Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Югорский государственный университет

Институт (НОЦ) систем управления и информационных технологий

Кафедра автоматизированных систем обработки информации и управления (АСОИиУ)

Дипломный проект

на тему: Интернет-сервис «Налоговый помощник»

Студент-дипломник группы 1170

Овчинников М. А. \_\_\_\_\_\_\_\_

(подпись)

Руководитель от ЮГУ:

к.т.н. Макеева М. А. \_\_\_\_\_\_\_\_

(подпись)

Консультант по безопасности жизнедеятельности:

к.т.н. Макеева М.А. \_\_\_\_\_\_\_\_

(подпись)

Консультант по экономике и организации производства:

к.т.н. Макеева М.А. \_\_\_\_\_\_\_\_

(подпись)

Нормоконтролер:

к.т.н. МакееваМ.А. \_\_\_\_\_\_\_\_

(подпись)

Рецензент:

к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_. \_\_\_\_\_\_\_\_

(подпись)

Ханты-Мансийск 2012

**Аннотация**

Тема выпускной квалификационной работы: интернет-сервис «Налоговый помощник» .

Лит

Лист

Листов

2

37

У

И(НОЦ)СУиИТ-ЮГУ

Интернет-сервис «Налоговый помощник»

17017

Измт

№ Докум

Лист

Подп

Дата

Овчинников М.А.

Разраб

Макеева М.А.

Пров

Макеева М.А.

Н. контр.

Моновская А.В.

Утв.

Целю дипломной работы является разработка интернет-сервиса налоговый помощник.

Объектом автоматизации является процесс получения информации по применению налоговых ставок и льгот по имущественным налогам и их администрирования.

Пояснительная записка к выпускной квалификационной работе состоит из 11 глав.

В первой главе представлено описание предметной области, обзор аналогов, описание существующих задач. Во второй главе проводится описание выбора программных средств, процесс проектирования, итоги реализации. В третьей главе описываются способы и варианты тестирования, а также типичные ошибки и варианты их устранения. В четвёртой главе проводится расчет экономической эффективности, результаты которого оправдывают внедрение системы. В пятой главе проводится анализ опасных и вредных факторов, учитываемых для данной системы. В заключении подводятся итоги и описываются возможные направления развития системы.

Пояснительная записка изложена на 85 страницах, включает 27 рисунков, 3 таблицы и 9 приложений. Список использованных источников содержит 19 наименований.

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Югорский государственный университет

Институт (НОЦ) систем управления и информационных технологий

Кафедра «Автоматизированные системы обработки информации и управления»

**"Утверждаю" З А Д А Н И Е**

**Зав. кафедрой на дипломный проект (работу)**

\_\_\_\_\_\_\_\_\_к.т.н., А.В. Моновская

"\_\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2012 г.

Студенту Овчинникову Максиму Андреевичу группы 1170

1. Тема проекта (работы) Интернет-сервис «Налоговый помощник»

(Утверждена приказом по университету от 20\_\_\_г. № )

1. Срок сдачи студентом проекта (работы) − 3 июня 2012 г.
2. Техническое задание разработано в соответствии с требованиями к интернет-сервису «Налоговый помощник»

IV. Оглавление пояснительной записки к дипломной работе

Введение

* + - 1. Постановка задачи
      2. Формирование требований к интернет-сервису «Налоговый помощник»
      3. Техническое задание
      4. Разработка концепции АСОИУ
      5. Детальное проектирование
      6. Реализация
      7. Апробация
      8. Организационно-экономическая часть
      9. Безопасность жизнедеятельности

Заключение

V. Перечень иллюстративного материала

1.Приложение А – Структура УФНС по ХМАО − Югре

2.Приложение Б – Структурно-функциональная модель системы «AS-IS»

3.Приложение В – Диаграмма потоков данных (DFD)

4.Приложение Г – Техническое задание

5.Приложение Д – Диграммы последовательностей детального проектирования

VI. Дата выдачи задания "\_\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_ 2012г.

Руководитель дипломного проектирования

(подпись, ФИО)

Задание принял к исполнению

(подпись, ФИО)

"\_\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2012 г.

**Содержание**

Введение

1. Постановка задачи
   1. Сбор и анализ данных об объекте автоматизации
   2. Формулирование цели и задач создания интернет-сервиса «Налоговый помощник»
   3. Анализ и выбор инструментов
2. Формирование требований к интернет-сервису «Налоговый помощник»
   1. C-требования
      1. Описание системы
      2. Описание требований с применением методологии SADT
      3. Описание требований с применением методологии UML
   2. D-требования
      1. Функциональные требования
      2. Нефункциональные требования
      3. Обратные требования
      4. Описание диаграммы последовательности UML
3. Техническое задание
4. Разработка концепции АСОИУ
   1. Описание архитектуры приложения
   2. Выбор шаблонов построения архитектурных слоев
   3. Проектирование архитектурных слоев
5. Эскизный проект
   1. Введение глобальных библиотек
   2. Описание диаграммы последовательности UML
   3. Описание диаграммы классов UML
      1. Описание слоя данных
      2. Описание слоя представлений
      3. Описание сервисного слоя

5.3.3 Описание слоя домена

1. Реализация
   1. Реализация слоя домена
   2. Реализация представления
   3. Реализация сервисного слоя
2. Апробация
   1. Тестирование
   2. Модульное тестирование
   3. Нагрузочное тестирование
3. Организационно-экономическая часть
4. Безопасность жизнедеятельности
   1. Характеристика условий труда
   2. Требования к производственным помещениям
   3. Эргономические требования к рабочему месту
   4. Режим труда
   5. Расчет освещенности
   6. Расчет уровня шума
   7. Пожарная безопасность

Заключение

Список использованных источников

## Введение

В рамках выпускной квалификационной работы рассматривается деятельность по предоставлению налогоплательщику информации о налоговых ставках.

Для получения сведений о налоговых ставках и льготах налогоплательщику необходимо позвонить в налоговую службу или явиться в налоговый орган. Этот процесс отнимает время налогоплательщика.

После начисления налога налогоплательщик получает уведомление о необходимости его уплаты с указанной суммой. Для проверки достоверности начисленного налога налогоплательщик может рассчитать его сам. Для вычисления налога ему необходима информация по налоговым ставкам, действующим в его муниципальном образовании. Также большинство налогоплательщиков, относящихся к льготной категории, не знают об этом.

Решением данной проблемы является разработанный интернет-сервис «Налоговый помощник», которая в оперативном порядке предоставляет налогоплательщику информацию о налоговых ставках и льготах, действующих на территории Ханты-Мансийского автономного округа – Югры.

Интернет-сервис «Налоговый помощник» направлен на физических лиц, особенно социально незащищенных групп, в том числе пенсионеров и людей с ограниченными возможностями. Интернет-сервис решает следующие задачи:

- предоставление быстрого доступа к информации о налоговых ставках и льготах;

- повышение качества обслуживания налогоплательщиков;

- обеспечение удобства получения информации;

- обеспечение полнота предоставляемой информации.

Актуальность данной работы является стратегии развития Федеральной налоговой службы Российской Федерации (ФНС РФ) как клиент-ориентированной организации. В рамках данной стратегии решаются следующие задачи:

- обеспечение прозрачности деятельности ФНС России и обеспечение доступности государственных услуг;

- повышение осведомленности налогоплательщиков о деятельности налоговой службы и предлагаемых сервисах;

- качественное расширение дистанционного взаимодействия для широких слоев налогоплательщиков при одновременном повышении качества очного взаимодействия,

- экстерриториальность обслуживания налогоплательщиков (Экстерриториальность – это свобода налогоплательщика в получении доступа к информации, сервисам и обслуживанию вне зависимости от места регистрации и нахождения, что для нашей огромной страны особенно актуально).

Одним из типовых решений создания поставленной задачи является разработка веб-приложения для работы с базой данных.

