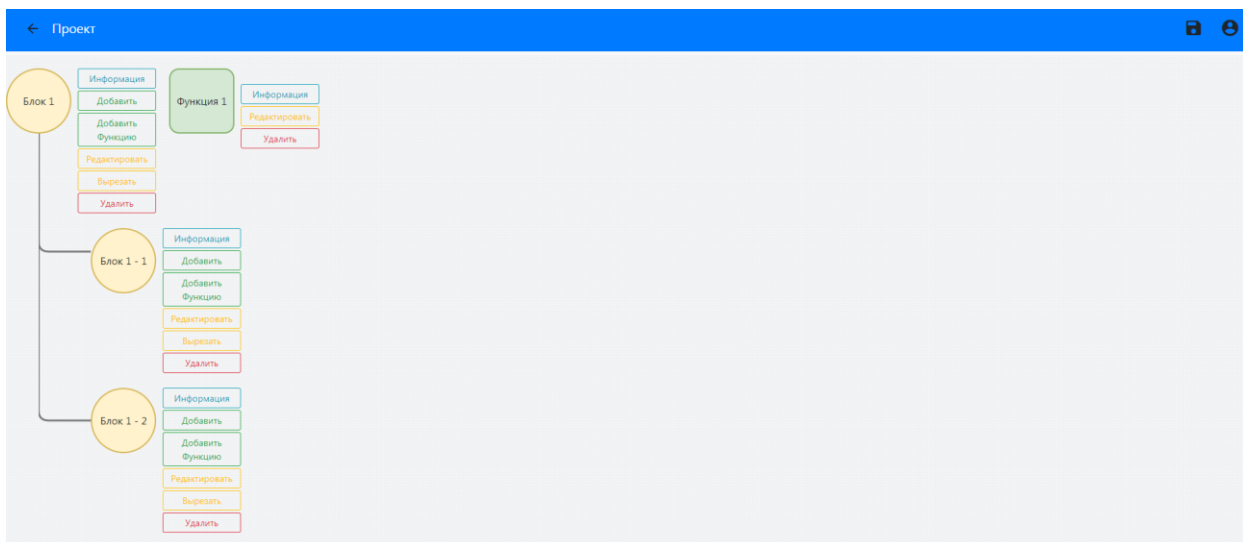


Рисунок 24 - Интерфейс "Редактор структуры подпроекта (сервиса)"

2.5 Описание работы

Приложение было разработано по принципу веб сервиса. Где хранятся данные и проводятся над ними операции в централизованном месте, в сервере. А клиенты получают набор функций при помощи авторизации в предоставляемом интерфейсе, по соединению интернета, сайта.

Приложение можно разделить на 3 основных модуля:

- Модель. Model.
- Интерфейс. View
- И связующие модуль Контролер. Controller

Модуль модель и контроллер это серверное приложение.

В качестве проверки работоспособности прототипа был запущен на сервисе Beget, под бесплатном выделенном удаленном сервере с готовым пакетом программ и предоставляемым собственным интерфейсом для управления и настройки некоторых конфигураций [21].

Файлы кода были помещены в новую созданную папку через файловый менеджер в подпапку `public_html`, как показано на рисунке 25.

Локальный сервер ▾		../c911161l.beget.tech/public_html/diplom				
Имя ▴		Тип	Размер	Владелец	Атри...	Изменён
..		[DIR]				
app			[DIR]	c911161l	700	24.01.2019 15:15...
lib			[DIR]	c911161l	700	24.01.2019 15:14...
tmp			[DIR]	c911161l	700	24.01.2019 15:14...
.htaccess			509 ...	c911161l	600	05.02.2019 15:39...
index		php	1.2 KB	c911161l	600	05.02.2019 15:42...

Рисунок 25 - Файлы программного кода проекта

База данных была создана через интерфейс панели администрирование СУБД. Чтобы перенести спроектированную базу данных на сервер была использована функция импорта, как показана рисунке 26 и 27.

Структура
SQL
Поиск
Запрос по шаблону
Экспорт
Импорт

Импорт в базу данных "c911161l_tree"

Импортируемый файл:

Файл может быть сжат в архив (gzip, bzip2, zip) или находиться без сжатия.
Имя сжатого файла должно заканчиваться в виде **.[формат].[сжатие]**. Пример: **.sql.zip**

Обзор вашего компьютера: Файл не выбран (Максимальный размер: 999МБ)

Вы также можете просто перетащить файл на любой странице.

Кодировка файла:

Рисунок 26 - Функция импорта в панели администрирование СУБД

2.6 Тестирование

Для проверки правильной работоспособности проекта надо протестировать функции. Был выбран ручной способ тестирования.

Ниже представлены тест кейсы, по которым была протестирована программа, в таблице 1, 2.

Таблица 1 - Тест кейс "Проверка модуля менеджера проекта"

Author: Зыльков Д.А.	Spec ID: T1	Priority: 1	Producer: Зыльков Д.А.	Developer: Зыльков Д.А.
OVERVIEW: Проверка функции добавления новых проектов				
GLOBAL SETUP and ADDITIONAL INFO: 1. Открыть сайт 2. Ввести данный для авторезации и нажать на кнопку "Войти"				
TC ID/Priority		T1-1	1	
IDEA: Проверка добавление нового проекта SETUP and ADDITIONAL INFO: Название проекта "Новый проект" Описание проекта "Описание нового проекта"				
Revision History				
Created on: 16.05.2019 by Зыльков Д. А.			"Проверка добавлении нового проекта " тест кейс	
Execution Part				
PROCEDURE			EXPECTED RESULT	
Нажать на кнопку "Добавит проект"			Выведен диалоговое окна "Добавит новый проект"	
Вести название проекта в поле "Название"				
Вести описание проекта в поле "Описание"				
Нажать на кнопку "Добавить"			В списке добавился новый проект	
TC ID/Priority		T1-2	2	
IDEA: Проверка валидации данных для добавление нового проекта SETUP and ADDITIONAL INFO: Описание проекта "Описание нового проекта"				
Revision History				
Created on: 16.05.2019 by Зыльков Д. А.			"Проверка валидации данных для добавлении нового проекта" тест кейс	
Execution Part				
PROCEDURE			EXPECTED RESULT	
Нажать на кнопку "Добавит проект"			Выведен диалоговое окна "Добавит новый проект"	
Нажать на кнопку "Добавить"			Вывод ошибки "Поле название должно иметь > 10 символов"	

Таблица 2 - Тест кейс "Проверка модуля менеджера проекта"

Author: Зыльков Д.А.	Spec ID: T2	Priority: 2	Producer: Зыльков Д.А.	Developer: Зыльков Д.А.
OVERVIEW: Проверка функции редактирование структуры сервиса				
GLOBAL SETUP and ADDITIONAL INFO: 1. Открыть сайт 2. Ввести данный для авторизации и нажать на кнопку "Войти" 3. Создать проект с названием "Тест проект". Нажать на кнопку "Открыть" на этом проекте. 4. Создать подпроект с названием "Тест сервис". Нажать на этом подпроекте кнопку "Открыть" 5. Открыть настройки и нажать на кнопку "Изменить архитектуру"				
TC ID/Priority		T2-1	1	
IDEA: Проверка добавление функционального блока. SETUP and ADDITIONAL INFO: Название функционального блока "Новый фб"				
Revision History				
Created on: 16.05.2019 by Зыльков Д. А.			"Проверка добавлении нового функционального блока " тест кейс	
Execution Part				
PROCEDURE			EXPECTED RESULT	
Нажать на кнопку "Добавить функциональный проект"			Выведен диалоговое окна "Добавит новый функциональный блок"	
Вести название функционального блока				
Нажать на кнопку "Добавить"			Выведен новый функциональй блок	
TC ID/Priority		T2-2	2	
IDEA: Проверка добавлении зависимого функционального блока SETUP and ADDITIONAL INFO: Название функционального блока "Новый фб 1-1"				
Revision History				
Created on: 16.05.2019 by Зыльков Д. А.			"Проверка добавлении нового зависимого функционального блока " тест кейс	
Execution Part				
PROCEDURE			EXPECTED RESULT	
Нажать на кнопку "Добавить" в инструменте функционального блока " Новый фб "			Выведен диалоговое окна "Добавит новый проект"	
Ввести название функционального				

блока		
Нажать на кнопку "Добавить"		Выведен новый зависимый функциональный блок
TC ID/Priority	T2-3	3
IDEA: Проверка функции перемещения SETUP and ADDITIONAL INFO: функционального блок для перемещения "Новый фб 1-1"		
Revision History		
Created on: 16.05.2019 by Зыльков Д. А.		"Проверка перемещение функционального блока " тест кейс
Execution Part		
PROCEDURE		EXPECTED RESULT
Нажать на кнопку "Вырезать" в инструменте функционального блока		Нажатый функциональный блок выделился. У остальных функциональных блоков и в начале дерева объектов появилась кнопка "Вставить"
Нажать на кнопку "Вставить" в начале дерева		Нажатый функциональный блок переместился
Нажать на кнопку "Добавить"		Выведен новый зависимый функциональный блок

3.1 Техничко-экономическое обоснование

В данном разделе дается технико-экономическое основание разработки ИС "Автоматизация проектирование ИС" для ИП "Гафуров Наиль Рахимьянович".