Вопросам проектирования и реализации корпоративных информационных систем в настоящее время посвящено значительное количество работ и, более того, активно внедряются и используются различного рода платформы (ERP-системы, инструментальные среды разработки (IDE), системы управления содержимым (CMS, CMF) и т.д.), которые позволяют в рамках единой программно-алгоритмической среды реализовать специфические для заданной предметной области требования.

Каркасные системы (фреймворки) в настоящее время могут рассматриваться как очередной виток эволюции инструментов создания информационных систем с web-интерфейсом, поскольку они предоставляют практически неограниченный набор возможностей, позволяющих облегчить процесс разработки информационной системы (ИС), повысить надежность и безопасность, а также расширить функционал ИС с минимальными ресурсными затратами. Современные каркасные системы управления контентом позволяют не только генерировать HTML-страницы, но и предоставляют интерфейс доступа к базе данных. В этой связи использование каркасных систем становится все более востребованным, так как в корпоративных приложениях наиболее распространенными функциями являются функции, отвечающие за взаимодействие с БД.

## Постановка задачи

* 1. **Сбор и анализ данных об объекте автоматизации**

Объектом автоматизации в данной дипломной работе является система предоставления информации о налоговых ставках по транспортному, земельному и имущественному налогах и льготах.

УФНС России по ХМАО имеет в своем составе 14 инспекций. Структура предприятия прилагается в приложении А (Рисунок А.1).

Законом «Об основах налоговой системы в Российской Федерации» введена трехуровневая система налогообложения предприятий, организаций и физических лиц.

Транспортный налог и установление льгот по нему относится к первому уровню. Первый уровень – это федеральные налоги России. Они действуют на территории всей страны и регулируются общероссийским законодательством.

Второй уровень − налоги республик в составе Российской Федерации и налоги краев, областей, автономной области, автономных округов.

Земельный и имущественный налоги включены в третий уровень налогов. Ставки и льготы по данным видам налогов определяются местными органами (т. е. налоги районов, городов, поселков и т. д.). Земельный налог классифицируется в зависимости от типа налогооблагаемой земли. Имущественные налоги классифицируются по суммарной инвентаризации стоимости объекта налогообложения. Налоговые льготы классифицируются по объему предоставляемых льгот. Классификатор граждан определяет типы физических лиц, которым налоговые льготы предоставляются, а также вид предоставляемой льготы.

Процедура по информированию налогоплательщиков определяется административным регламентом №9 федеральной налоговой службы по исполнению государственной функции по бесплатному информированию (в том числе в письменной форме) налогоплательщиков [1].

При обследовании объекта автоматизации была построена модель «черного ящика» системы, представляющая собой перечень входов и выходов системы, и, предназначенная, с одной стороны, для указания границ проектируемой системы, с другой стороны, для фиксирования значимых с точки зрения заказчика входных и выходных связей. Рассмотрение объекта автоматизации как «черного ящика» системы «AS-IS», приведено в структурно-функциональной диаграмме с применением методологии IDEF0 приложение Б рисунок Б.1.

На вход системы поступает запрос налогоплательщика на получение информации о налоговых ставках и льготах. Управляющими элементами в системе являются административный регламент №9 «О бесплатном информировании налогоплательщика», который определяет порядок исполнения государственной функции по предоставлению информации, федеральные законы РФ о налогах, решения субъектов РФ и муниципальных образований субъектов, которые устанавливают налоговые ставки и льготы. Механизмами системы являются сотрудники налоговой службы и налогоплательщики. Выходами системы являются письмо с ответом на письменное обращение и обслуженный налогоплательщик.

Декомпозиция подсистем определит функциональные процессы системы и логическое отношение между ними. Проводя декомпозицию модели «черного ящика» системы «AS-IS», получаем модель, приведенную в приложении Б на рисунке Б.2. Из декомпозиции видно, что налогоплательщик может получить информацию по налоговым ставкам и льготам тремя способами: обращение в налоговую службу лично, звонок в справочную службу налоговой инспекции, обращение в письменной форме.

При информировании налогоплательщиков на основании обращений в письменной форме поступившее обращение проверяется на соответствие требований заполнения. Затем оно регистрируется в информационном ресурсе системы электронного документооборота налогового органа "Журнал учета письменных запросов налогоплательщиков по информированию и организации работы с налогоплательщиками". Далее подготавливается ответ согласно сути обращения, после подготовленный документ визируется начальником отдела и передается на подпись начальнику инспекции. Далее исходящий документ регистрируется в журнале учета письменных запросов налогоплательщиков по информированию и организации работы с налогоплательщиками и направляется почтовым отправлением налогоплательщику. Декомпозиция процесса информирования на основании письменных обращений представлена в приложении Б на рисунке Б.3.

Декомпозиция получения информации о налоговых ставках и льготах по телефону справочной службы налогового органа представлена в приложении Б на рисунке Б.4. При обращении по телефону лицу, исполняющему государственную функцию, необходимо представиться, назвав налоговый орган, куда поступило обращение, свою фамилию, имя, отчество и должность. Затем необходимо представиться налогоплательщику и изложить суть обращения. Далее для исполнения информирования о налоговых ставках налоговому сотруднику необходимо запросить информацию у отдела налогообложения и затем предоставить информацию налогоплательщику. Телефонные обращения регистрируются в журнале учета запросов налогоплательщика по телефону.

В приложении Б на рисунке Б.5 представлена декомпозиция получения информации на основании устного обращения в налоговый орган лично. Для получения информации лично налогоплательщику необходимо обратиться в налоговый орган согласно установленному графику работы и возможно ожидать очереди. После налогоплательщику необходимо представиться и высказать суть обращения. Чтобы предоставить информацию о налоговых ставках и льготах, налоговый сотрудник запрашивает информацию в отделе налогообложения.

**1.2. Формулирование целей и задач создания** **интернет-сервиса «Налоговый помощник»**

Целью выпускной квалифицированной работы является разработка интернет-сервиса «Налоговый помощник». Сервис направлен на минимизацию времени получения информации, повышению качества обслуживания налогоплательщиков, обеспечение доступности и полноты информации о налоговых ставках и льготах и создание единой информационной базы о налоговых ставках и льготах для совершенствования администрирования налоговых ставок и льгот.

Интернет-сервис налоговый помощник в режиме интерактивного взаимодействия предоставляет возможность получить налогоплательщику информацию о налоговый ставках и льготах. В режиме администрирования налоговый сотрудник может изменять информацию о налогах, налоговых ставках и льготах.

Для достижения поставленной цели необходимо решить следующие задачи.

1. Разработать концепцию автоматизированной системы.

2. Разработать программный интерфейс (ПС) для взаимодействия налогоплательщика с системой.

3. Выбрать инструменты и средства реализации интернет-сервиса.

4. Разработка структуры базы данных для хранения информации о налогах, пользователях и статистики.

**1.3. Анализ и выбор инструментов**

Ограничением, которое наложено заказчиком системы на разработку интернет-сервиса, является использование интерпретируемого языка программирование PHP.

PHP позволяет формировать страницы в режиме интерактивного взаимодействия в системах «клиент-сервер» и используется в качестве технологии построения веб-приложений. Веб-приложение − клиент-серверное приложение, в котором клиентом выступает браузер, а сервером − веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными сервисами.

Эффективность использования веб-интерфейса для решения задач автоматизации организаций очевидна. Переход к его использованию в программных продуктах для автоматизации деятельности организаций привел к возникновению новых видов ресурсов, позволяющих значительно повысить доступность для пользователей как информации, так и вычислительных ресурсов. В то же время разработка интернет-приложений остаётся в настоящий момент достаточно сложным и трудоёмким процессом, что не позволяет в полной мере использовать все их преимущества начинающему разработчику.

Однако переход от традиционных программных продуктов к технологии «тонких» клиентов становится все более очевидным. Примером этого может служить возникновение широкодоступных веб-приложений в таких областях, как текстовые редакторы и электронные таблицы, которые ранее считались областью применения традиционных программных продуктов.