Цель работы: Разработать ИС, которая выполняет следующие функции:

- Предоставляет возможность авторизоваться под зарегистрированной учётной записи;
- Предоставляет возможность редактировать список проектов пользователя;
- Предоставляет возможность просматривать и редактировать информацию об проекте;
- Предоставляет возможность просматривать и редактировать модули проекта (сервисы / подпроекты)
- Предоставляет возможность редактировать и просматривать архитектуру сервиса в специальном разработанном интерфейсе редактора.

3.2 Организационная часть

Таблица 3 – Этапы и трудоемкость разработки

Этапы работ	Содержание работ	Исполнители			Время выполнения , дни
		Кол-во, чел.	Должность	Оклад, руб.	
Подготовительный	Ознакомление с заданием и выбор группы исполнителей	1	Инженер - программист	35000	1
Подготовительный	Разработка технического задания	1	Инженер - программист	35000	2
Подготовительный	Анализ методов решений	1	Инженер - программист	35000	3
Разработка	Установка и настройка требуемого ПО	1	Инженер - программист	35000	2
Разработка	Создание БД	1	Инженер - программист	35000	2
Разработка	Создание интерфейса	1	Инженер - программист	35000	10
Разработка	Создание серверного приложения	1	Инженер - программист	35000	5
Завершающий	Тестирование	1	Инженер - программист	35000	3
Завершающий	Оформление документации	1	Инженер - программист	35000	2
Всего (общая трудоемкость выполнения разработки):					30

В смету затрат включаются расходы:

- затраты на материалы, расходуемые в процессе проектирования;
- заработная плата работников, занятых разработкой;
- страховые взносы;
- прочие расходы (95% от заработной платы или по данным предприятия).

Затраты на материалы определяются в таблице 4.

Таблица 4 – Затраты на материалы

Наименование материалов	Ед. изм.	Количество	Цена, руб.	Стоимость (без НДС), руб.
Бумага для принтера Svetocopy, A4, 500 листов	пачка	1	230	230
Ручка шариковая Stabilo STABLiner 308, синяя	шт.	2	55	110
Карандаш чернографитный Stabilo Othello с ластиком, твердость HB	шт.	1	35	35
Клейкие стикеры Auchan, 100 листов, 75х75 см	пачка	1	30	30
USB Флеш-накопитель Exployn 580, 16 Гб	шт.	1	360	360
Итого затрат на материалы				765
Транспортные расходы по доставке материалов (15% от итого затрат на материалы).				115
Всего затрат на материалы с учетом транспортных расходов по доставке				880

Заработную плату можно определить в таблице 5.

Таблица 5– Заработная плата работников

Должность	Оклад, руб./мес.	Оплата в день, руб.	Трудоемкость, дни	Основная заработная плата, руб.
Инженер - программист	35000	1129	30	33871
Итого основная заработная плата				33871
Дополнительная заработная плата (12% от основной заработной платы)				4065
Всего заработная плата (сумма основной и дополнительной заработной платы)				37936
Страховые взносы(10% от заработной платы)				3794

Таблица 6 – Смета затрат на разработку ИС для ИП "Гафуров Наиль Рахимьянович"

Наименование затрат	Сумма, руб.
Затраты на материалы	880
Заработная плата	37936
Страховые взносы	3794
Прочие расходы	7500
Итого затрат	50110

3.3 Экономическая часть

В экономической части определяется себестоимость создания ИС оборудования в офисе

В себестоимость создания проекта входят:

- затраты на техническое и программное обеспечение;
- заработная плата работников, занятых внедрением;
- страховые взносы.

Затраты на техническое и программное обеспечение определяются аналогично затратам на материалы в таблице 7.

Таблица 7 – Затраты на техническое и программное обеспечение

Наименование	Ед. изм.	Расход	Цена, руб.	Стоимость, руб.
Виртуальный хостинг Beget (3GB места, 2 сайта)	мес.	1	140	140
Аренда ноутбука (включает в себе установленную ОС Windows 10 и лицензию пакета Microsoft Office)	день	30	350	10500
Итого (общая стоимость)				10640
Затраты по доставке (10% от общей стоимости)				1064
Итого затрат с учетом доставки				11704

Заработную плату высчитывается на протяжении работы специалиста во время установки и подготовки оборудования можно определить в таблице 8.

Таблица 8– Заработная плата работников

Должность	Оклад, руб./мес.	Оплата в день, руб.	Трудоемкость, дни	Основная заработная плата, руб.
Инженер - программист	35000	1129	2	2258
Итого основная заработная плата				2258
Дополнительная заработная плата (12% от основной заработной платы)				271
Всего заработная плата (сумма основной и дополнительной заработной платы)				2529
Страховые взносы(10% от заработной платы)				253

Таблица 9 - Смета затрат на внедрение ИС Контроля оборудования в офисе

Наименование затрат	Сумма, руб.
Затраты на материалы	11704
Заработная плата	2529
Страховые взносы	253
Прочие расходы	500
Итого затрат	14986

3.4 Определение цены ИС для ИП "Гафуров Наиль Рахимьянович"

Цена разработки программного продукта с учетом затрат на внедрение проекта определяется по формуле 1:

$$Ц_{пп} = З_{общ} + П_{п} + НДС, \text{ руб.} \quad (1)$$

где $З_{общ}$ – сумма общих затрат по разработке и внедрению проекта.;

$П_{п}$ – планируемая прибыль руб.;

НДС – налог на добавленную стоимость (ставка НДС=20%), руб.

$$Ц_{ПП} = 65\,096 + 14\,972,08 + 14\,412,25 = 94\,480,33 \text{ руб.}$$

Планируемая прибыль может быть рассчитана исходя из нормативной рентабельности программного продукта, определяемая организацией и рассчитывается по формуле 2:

$$П_{п} = З_{общ} \times R_{п}, \text{ руб.} \quad (2)$$

где $R_{п}$ – коэффициент рентабельности, определенный предприятием.

$$П_{п} = 65\,096 \times 0,23 = 14\,972,08 \text{ руб.}$$

Налог на добавленную стоимость определяется по ставке установленной государством от суммы затрат на разработку программного продукта и величины плановой прибыли по формуле 3:

$$\text{НДС} = (\text{Зобщ} + \text{Пп}) \times \text{Кндс}, \text{ руб.} \quad (3)$$

где Зобщ – сумма общих затрат по разработке и внедрению проекта.;

Пп – планируемая прибыль руб.;

Кндс – коэффициент НДС(Кндс=0,2 при ставке НДС-20%).

$$\text{НДС} = (65\,096 + 14\,972,08) \times 0,2 = 14\,412,25 \text{ руб.}$$

В результате полученного положительного финансового результата (прибыли) предприятие должно начислить налог на прибыль(ставка налога-20% от прибыли) по формуле 4:

$$\text{НП} = \text{Пп} \times \frac{\text{ставка налога на прибыль}\%}{100\%}, \text{ руб.} \quad (4)$$

где Пп – планируемая прибыль руб.;

$$\text{НП} = 14\,972,08 \times 0,2 = 2\,994,42 \text{ руб.}$$

Прибыль оставшаяся на предприятии после уплаты налога на прибыль, определяется по формуле 5:

$$\text{Пч} = \text{Пп} - \text{НП}, \text{ руб.} \quad (5)$$

где Пп – планируемая прибыль руб.;

НП – налог на прибыль.

$$\text{Пч} = 14\,972,08 - 2\,994,42 = 11\,977,66 \text{ руб.}$$

В целом поступления в бюджет от реализации программного продукта составят 6:

$$\text{БП} = \text{НП} + \text{НДС}, \text{ руб.} \quad (6)$$

где НП – налог на прибыль.;

НДС – налог на добавленную стоимость (ставка НДС=20%), руб.

$$\text{Бп} = 2\,994,42 + 14\,412,25 = 17\,406,67 \text{руб.}$$

Вывод: внедрение продукта обойдется ИП в 65 096 руб. Для ИП "Гафуров Наиль Рахимьянович" это довольно не большая сумма учитывая, что система должна увеличить эффективность работы студии. А также при продаже продукта поступления прибыли будет составлять примерное 17 407 руб.