В сфере корпоративного и научного программного обеспечения из-за высокой конкуренции существуют жёсткие требования как к срокам разработки, так и к её качеству и стабильности. В случае разработки интернет-приложения для доступа к существующим информационным ресурсам необходимо обеспечить минимально возможное количество ошибок с целью повышения общей отказоустойчивости системы. Это важно из-за того, что полученные от новой среды доступа преимущества могут быть нивелированы синергетическим ростом нестабильности за счёт взаимного влияния уже существовавших и новых ошибок. В свою очередь, это требует создания новых инструментальных средств и методических подходов, позволяющих систематизировать процесс создания программного продукта [2].

В настоящее время существует огромное количество инструментов, значительно упрощающих разработку приложений, одним из которых являются так называемые каркасные среды разработки.

Каркасная система (или программный фреймворк) − программная среда, позволяющая строить сложные программные приложения используя представленный единый программный интерфейс или API (application programm interface), облегчающая разработку и объединение разных компонентов большого программного проекта, включающая вспомогательные программы, библиотеки кода, язык сценариев и другое программное обеспечение. Отличие программного фреймворка от библиотеки: в программном фреймворке выполняется код, написанный для него, а не исполняется один из скриптов подключенной библиотеки. Также, в отличие от библиотеки, которая объединяет в себе набор близкой функциональности, фреймворк содержит в себе большое число разных по тематике библиотек. При этом использование единого API позволяет расширять фреймворк, не затрагивая ядро системы, и использовать собственные расширения в будущих проектах [3]. В ходе поиска подходящего решения были найдены большое количество каркасных систем. На данный момент самыми распространенными являются CodeIgniter 2.1.0, Yii, Zend Framework и CakePHP.

CodeIgniter (Версия 2.1.0) – Model-View-Controller (MVC) фреймворк c открытым исходным кодом, написанный на языке программирования PHP [4]. Разработан компанией EllisLab.

Особенности системы:

* качественная и полная документация с примерами, большое сообщество;
* быстрая производительность;
* малый размер дистрибутива;
* гибкость в разработке приложений.

Возможности CodeIgniter (Версия 2.1.0):

* поддержка баз данных MySQL, PostgreSQL, Microsoft SQL, SQLite, Oracle;
* поддержка псевдо-ActiveRecord;
* расширяемость за счет использования сторонних библиотек и переопределения существующих;
* поддержка Friendly URL – веб-адресов, удобных для восприятия человеком;
* широкий набор встроенных библиотек;
* возможность миграции баз данных;
* поддержка модульности (HMVC) с помощью дополнений.

Недостатки фреймворка:

* слабая поддержка работы с сессиями;
* по умолчанию модели не поддерживаю ORM;
* отсутствует возможность расширения класса работы с базами данных.

Следующая рассматриваемая система Zend Framework − MVC фреймворк для создания веб приложений на платформе PHP. Разрабатывается компанией Zend Technologies, имеет открытый исходный код (лицензия BSD, позволяет использовать код в закрытых коммерческих системах, без необходимости публикации кода) [5]. К плюсам данного фреймворка можно отнести:

* кроссплатформенность,
* большое количество классов и компонентов для решения стандартных задач веб разработки и работы со сторонними сервисами,
* использование постоянной структуры проектов (или соглашение по именованию),
* документированность,
* поддержка современных СУБД,
* поддержка Friendly URL – веб-адресов, удобных для восприятия человеком.

К недостаткам можно отнести отсутствие поддержки ORM, для работы с БД.

Yii (Версия 1.1) – это фреймворк для веб-программирования общего назначения, который может быть использован для разработки практически любых веб-приложений. Благодаря небольшому размеру и наличию технологичных средств кэширования, Yii особенно подходит для разработки приложений с большим потоком трафика, таких как порталы, форумы, системы управления контентом, системы электронной коммерции и др [6].

Ниже приведен список возможностей [7]:

* Database Access Objects (DAO) и Active Record: Yii позволяет разработчикам моделировать данные БД в качестве объектов, в тоже время оставляя возможность написания длинных и повторяющихся SQL выражений,
* ввод форм и проверка: Yii обрабатывает данные форм легко и безопасно. Yii Framework поставляется с набором валидаторов, которые могут использоваться для проверки входящих данных, а также имеет helper методы и виджеты для отображения ошибок проверки,
* аутентификация и авторизация: Yii имеет встроенную поддержку аутентификации. Он также поддерживает авторизацию через иерархическое распределение прав (RBAC),
* безопасность: Yii оборудован многими мерами безопасности для защиты веб-приложений от атак. Эти меры включают в себя предотвращение кросс-сайтового скриптинга (XSS), предотвращение кросс-сайтовой подделки запросов (CSRF), предотвращение фальсификации cookies и др,
* автоматическая генерация кода: Yii предоставляет утилиты, которые могут автоматически сгенерировать нужный для вас код, такой как скелет приложения, CRUD приложения и др.,
* полностью объектно-ориентированный,
* совместимость с кодом сторонних приложений,
* подробная документация: каждый отдельный метод или свойство подробно документировано. Приводятся полноценные уроки вместе с небольшими советами.

CakePHP − это [программный каркас](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%BA%D0%B0%D1%80%D0%BA%D0%B0%D1%81) для создания [веб-приложений](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), написанный на языке [PHP](http://ru.wikipedia.org/wiki/PHP) и построенный на принципах [открытого ПО](http://ru.wikipedia.org/wiki/Open_Source). CakePHP реализует структурный шаблон проектирования MVC [8].

Изначально создавался как клон популярного [Ruby on Rails](http://ru.wikipedia.org/wiki/Ruby_on_Rails), и многие идеи были заимствованы именно оттуда:

1. Своя файловая структура.
2. Поддержка множества плагинов.
3. Абстракция данных ([PEAR](http://ru.wikipedia.org/wiki/PEAR)::DB, ADOdb и собственная разработка Cake)
4. Поддержка множества [СУБД](http://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) ([PostgreSQL](http://ru.wikipedia.org/wiki/PostgreSQL" \o "PostgreSQL), [MySQL](http://ru.wikipedia.org/wiki/MySQL), [SQLite](http://ru.wikipedia.org/wiki/SQLite), [Oracle Database](http://ru.wikipedia.org/wiki/Oracle_Database)), может быть подключен к другим СУБД, например [Firebird](http://ru.wikipedia.org/wiki/Firebird).

Описание возможностей CakePHP:

* диспетчер URL с применением регулярных выражений,
* генерация всего кода по схеме базы данных (требуется соблюдать стандарт именования столбцов),
* компоненты для авторизации, ограничения доступа (ACL), управления сессиями, cookies, представления древовидной информации (в виде Nested Sets),
* генерация SQL-запросов, в том числе для таблиц с отношениями один ко многим и многие ко многим, ORM,
* скаффолдинг и генерация CRUD-страниц для сущностей, Router::mapResources c Put Delete Get Post,
* автогенератор кода Bake,
* поддержка Simple Test,
* низкий порог вхождения — быстрая разработка, основана на соглашениях.

Проанализировав фреймворки, были выбраны значимые при проектировании сервиса показатели, оценка которых позволила выбрать инструментальную среду, подходящую для разработки интернет-сервиса «Налоговый помощник». Были выявлены следующие значимые показатели:

* сложность установки и настройки системы,
* возможности базовой платформы,
* поддержка БД и ORM,
* документация и сообщество,
* расширяемость и масштабируемость,
* производительность,
* управление пользователями.

Все показатели сводятся в таблицу (табл. 1.1), в которой на основании экспертных оценок каждому показателю назначается весовой коэффициент Vj (от 1 до 10), характеризующий значимость данного показателя по сравнению с остальными. Сумма значений всех весовых коэффициентов должна равняться его верхней границе весового коэффициента.

С использованием данных, полученных из литературных источников и/или от экспертов, по каждому j-ому показателю для каждой i-ой системы определяется степень полезности Ui,j.

Для каждой i-ой сравниваемой системы вычисляется значение оценочной функции по формуле: Fi = V1 x Ui,1 + V2 x Ui,2 + …=Σ Vj x Ui,j.