4.1 Описание предметной области

Рабочее место программиста - это его рабочий кабинет, либо часть помещения, в котором он проводит большую часть работы. Совершенно очевидно, что чем лучше приспособлено рабочее место для выполнения функций программиста, тем производительнее и эффективнее будет его трудовая деятельность.

Программист имеет собственное рабочее место, оборудованное ПЭВМ. Рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также – расстоянию спинки до переднего края сиденья.

Рабочая поверхность стола сделана из дерева. На столе размещены ЭВМ, монитор и принтер. Площадь поверхности стола предполагает размещение монитора, устройств ввода/вывода и рабочей зоны с местом расположения оперативной документации.

Кабинет, в котором расположено рабочее место, представляет собой помещение площадью 21м², длиной 7м, шириной 3 м и высотой 2.5 м.

Для создания благоприятных условий для зрительного восприятия кабинет оформлен в светло-бежевом цвете.

Для снижения шума используются ковровые покрытия на пол. Для создания и поддержания независимо от наружных условий оптимальных значений температуры, влажности, чистоты воздуха, в холодное время используется водяное отопление, в теплое время года применяется кондиционирование воздуха.

Режим работы программиста - односменный с пятидневной рабочей неделей и ненормированным рабочим днем. Продолжительность ежегодного отпуска - 28 календарных дней.

При идентификации вредных производственных факторов было установлено, что опасными могут являться:

- Шум и вибрация, источником которого является оргтехника;

- Электромагнитные поля и излучения от дисплея ПК;
- Статическое электричество, накапливаемое на клавиатуре, дисплее, корпусе системного блока;
- электрический ток в электрических сетях;
- запыленность рабочей зоны, так как рабочий стол располагается рядом с окном, и влажная уборка рабочей поверхности проводится не ежедневно, а только по мере загрязнения;
- статические перегрузки, так как работа программиста связана с неизменной статической позой, иногда неудобной позе (работа за ПК), что может привести к искривлению позвоночника, остеохондрозу, застою в органах малого таза;
- перенапряжение анализаторов, в большей степени зрительного, что может привести к различным нарушениям зрения (снижение остроты, спазм и уменьшение запаса аккомодации), помимо этого повышение зрительной нагрузки может приводить к общему утомлению, возникновению головной боли, ухудшению самочувствия.
- умственное перенапряжение, так как в процессе деятельности программист анализирует и перерабатывает огромное количество административных сетей, разрабатывает программные продукты, поддерживает технику в рабочем состоянии.

В целях выявления вредных и (или) опасных производственных факторов и осуществления мероприятий по приведению условий труда в соответствие государственными нормативными требованиями охраны труда ежегодно проводится аттестация рабочих мест по условиям труда - оценка условий труда на рабочих местах. Аттестации подлежат все имеющиеся в организации рабочие места.

При аттестации рабочих мест проводят оценку условий труда, гигиеническую оценку и оценку травмоопасности рабочих мест. При этом учитывают наличие средств коллективной защиты, обеспеченность

работников средствами индивидуальной защиты и определяют эффективность этих средств.

Согласно санитарно-гигиеническим нормам (СанПиН 2.2.4.723-98), дисплей должен располагаться на расстоянии 60-70 см, но не менее 50 см от глаз. Между боковыми поверхностями мониторов должно быть не менее 1,2 м. При использовании жидкокристаллических дисплеев на каждый компьютер должно приходиться не менее 4,5 м² площади. На дисплей ПК не должен попадать прямой солнечный свет во избежание бликов и повышенной нагрузки на зрение.

Поэтому на рабочем месте, где работает программист, стоит монитор на достаточно далеком расстоянии от начала стола, а комплектующая оргтехника расположена достаточно далеко непосредственно от самого специалиста. Это обусловлено тем, что все копировальные аппараты, принтеры и факс в периоды своей работы выделяют не только вредные токсичные газы, но и шумы и вибрацию. Данные факторы оказывают гораздо меньшее негативное влияние, нежели излучение от компьютера, но также являются вредными и могут нанести вред здоровью сотрудника.

Для снижения напряжения с глаз, а также для снятия мышечного напряжения во время рабочего дня проводится проветривание в течении 10 – 15 минут каждые 1,5 часа.

4.2 Расчет искусственного освещения в помещении в ИП «Гафуров Наиль Рахимьянович», где находится рабочее место программиста

Через зрительный анализатор человек получает около 80% из общего объема информации. Качество поступающей информации во многом зависит от освещения: неудовлетворительное количественно или качественно, оно не только утомляет зрение, но и вызывает утомление организма в целом. Кроме того, нерациональное освещение может явиться причиной травматизма:

плохо освещенные опасные зоны, слепящие источники света и блики от них, резкие тени ухудшают видимость настолько, что вызывают полную потерю ориентировки работающих, снижают производительность труда и увеличивают брак продукции. Поэтому необходимо достаточное искусственное освещение для рабочего места.

Условия работы для расчета показателя освещенности рабочего места программиста являются:

- Помещение размером 21 квадратный метр;
- Высота помещения 2.5 метра, длина 7 метров, ширина 3 метра;
- Лампы люминесцентные (дневного света) в количестве 5 штук, новые;
- Крупногабаритная мебель отсутствует.

Расчет общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента использования светового потока. Световой поток (лм) одной лампы рассчитывается по следующей формуле:

$$\Phi_{л} = \frac{E_n \cdot S \cdot z \cdot K}{N_c \cdot \gamma \cdot \eta},$$

где E_n - нормируемая минимальная допустимая освещенность по СНиП 23-05-95, для помещения 2000 лк;

S - площадь освещаемого помещения, $S=21 \text{ м}^2$;

z - коэффициент неравномерности освещения, $z = 1,1$;

K - коэффициент запаса, учитывающий снижение освещенности в процессе эксплуатации вследствие загрязнения и старения ламп и светильников, а также снижения отражающих свойств поверхностей помещения для кабинета будет равен 1,3;

N_c - число светильников в помещении, $N_c=5$;

γ - коэффициент затенения, обычно 1;

$\eta_{=i}$ - коэффициент использования светового потока.

Коэффициент использования светового потока, давший название методу расчета, определяется по индексу помещения i в зависимости от типа светильника и коэффициентов отражения света от потолка, стен и пола:

$$i = \frac{A \cdot B}{Hc(A+B)},$$

где А, В - длина и ширина помещения в плане, м;

Нс - высота подвеса светильников над рабочей поверхностью, м.

$$i = \frac{21}{2.6(7 + 3)}$$

$$i = 0.8$$

$$\Phi_{л} = \frac{2000 \cdot 21 \cdot 1,1 \cdot 1,3}{5 \cdot 1 \cdot 0.8}$$

$$\Phi_{л} = 15015 \text{ лм}$$

Так как в СНиП величина освещенности дается в лк, переведем рассчитанную величину из лм в лк. Для этого нужно разделить получившийся в расчетах результат на количество квадратных метров указанного помещения:

$$\Phi_{л} = 715 \text{ лк}$$

Проверяем полученную величину освещенности $\Phi_{л}$ согласно СНиП 52.13330.2016 по таблице 4.2 «Требования к освещению помещений жилых и общественных зданий», так как офис ИП «Гафуров Наиль Рахимянович» относится к общественным зданиям, и таблице Л1, Приложение Л «Нормативные показатели освещения основных помещений общественных, жилых и вспомогательных зданий». Величина светового потока является достаточной для рассматриваемого помещения.

Также по таблице К1, Приложение К выбираем рекомендуемые источники света. Для обеспечения зрительного комфорта в помещениях при выполнении зрительных работ разрядов А-В рекомендуется освещенность от 300-500 лк, индекс цветопередачи и источников света R_a - 90-95, диапазон цветовой температуры источников света $T_{ц}$, К - 5000-6500. Рекомендуемые источники света для общего освещения: СД - светодиодные лампы; ЛЛ - люминесцентные лампы типов ЛДЦ - лампы дневного света с улучшенной цветопередачей с индексом цветопередачи $R_a = 90$ и цветовой температурой $T_{ц} = 5000-6500$ К.

4.3 Электробезопасность в ИП «Гафуров Наиль Рахимьянович»

При поступлении на работу сотрудник должен пройти инструктаж по технике безопасности и электробезопасности. Работника знакомят с основными правилами по технике безопасности, предлагают внимательно прочитать действующие на предприятии инструкции, поясняя при этом отдельные правила и требования.

Инструктаж по технике безопасности при выполнении конкретной работы проводит руководитель соответствующего производственного участка, показывая инструктируемому правильные безопасные приемы работы.

Повторный инструктаж проводится для рабочих независимо от их квалификации, стажа и опыта работы, не реже 1 раза в 6 месяцев по программе инструктажа на рабочем месте.

При нарушении работающими правил и инструкций по технике безопасности, технологической и производственной дисциплины, а также в случаях изменения технологического процесса или вида работы проводят дополнительные инструктажи.