На основании значения оценочной функции делается вывод о целесообразности использования той или иной системы в данном проекте при учете выбранных критериев и заданных ограничений.

Таблица 1.1 – Таблица оценок показателей сравниваемых фреймворков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Показатель | Вес  (1-15) | Zend Framework  (1-10) | CakePHP  (1-10) | Yii  (1-10) | CodeIgniter  (1-10) |
| 1 | Легкость  инсталляции и настройки системы | 12 | 6 | 10 | 8 | 10 |
| 2 | Возможности базовой платформы | 10 | 5 | 4 | 8 | 8 |
| 3 | Поддержка БД и ORM | 8 | 8 | 8 | 10 | 9 |
| 4 | Документация и сообщество | 13 | 9 | 8 | 7 | 10 |
| 5 | Гибкость и масштабируемость | 12 | 10 | 7 | 8 | 10 |
| 6 | Производительность | 8 | 8 | 7 | 9 | 9 |
| Итог |  |  | 487 | 478 | 530 | 578 |

На основании значений оценочной функции и результата анализа можно сделать вывод о том, что наиболее целесообразно использование системы CodeIgniter.

**2. Формирование требований к интернет-сервису «Налоговый помощник»**

Все требования к проектируемой системе предлагается размещать на нескольких иерархических уровнях. На самом нижнем уровне располагаются требования, которые одинаково подходят для автоматизации технологических процессов в целом без учета особенностей конкретной прикладной области. Здесь необходимо обратиться к ГОСТам и другим нормативным документам.

Второй уровень предполагает бизнес-требования. Бизнес-требования (business requirements) содержат высокоуровневые цели организации или заказчиков системы. При формировании бизнес-требований определяются границы системы.

Следующий уровень составляют требования к конкретной системе. Крое того, в стандарте IEEE 830-1993 Спецификация требований к ПО (Software Requirements Specification – SRS) [9] проведено деление всех требований на две группы. Первая группа документирует потребности заказчика и записывается на языке, понятном заказчику – это т.н. С-требования. Вторая группа документирует требования в специальной, структурированной форме. Этот документ называют требованиями разработчика, или D-требованиями.

**2.1 Общие требования к созданию АСОИУ**

В ГОСТ 24.103-84 «Автоматизированные системы управления. Основные положения» [10] перечислены общесистемные принципы, которые необходимо соблюдать при создании автоматизированных систем обработки информации и управления (АСОИУ):

* системность – заключается в том, что при создании, функционировании и развитии автоматизированных систем управления (АСУ) должны быть установлены и сохранены связи между структурными элементами, обеспечивающие ее целостность;
* развитие – заключается в том, что АСУ должна создаваться с учетом возможности пополнения и обновления функций и видов ее обеспечения путем доработки программных и (или) технических средств или настройкой имеющихся средств;
* совместимость – заключается в обеспечении способности взаимодействия АСУ различных видов и уровней в процессе их совместного функционирования;
* стандартизация и унификация – заключается в рациональном применении типовых, унифицированных и стандартизованных элементов при создании и развитии АСУ;
* эффективность – заключается в достижении рационального соотношения между затратами на создание АСУ и целевыми эффектами, получаемыми при ее функционировании.

Кроме того, в п.3.4. ГОСТ 24.103-84 [10] при создании АСУ рекомендуется максимально использовать типовые проектные решения, пакеты прикладных программ, унифицированные пакеты а также применять для новых объектов управления ранее созданные проекты АСУ.

Это положение полностью соответствует принципам инженерии программного обеспечения (ПО), в особенности, концепции повторного использования компонентов.

ГОСТ 24.104-85 «Автоматизированные системы управления. Общие требования» [11] устанавливает требования к каждому виду обеспечения отдельно. Перечислим те из них, на которые нужно обратить внимание:

Требования к программному обеспечению:

1. В АСУ должны быть преимущественно использованы системы управления базами.
2. Данных систем управления базами данных (СУБД), зарегистрированные в установленном порядке.
3. Программное обеспечения АСУ должно иметь средства диагностики технических средств АСУ и контроля на достоверность входной информации.

**2.2. Бизнес-требования**

Интернет-сервис «Налоговый помощник» позволяет получить налогоплательщику информацию:

1. О налоговых льготах, установленных федеральным законодательством:

* налоговым кодексом Российской Федерации – по транспортному налогу, налогу на имущество организаций и земельному налогу,
* законом Российской Федерации от 09.12.1991 № 2003-1 «О налогах на имущество физических лиц» по налогу на имущество физических лиц.

1. О ставках и льготах установленных на региональном и местном уровнях:

* законами субъектов Российской Федерации (по транспортному налогу),
* нормативными правовыми актами органов местного самоуправления (по земельному налогу и налогу на имущество физических лиц).

1. О реквизитах соответствующих законов субъектов Российской Федерации, нормативных правовых актов муниципальных образований об установлении соответствующих налогов на их территории.
2. О документах, которые необходимо представить в налоговый орган, для подтверждения права на льготу или применения налоговой льготы.

В системе налогоплательщику доступно скачивание нормативных документов.

В системе необходимо обеспечить редактирование информации, описанной выше. В системе должен вестись учет посещений и браузеров, с которых входил налогоплательщик.

По требованию заказчика интернет-сервис должен быть реализован на языке PHP. Это обусловлено квалификацией налоговых сотрудников, которые будут поддерживать в течение жизненного цикла программного обеспечения.

**2.3. C-требования**

В начале проектирования системы было проведено обследование объекта автоматизации, были выбраны функции системы, которые подлежат автоматизации. В ходе этого обследования были организованы интервью с представителями заказчика, по результатам которых определили желания и потребности.

Список требований заказчика:

1. Пользовать выбирает район округа, в котором находится его населенный пункт, с помощью карты районов округа.
2. Пользователью выбирает населенный пункт, для которого необходимо получить информацию о налоговых ставках и льготах.
3. Пользователь может выбрать вид налога, по которому необходимо просмотреть информцию.
4. Налоговый сотрудник входит в систему.
5. Налоговый сотрудник может редактировать информацию о налоге.
6. Налоговый сотрудник может просматривать статистику по использованию интернет-сервиса.

**2.3.1. Описание диаграммы вариантов использования стандарта UML**

Диаграмма Use Case предназначены для создания списка операций, которые выполняет система и часто ее называют диаграммой функций, потому что на основе набора таких диаграмм строится набор функций, выполняемых системой и описываются сценарии поведения объектов, которые взаимодействуют с системой [12(В курс15)]. Диграмма вариантов использования для интернет-сервиса «Налоговый помощник» представлена на рисунке 2.1.

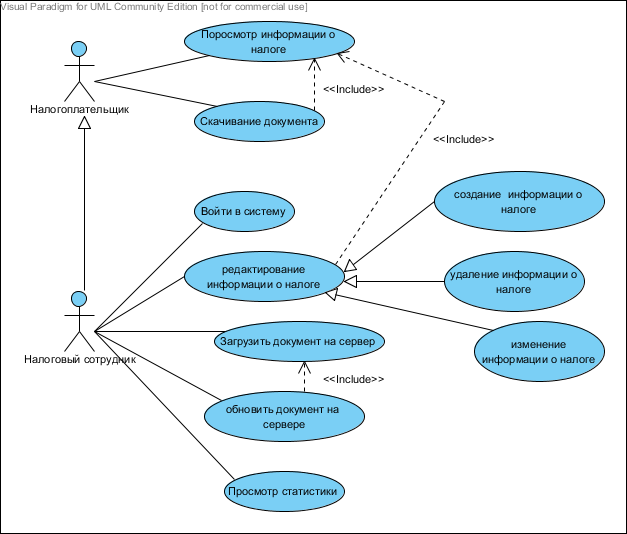


Рисунок 2.1 – Диаграмма вариантов использования UML

Согласно диаграмме в системе присутствуют две роли: налогоплательщик и налоговый сотрудник. Налогоплательщик в системе может просмотреть информацию об определенном виде налога и скачать по нему нормативный документ. Для пояснения некоторых вариантов использования приводится спецификация. Детальное описание варианта использования «Просмотр информации о налоге» для роли «Налогоплательщик» представлена спецификацией в таблице 2.1.