Все инструктажи оформляются записями в специальном журнале с указанием номеров или шифров инструкций. Журнал о проведении инструктажа хранится у руководителя подразделения.

Работник на рабочем месте должен соблюдать общие меры электробезопасности. При этом запрещается:

- прикасаться к задней панели системного блока при включенном питании;
- допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств;
- производить самостоятельно вскрытие и ремонт оборудования;

- пользоваться неисправными электроприборами и электропроводкой;
- ремонтировать электроприборы самостоятельно;
- подвешивать электропровода на гвоздях, металлических и деревянных предметах, перекручивать провод, закладывать провод и шнуры на водопроводные трубы и батареи отопления, вешать что-либо на провода, вытягивать за шнур вилку из розетки;
- прикасаться одновременно к персональному компьютеру и к устройствам, имеющим соединение с землей (радиаторы отопления, водопроводные краны, трубы и т.п.);
- пользоваться самодельными электронагревательными приборами и электроприборами с открытой спиралью;
- наступать на переносимые электрические провода, лежащие на полу.

При работе с электроприборами и оргтехникой (персональные компьютеры, принтеры, сканеры, копировальные аппараты, факсы, бытовые электроприборы, приборы освещения):

- Автоматические выключатели и электрические предохранители должны быть всегда исправны.
- Изоляция электропроводки, электроприборов, выключателей, штепсельных розеток, ламповых патронов и светильников, а также шнуров, с помощью которых включаются в электросеть электроприборы, должны быть в исправном состоянии.
- Электроприборы необходимо хранить в сухом месте, избегать резких колебаний температуры, вибрации, сотрясений.
- Для подогрева воды пользоваться сертифицированными электроприборами с закрытой спиралью и устройством автоматического отключения, с применением несгораемых подставок.

4.4 Пожарная безопасность в ИП «Гафуров Наиль Рахимьянович»

Инструктаж по пожарной безопасности проводится по программе, разработанной инженером по охране труда ФГБОУ ВО СПбГПМУ Минздрава России, с учетом требований стандартов, правил, норм и инструкций о мерах пожарной безопасности. Продолжительность инструктажа устанавливается в соответствии с утвержденной программой. Инструктаж по пожарной безопасности, как правило, проводится совместно с инструктажем по технике безопасности и в те же сроки.

Инструктаж по пожарной безопасности проходят все работники организации, независимо от их образования, стажа работы по данной профессии или должности, временные работники, командированные, обучающиеся и студенты, прибывшие на практику.

О проведении инструктажа по пожарной безопасности работник, проводивший инструктаж, делает запись в журнале учета проведения инструктажей по пожарной безопасности, с обязательной подписью инструктируемого и инструктирующего. При регистрации внепланового инструктажа по пожарной безопасности указывают причину его проведения.

Все работники организаций должны допускаться к работе только после прохождения противопожарного инструктажа, а при изменении специфики работы проходить дополнительное обучение предупреждению и тушению возможных пожаров в порядке, установленном руководителем.

На проведение первичного противопожарного инструктажа необходимо отводить не менее 1 ч. Инструктируемые должны ознакомиться:

- С действующими на объекте правилами пожарной безопасности и инструкциями;
- С производственными участками, наиболее опасными в пожарном отношении, где запрещается курить, применять открытый огонь;
- С возможными причинами возникновения пожара и мерами его предупреждения;

– С практическими действиями в случае возникновения пожара - вызов пожарной помощи, использование первичных средств пожаротушения, место расположения ближайшего телефона и ознакомление с правилами поведения в случае возникновения пожара, эвакуации людей и материальных ценностей.

При первичном инструктаже инструктирующий обязан рассказать о производственных установках с повышенной пожарной опасностью, мерах предотвращения пожаров и загораний, указать место курения, ознакомить вновь поступившего с имеющимися на объекте средствами пожаротушения, показать ближайший телефон и объяснить правила поведения в случае возникновения пожара.

Проведение противопожарного инструктажа в обязательном порядке должно сопровождаться практическим показом способов использования имеющихся на объекте средств пожаротушения (противогазы, респираторы, огнетушители и т.д.).

Соблюдение рассмотренных в данном разделе мероприятий по охране труда и технике безопасности в ИП «Гафуров Наиль Рахимьянович» позволяет снизить утомляемость и травматизм, повысить производительность труда, обеспечить комфортные условия трудовой деятельности специалиста, работающего в должности программиста.

ЗАКЛЮЧЕНИЕ

Во время прохождения всего дипломного проекта была разработана и протестирована система, которая автоматизирует создания архитектуры проекта. В этой системе зарегистрированный пользователь может создать проект, изменить информацию проекта, а также создать для проекта архитектуру через специально разработанный интерфейс.

В начале дипломного проекта была поставленная следующая цель - разработка информационной системы, которая автоматизирует этап проектирование разработки сервисов и программных продуктов. Поставленная цель была частично достигнута. Созданная система автоматизирует один из пару этапов проектирования, но не все.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Гамма, Э. Приемы объектно-ориентированного проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – М. : Питер, 2013. - 368 с.
- 2 Каллум, Х. PHP. Быстрый старт / Х. Каллум. – М. : Litres, 2017. – 300 с.
- 3 Алексей, В. JavaScript в примерах и задачах / В. Алексей. – М. : Litres, 2017. – 715 с.
- 4 Дунаев, В. В. HTML, скрипты и стили. 4–е изд. / В. В. Дунаев. – БХВ. : Петербург, 2015. – 816 с.
- 5 Лазрис, Л. CSS. Быстрый старт / Л. Лазрис. – М. : Litres, 2018. – 2018. – 191 с.
- 6 Машин, Т. Bootstrap: Быстрое создание современных сайтов / Т. Машин. – М. : Litres, 2017. – 123 с.
- 7 Беэр, Б. jQuery в действии. 3–е издание / Б. Беэр, К. Иегуда, Р. Аурелио. – М. : Питер, 2017. – 528 с.
- 8 PHP: Что такое PHP? – Manual [Электронный ресурс]. – М. : The PHP Group, 2001–2019. – Режим доступа: [www.url: https://www.php.net/manual/ru/intro-what-is.php](http://www.url:https://www.php.net/manual/ru/intro-what-is.php) – 05.06.2019
- 9 PHP: фрактал плохого дизайна / Хабр [Электронный ресурс]. – М. : ТМ, 2006–2019. – Режим доступа: [www.url: https://habr.com/ru/post/142140/](http://www.url:https://habr.com/ru/post/142140/) – 05.06.2019
- 10 Сухов, К. Node.js. Путеводитель по технологии / К. Сухов. – М. : Litres, 2017. – 415 с.
- 11 Нимейер, П. Программирование на Java / П. Нимейер, Д. Леук. – М. : Litres, 2017. – 1211 с.
- 12 Java теперь платная? Развенчиваем слухи (или нет?) / Хабр [Электронный ресурс]. – М. : ТМ, 2006–2019. – Режим доступа: [www.url: https://habr.com/ru/post/448632/](http://www.url:https://habr.com/ru/post/448632/) – 05.06.2019

- 13 Fat-Free Framework for PHP [Электронный ресурс]. – М. : F3 Community – Режим доступа: [www.url: https://fatfreeframework.com/3.6/development](http://www.fatfreeframework.com/3.6/development) – 05.06.2019
- 14 MySQL. / What is MySQL? [Электронный ресурс]. – М. : Oracle Corporation, 2019. – Режим доступа: www.url: https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html – 05.06.2019
- 15 Setting up Visual Studio Code [Электронный ресурс]. – М. : Microsoft, 2019. – Режим доступа: www.url: https://code.visualstudio.com/docs/setup/setup-overview – 05.06.2019
- 16 Чактон, С. Git для профессионального программиста / С. Чактон, Б. Штрауб. – М. : Питер, 2015. – 496 с.
- 17 Работа в Санк – Петербурге, поиск персонала и публикация вакансий / hh.ru [Электронный ресурс]. – М. : Группа компаний HeadHunter, 2019. – Режим доступа: www.url: https://spb.hh.ru/ – 05.06.2019
- 18 SALATEX [Электронный ресурс]. – М. : SALATEX, 2011–2019. – Режим доступа: www.url: http://www.saletex.ru/ – 05.06.2019
- 19 Uninsy [Электронный ресурс]. – М. : Uninsy, 2019. – Режим доступа: www.url: https://uninsy.com/ – 05.06.2019
- 20 Gafurov Studio [Электронный ресурс]. – М. : Gafurov Studio, 2017 – 2019. – Режим доступа: www.url: https://www.gafurov.ru/ – 05.04.2019
- 21 Beget – платный хостинг. [Электронный ресурс]. – М. : Бегет, 2007 – 2019. – Режим доступа: www.url: https://beget.com/ru – 05.04.2019