Таблица 2.1 – Спецификация для варианта использования «Просмотр информации о налоге»

|  |
| --- |
| Прецедент: ПросмотретьИнформациюОНалоге |
| ID:2 |
| Краткое описание: позволяет налогоплательщику просмотреть информацию о налоговых ставках по каждому виду налога для выбранного населенного пункта. |
| Главные актеры: налогоплательщик  Второстепенные актеры: нет |
| Предусловия: Налогоплательщик должен выбрать район |
| Основной поток:   1. include(Выбор населенного пункта) 2. Налогоплательщик выбирает вид налога 3. Система показывает данные по налогу для выбранного населенного пункта |
| Постусловия: нет |
| Альтернативные потоки: нет |

Для пояснения варианта использования «Редактирование информации о налоге» в таблице 2.2. представлена спецификация.

Таблица 2.2 – Спецификация варианта использования «Редактирование информации о налоге»

|  |
| --- |
| Прецедент: РедактироватьИнформациюНалога |
| ID:4 |
| Краткое описание: позволяет налоговому сотруднику изменять информацию налога для выбранного населенного пункта |
| Главные актеры: налоговый сотрудник  Второстепенные актеры: нет |
| Предусловия: Налоговый сотрудник входит в систему |
| Основной поток:   1. include(Выбор населенного пункта) 2. include(Просмотр информации налога) 3. Система отображает информацию по налогу 4. Налоговый сотрудник редактирует информацию |
| Постусловия: Данные изменений сохраняются в БД |
| Альтернативные потоки: нет |

**2.3.2. Описание диграммы с применением методологии DFD**

Диаграммы потоков данных (Data Flow Diagrams — DFD) представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами. Диаграмма потоков данных представлена в приложении Б на рисунке Б.1. Согласно диаграмме система взаимодействует с двумя внешними сущностями «Налогоплательщик» и «Налоговый сотрудник». Проводя декомпозицию модели DFD «TO-BE», приведенную в приложении Б на рисунке Б.2, система имеет 3 блока: получение информации о налоговых ставках и льготах, блок входа в систему и административная панель налогового сотрудника.

На декомпозиции блока «Получение информации о налоговых ставках и льготах» (Приложение Б рисунок Б.3) налогоплательщик заходит на главную страницу и с помощью интерактивной карты выбирает район или город соответствующий инспекции. Затем необходимо выбрать муниципальное образование, для которого необходимо получить информацию, и нажать кнопку запроса информации. Система обрабатывает запрос и отображает информацию налогоплательщику.

Для доступа к административной панели налоговому сотруднику необходимо войти в систему. В административной панели (Приложение Б рисунок Б.4) налоговому сотруднику доступные функции добавления, редактирования и удаления информации о налоговых ставках и льготах. Также сотрудник, задав параметры, может посмотреть общую статистику посещения интернет-сервиса, статистику по браузерам и статистику по операционным системам.

**2.4. D-требования**

Разработчикам программного обеспечения нужна база для проектирования и разработки. Эта база представляет собой детальные требования (D-требования). D-требования состоят из полного списка конкретных свойств и функциональности, которую должна иметь программа. Каждое из этих требований пронумеровано, помечено и отслеживается по ходу разработки. D-требования должны быть согласованы с С-требованиями.

**2.4.1. Функциональные требования**

Функциональные требования – определяют работу, которую должно выполнять проектируемая система.

1. Система должна отображать карту выбора районов округа.
2. При выборе района система должна запрашивать и выводить список населенных пунктов.
3. При выборе населенного пункта и нажатии кнопки подтверждения система должна запрашивать и выводить информацию о налогах, налоговых ставках и льготах.
4. Система должна учитывать посещение налогоплательщика.
5. Система должна авторизовать и аутентифицировать налогового сотрудника.
6. Система должна проверять данные, вводимые налоговым сотрудником при редактировании информации о налоге, налоговых ставках и налоговых льготах.
7. По запросу налогового сотрудника система должна отображать статистику посещений интернет-сервиса.

**2.4.2. Нефункциональные требования**

Рассмотрим нефункциональные требования:

2.1. Производительность. Требования к производительности определяют временные ограничения, которые должны быть выполнены в программе. По требованию заказчика обработка запросов должна производится не более 5 секунд, это связано с обеспечением комфортности работы пользователей.

2.2. Надежность и безопасность. Для решения вопроса обеспечения безопасности системы, связанного с исключением возможности искажения данных посредством несанкционированного доступа в системе, должна быть предусмотрена предварительная аутентификация пользователей перед началом работы. Так же необходимо обеспечить устойчивость к нагрузкам, система должна справляться с запросами не менее 5000 пользователей одновременно.

2.3. Обработка ошибок. Система должна проверять корректность введенных пользователем данных, возможность подключения к БД, и в случае ошибки выдавать соответствующее сообщение.

2.4. Ограничения. Ограничения на проектирование или реализацию описывают границы или условия того, как приложение должно быть задумано и разработано. Ввиду сложившихся традиций организации и опыта программистов накладывается ограничения на язык программирования – система должна быть реализована на PHP. Приложение должно работать с базой данных MS SQL.

**2.4.3. Обратные требования**

Нефункциональные требования – это тот функционал, который система не обеспечивает.

* версия для печати,
* история изменений в БД,
* ролевая политика.

**2.4.4. Описание диаграмм последовательности UML**

Обмен сообщениями происходит в определенной последовательности, и диаграмма последовательности (Sequence diagram) позволяют получить отражение этого обмена во времени.

Диаграмма последовательностей отражают временную зависимость событий, происходящих в рамках варианта использования. Стрелки соответствуют сообщениям, передаваемым между действующим лицом и объектами для выполнения требуемой функции [13 (в курс18)].

Диаграммы последовательности сделаны для вариантов использования «Выбор муниципального образования», «Просмотр информации о налоге», «Редактирование налога».

На диаграмме последовательности (Рисунок 2.2) действующее лицо налогоплательщик выбирает объект «Район» класса «Муниципальное образование» (1). Затем объект «Район» получает список городов (1.1), входящий в его состав. И информация передается обратно налогоплательщику(2).

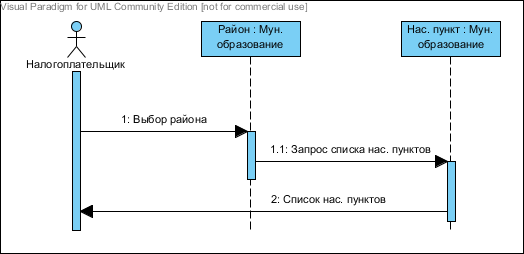
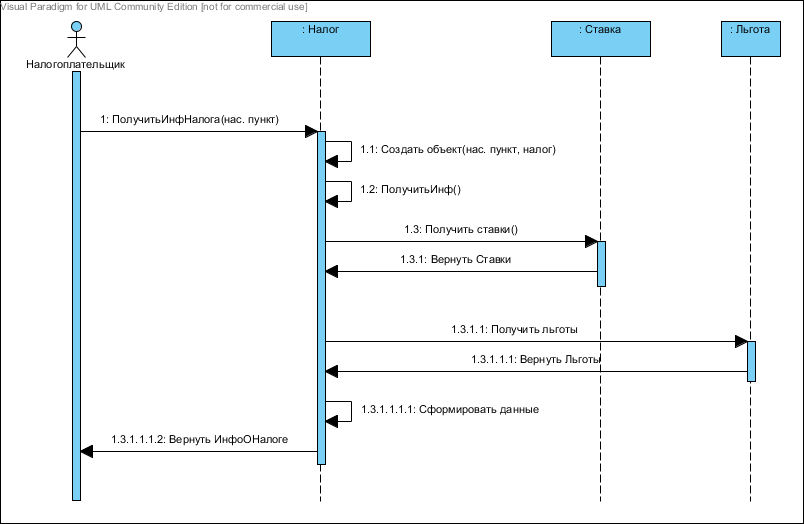
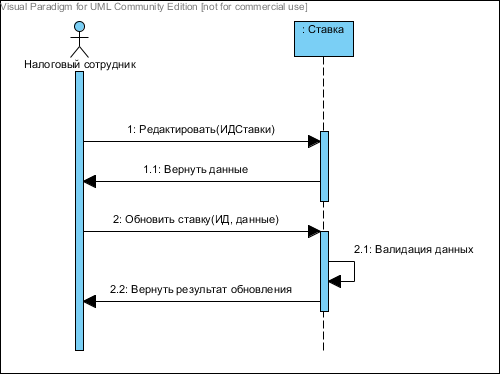


Рисунок 2.2 – Диаграмма последовательности для варианта использования «Выбор населенного пункта»



* 1. − Диаграмма последовательности для варианта использования «Просмотр информации о налоге»



* 1. − Диаграмма последовательности для варианта использования «Редактирование налога»
  2. **Черновой эскиз ГПИ**

Согласно требованиям был разработан эскиз графического интерфейса пользователя.

Интерфейс выбора района (Рисунок 2.4) представляет собой интерактивную карту Ханты-Мансийского автономного округа.

Для выбора муниципального образования представлена на рисунке

Интерфейс представляет собой форму выбора из списка элементов.

Отображение информации о налоговых ставках представлено на рисунке

**3. Техническое задание**

Техническое задание (ТЗ) на АСОИУ является основным документом, определяющим требования и порядок создания (развития или модернизации) автоматизированной системы, в соответствии с которым проводится разработка АСОИУ и ее приемка при вводе в эксплуатацию. ТЗ включает в себя системное описание расширенных требований к разрабатываемому изделию и составляется на основе исходного задания по курсовому проекту с учетом всей проведенной выше работы в отношении формулирования и спецификации требований [14].

Помимо требований, описанных ранее в дипломном проекте, в техническом задание также приведен перечень и критерии отказов системы, то есть причины, по которым могут возникнуть неполадки или сбои в работе системы. Также в ТЗ указаны требования к форме представления выходной информации и требования к используемой БД.

Техническое задание на создание интернет-сервиса «Налоговый помощник», оформленное в соответствии с ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» [14], расположено в приложении Г.

**4. Разработка концепции АСОИУ**

**4.1. Описание архитектуры приложения**

С целью выбора платформы для реализации проектируемой системы на страницах пояснительной записки в разделе 1.3 были описаны xtnsht структуры программной системы: CodeIgniter, Zend Framework, Yii и CakePHP. Как показал анализ, наилучшими характеристиками для выполнения данного проекта обладает CodeIgniter.

Фреймворк CodeIgniter построена с использованием распространенной архитектуры (шаблона проектирования) Модель-Представление-Контроллер (Model-View-Controller − MVC). MVC упрощает разработку приложений посредством его деления на несколько слоев, каждый из которых имеет соответствующие роли и возможности. Слой модели отвечает за поддержание состояния приложения. Он инкапсулирует данные приложения и бизнес логику. Слой представления обеспечивает реализацию пользовательского интерфейса приложения. Слой управления интерпретирует обращение пользователей и ответ пользователям посредством передачи данных между слоем модели и слоем представления.



Рисунок 4.1 – Ход выполнения приложения на базе CodeIgniter

Ход выполнения приложения, представленный на рисунке 4.1, следует следующему алгоритму:

1. index.php действует как фронт-контроллер, инициализируя основные ресурсы, необходимые для запуска CodeIgniter.
2. Роутер проверяет HTTP-запрос, чтобы определить, что с ним нужно сделать.
3. Если существует файл кэша, он будет отправлен в браузер, в обход нормального выполнения системы.
4. Безопасность. Перед тем, как будет загружен контроллер приложения, HTTP-запрос и любые данные, переданные пользователем фильтруются для обеспечения безопасности.
5. Контроллер загружает модели, библиотеки ядра, помощники, и любые другие ресурсы, требуемые для обработки конкретного запроса.
6. Финализированное отображение рендерится, и затем отправляется в браузер. Если включено кеширование, отображение сначала кешируется, таким образом оно может быть использовано при следующих запросах.

MVC-структура фреймворка CodeIgniter строго отделяет архитектурные слои приложения друг от друга, а правила построения каждого слоя в отдельности исключают необходимость дополнительной проработки вопросов проектирования межмодульных интерфейсов.

**4.2 Выбор шаблонов построения архитектурных слоев**

Для выбора шаблонов построения архитектурных слоев был проведен их анализ.

Проектирование архитектуры слоев ИС начинается с выбора паттерна слоя предметной области, который определит типовое решение слоя источника данных.

К организации слоя бизнес логики распространение получили три подхода []:

1. Сценарий транзакций. Организует логику вычислительного процесса преимущественно в виде единой процедуры, которая обращается к базе данных напрямую или при посредничестве кода тонкой оболочки. При использовании типового решения сценарий транзакций логика предметной области распределяется по транзакциям в системе.

Одно из преимуществ сценария транзакций заключается в том, что при написании сценария не нужно учитывать наличие и варианты функционирования других параллельных задач. Одна из проблем связана с повторением фрагментов кода. По мере усложнения бизнес-логики становится труднее поддерживать код в структурированном виде.

1. Модель предметной области. Типовое решение «модель предметной области» (domain model) строится на базе объектно-ориентированного подхода. Вместо использования одной подпрограммы, несущей в себе всю логику, которая соответствует набору функций, реализуемых системой, каждый объект наделяется только теми методами, которые отвечают его природе.

В первом приближении модели предметной области делят на два типа: «простые» модели предметной области. В этом случае структура домена совпадает с ER-моделью базы данных: одному объекту домена соответствует одна таблица БД; «сложные» модели – структура домена не совпадает с ER-моделью БД и включает в свой состав иерархии наследования, а также сложные сети мелких взаимосвязанных объектов. В простых моделях подчас достаточно применить варианты тривиального типового решения источника данных активная запись (Active Record), в то время как в сложных необходимо применить решение преобразователь данных (Data Mapper).

1. Модуль таблицы. Типовое решение предусматривает создание по одному классу на каждую таблицу, и единственный экземпляр класса содержит всю логику обработки данных.

Типовое решение «модуль таблицы» предусматривает создание по одному классу на каждую таблицу БД, единственный экземпляр класса содержит всю логику обработки данных таблицы. Основное отличие паттернов «модуль таблицы» и «модель предметной области» состоит в том, что если, например, приложение обслуживает множество заказов, в соответствии с моделью предметной области необходимо создать по одному объекту на каждый заказ, а при использовании модуля таблицы понадобится всего один объект, представляющий одновременно все заказы.

В большинстве случаев для решения общей задачи в процессе функционирования системы создается несколько модулей таблицы, которые могут манипулировать одним и тем же множеством записей.

Вместе с тем, модуль таблицы не позволяет воспользоваться всеми возможностями объектно-ориентированной методологии для реализации сложной бизнес-логики (в частности, не полноценно функционирует механизм полиморфизма). В этой связи, реализация крупных корпоративных приложений возможна только с использованием модели предметной области, не имеющей ограничений.

На этапе формирования требований были выявлены следующие классы анализа: «Муниципальное образование», «Налог», «Ставка», «Льгота» «Статистика», «Налоговый сотрудник».

Каждый класс имеет собственный набор атрибутов и методов. Следовательно, применение модели предметной области позволяет реализовать классы, основываясь на объектно-ориентированном подходе. Также одним из оснований является следование структуре платформы CodeIgniter, которая в качестве источника данных использует шаблон активная запись (Active Record).

Нередко при реализации слоя бизнес-логики его расщепляют на два подуровня, выделяя помимо модели предметной области или модуля таблицы, слой служб, который предназначен, например, для управления транзакциями или обеспечения безопасности. Кроме того, рассмотренные типовые решения реализации слоя бизнес-логики не являются взаимоисключающими. Например, сценарий транзакций может использоваться для некоторого фрагмента бизнес-логики, а модель предметной области или модуль таблицы – для оставшейся части. В CodeIgniter таким слоем служб выступает контроллер.

Контроллер MVC-модели выполняет обработку и диспетчеризацию HTTP-запросов. В контроллере MVC-модели используемой платформы применено типовое решение контроллер приложения.

Контролер приложения выполняет две основные функции: выбор логики домена, которую нужно применить в конкретной ситуации, и выбор представления, которое следует отобразить в ответ на запрос. Для осуществления этих функций контроллер приложения должен поддерживать две коллекции ссылок на классы: одну для команд, выполняющихся в слое домена, и одну для представлений. В качестве команд домена могут выступать объекты команд, являющиеся частью слоя контроллера приложения, либо ссылки на сценарий транзакции или методы объектов домена.

В качестве входного контроллера в CodeIgniter применяется контроллер запросов, который предполагает использование единственного объекта, предназначенного для обработки всех запросов, но включает в свой состав две части:

* web-обработчик – объект, который выполняет фактическое получение POST- и GET-запросов, поступивших на web-сервер. Он извлекает необходимую информацию из URL и входных данных запроса, после чего решает, какое действие необходимо инициировать, и делегирует его выполнение соответствующей команде.
* иерархия команд из числа которых web-обработчиком статически или динамически выбирается необходимая команда для обработки поступившего запроса. Статический выбор команды подразумевает проведение синтаксического анализа адреса URL и применение условной логики, а динамический – извлечение некоторого стандартного фрагмента адреса URL и динамическое создание экземпляра класса команды.

В качестве слоя источника данных CodeIgniter реализуется шаблон активная запись (Active Record). Как правило, активная запись включает в себя методы, предназначенные для выполнения следующих операций: (i) создание экземпляра активной записи на основе строки, полученной в результате выполнения SQL-запроса; (ii) создание нового экземпляра активной записи для последующей вставки в таблицу; (iii) статические методы поиска, выполняющие стандартные SQL-запросы и возвращающие активные записи; (iv) обновление БД и вставка в нее данных из активной записи; (v) извлечение и установка значений полей (get- и set-методы); (vi) реализация фрагментов бизнес-логики. Недостатком активной записи является тесная зависимость структуры ее объектов от структуры БД.

В MVC-модели представление взаимодействует с контроллером. Поэтому выбор типового решения для обеспечения взаимодействия «представление-контроллер» во многом зависит от используемой инструментальной среды.

Результаты выбора шаблонов архитектурных слоев информационной системы представлены на рисунке 4.2.

<<Слой представления>>

<<Сервисный слой>>

<<Слой домена>>

<<Слой источника данных>>

Представление по шаблону

Контроллер приложения

Модель предметной области

Активная запись

Рисунок 4.2 – Архитектура слоев интернет-сервиса «Налоговый помощник»

**4.3. Проектирование архитектурных слоев**

Классы системы были разделены на модули в соответствии с патерном MVC. Декомпозиция системы на модули производится с целью разделения системы на части, которые могут независимо конфигурироваться или поставляться; разрабатываться; размещаться в распределенной среде; изменяться без воздействия на остальные части системы; а также из соображений ограничения доступа к основным ресурсам.

В каждом слое были созданы пакеты для пользовательской и административной частей интернет-сервиса.

**5. Детальное проектирование**

На этапе детального проектирования уточняются классы, которые были определены в главе 4 «Проектирование архитектуры приложения», и разрабатывается документация, которая впоследствии может быть использована для реализации интернет-сервиса.

**5.1. Введение глобальных библиотек**

Для ускорения разработки интернет-сервиса были использованы в своем составе платформа CodeIgniter помощники и классы библиотеки [CIdoc]. Помощники помогают решать определенный набор задачи. В отличие от большинства других систем в CodeIgniter, помощники написаны не в ООП-формате. Они являются простыми процедурными функциями. Каждая функция-помощник выполняет одну специфическую задачу, вне зависимости от других функций.

В CodeIgniter имеется библиотека «DataBase», которая предоставляет интерфейс для работы с базой данных. Она использует модифицированный шаблон проектирования источника данных «ActiveRecord». В данном случае один класс не привязан к базе данных.

Для ведения статистики необходимо учитывать такие параметры как дата посещения, ip-адрес пользователя, используемы браузер, версия браузера. Для решения подобной задачи используется класс «UserAgent». Класс агента пользователя предоставляет функции, которые помогают идентифицировать информацию о браузере, мобильном устройстве или роботе, обращающихся к интеренет-сервису.

В интернет-сервисе вводимую пользователями информацию необходимо проверять на ошибки. CodeIgniter в своем составе имеет класс валидации «Form Validation», который позволяет минимизировать количество кода покрывает широкий набор задач подготовки и проверки данных.

Налоговые ставки и льготы отображаются в виде таблиц. Для построения таблиц используется класс «HTML Table». Класс таблиц предоставляет функции, которые позволяют вам автоматически генерировать HTML-таблицы из массивов или результатов выборок базы данных.

Для размещения форм на сайте был использован помощник Form. Помощник содержит функции создания различных типов форм.

Загрузка файлов осуществляется с помощью помощника «File».

Платформа CodeIgniter позволяет использовать библиотеки сторонних разработчиков. Для исполнения требований предоставления доступа к административной части, редактирования информации и просмотра статистики используются библиотеки сторонних разработчиков.

При анализе различных библиотек авторизации решение было принято в пользу библиотеки Ion Auth []. В отличии от остальных эта библиотека имеет большинство возможность по управлению пользователями, а также на данный момент продолжает поддерживаться разработчиком. Простой уровень инсталляции системы и минимальная настройка позволяет быстро адаптировать библиотеку под собственную систему. Библиотека имеет функции авторизации, распределения прав пользователей по группам, регистрации, активации и деактивации пользователейт [].

Для обеспечения функционала Create-Read-Update-Delete (CRUD) для администрирования информационной базы интернет-сервиса используется библиотека Grocery CRUD [].

Список возможностей библиотеки []:

* разбивка списка записей на страницы, сортировка, поиск по списку записей (все функции реализованы на ajax),
* автоматическая генерация полей input в соответствии с типом поля БД,
* реализация зависимостей 1-1, 1-n и n-n всего одной строкой кода,
* быстрый выбор тем оформления (сейчас их реализовано две: datatables и flexigrid),
* валидация вводимых значений на стороне клиента и на стороне сервера, согласно заданным правилам валидации,
* гибкий выбор полей, которые будут отображаться и редактироваться,
* блокировка операций удаления, редактирования и добавления записей,
* реализованы вызовы callbacks для самостоятельной настройки дополнительных действий после выполнения стандартной операции,
* расширяемость базовой модели Grocery CRUD.

**5.2. Описание диаграмм последовательности UML**

В соответствии с разработанной концепцией архитектуры построены диаграммы последовательности для вариантов использования «Выбор муниципального образования», «Просмотр информации о налоге», «Просмотр статистики».

Диаграмма последовательности для варианта использования «Выбор муниципального образования» представлена в приложении Такомто на рисунке Такомто. Действующим лицом является налогоплательщик. Он запрашивает главную страницу сайта (1) у контроллера «TaxStavki». Контроллер загружает представление (1.1), и приложение передает его клиенту (1.2).

Далее от налогоплательщика после выбора района передается контроллеру параметр (2), содержащий название инспекции, к которой относится выбранное муниципальное образование. Контроллер создает объект инспекции (2.1) и вызывает метод получения списка муниципальных образований (2.2). Метод запрашивает эту информацию, вызывая функцию (2.2.1) класса «Municipality» для загрузки из базы списка муниципальный образований. Далее информация возвращается контроллеру (шаги 2.2.1.1, 2.2.1.1.1). Потом информация формируется (2.2.1.1.1.1) и загружается отображение (2.2.1.1.1.2), которое передается налогоплательщику (2.2.1.1.1.2.1).

В приложении ТТТ на рисунке ТТТ показана диаграмма последовательности для варианта использования просмотр информации о налоге. После выбора муниципального образования налогоплательщик через форму посылает запрос на получение информации (1). Контроллер методом POST считывает значение с формы. Затем в контроллере создается объект класса муниципальное образование. Для получения информации о налоге вызывается метод «get\_tax(3.2)», в котором создается объект класса налог (4.1). Далее вызываются методы объекта налог для сбора информации о налоге. Метод обращается за информацией к объекту налог (4.3), получая информацию о нормативном документе (4.4). Получение списка налогов происходит обращением за информацией объекту класса ставка (5.2). Информация о льготах запрашивается подобным образом. Когда объект класса муниципальное образование получил информацию (3.3), контроллер загружает представление (3.4) и передает ее налогоплательщику (2).

**5.3. Описание диаграмм классов**

**5.3.1. Диаграмма классов слоя домена**

Слой домена описывает элементы бизнес-логики, т.е. основные классы, методы которых выполняют функциональные требования. Сервисный слой разработан как типовое решение «Модель предметной области» (Domain Model) [19]. Каждый класс содержит методы, связанные с определенной работой.

Диаграмма классов слоя домена представлена на рисунке 5.1. На диаграмме класс «Инспекция» агрегирует с классом «Муниципальное образование». Агрегация встречается, когда один класс является коллекцией или контейнером других. Причём по умолчанию, агрегацией называют агрегацию по ссылке, то есть когда время существования содержащихся классов не зависит от времени существования содержащего их класса. Если контейнер будет уничтожен, то его содержимое — нет. Между классами «Муниципальное образование» и «Налог» установлена связь композиции. Композиция − форма агрегации, при которой части принадлежат целому, причем время жизни частей совладает со временем жизни композита. Часть может принадлежать только одному композиту.

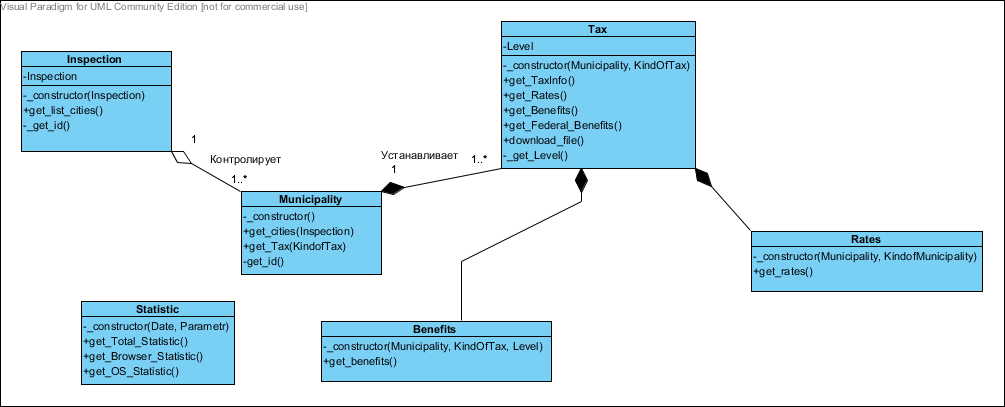
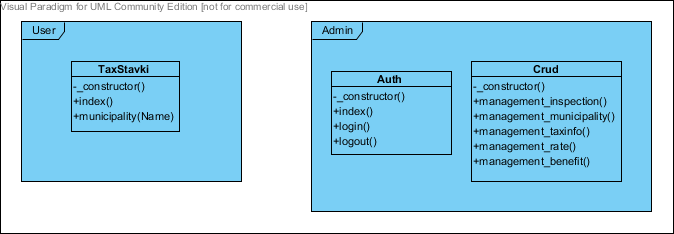
****

Рисунок 5.1 – Диаграмма классов слоя домена

**5.3.2. Диаграмма классов сервисного слоя**

На рисунке 5.2 представлена диаграмма классов слоя домена. Слой разделен на пользовательскую и административную часть. Пользовательская часть реализуется контроллером «TaxStavki». Административная часть состоит из контроллера «Auth», которая реализует вход налогового сотрудника в систему.

****

**5.3.3. Диаграмма классов слоя представления**

Слой представлений (Приложение М) описывает формы существующие в системе. Каждый класс соответствует своей форме. Атрибуты класса обозначают элементы ввода или отображения информации. Списки в классах указывают на то, что на форме будет использоваться элемент отображающий записи считанные из таблицы БД. Методы класса это результат нажатия управляющих кнопок.

**6. Реализация**

В этой главе рассматривается методы организации сбора и хранения данных, реализации классов и методов архитектурных слоев интернет-сервиса «Налоговый помощник» с помощью выбранных инструментов и средств программирования.

**6.1. Развертывание инструментальной среды**

Согласно требованиям для функционирования приложения на платформе CodeIgniter требуется веб-сервер, с установленным интерпретатором языка PHP версии 5.1.6 и база данных.

В Управлении федеральной налоговой службы по ХМАО − Югре на серверах для web-приложений используется серверные операционные системы семейства Windows. В комплексе приложений операционной системы имеется набор интернет служб Internet Information Services (IIS). Основным компонентом IIS является веб-сервер — служба WWW (World Wild Web), которая предоставляет клиентам доступ к сайтам по протоколам HTTP и HTTPS. Для публикации содержимого в сети необходимо создать конфигурацию сайта в панели управления сайтами IIS Manager согласно документации [http://technet.microsoft.com/ru-ru/library/cc772350(v=ws.10).aspx].

После создания конфигурации в директорию сайта был инсталлирован и настроен CodeIgniter. Файлы настройки находятся в папке \application\config. В файле config.php необходимо указать адрес сайта и задать ключ шифрования, так как используются сессии. Настройки соединения с базой данных прописываются в файле database.php.

**6.2. Реализация слоя домена**

**6.3. Реализация сервисного слоя**

Сервисный слой реализуется по принципу шаблона проектирования контроллер приложения.

Контроллер работает как посредник между моделью, отображением и другими ресурсами, требуемыми для обработки HTTP-запроса и генерации веб-страницы.

Контроллер в CodeIgniter – это класс, который ассоциируется URI.

Класс контроллера должен начинаться с заглавной буквы. «Индексная» функция index() всегда загружается по умолчанию, если второй сегмент унифицированного идентификатора ресурса (URI) пуст.

Вызов функции контроллера осуществляется запросом «example.com/index.php/blog/comments/».

Пользовательская часть реализуется с помощью класса «TaxStavki». Листинг класса представлен в приложении лдл. При обращении к классу выполняется конструктор, который загружает библиотеки и помощники, которые используются в классе. Индексная функция загружает главную страницу сайта. При обращении к методу municipality() с помощью класса валидации выполняется проверка формы выбора муниципального образования. Если выбор сделан, то с использованием встроенного класса Input считывается введенное значение. Далее создается объект налог с параметром вида налога. Затем вызывается метод объекта для получения информации о налоге. Метод вызывается для каждого вида налога. Данные формируются в массив. Последним шагом происходит вызов функции загрузки отображения с передачей данных.

Функции авторизации реализуются в классе «Auth». Конструктор класса загружает библиотеки авторизации, сессии, валидации форм. Индексная функция проверяет авторизован ли пользователь. Если да, то происходит перемещение на страницу административной панели. В обратном случае пользователю вызывается метод login(), который загружает файл отображения формы ввода. Для выхода из административной части используется функция logout().

**6.4. Реализация слоя представления**

Отображением является информация, которая будет отправлена пользователю. Отображения это обычные веб-страницы, но в CodeIgniter отображения могут быть также фрагментами страниц, например шапкой и подвалом, что исключает дублирование кода в приложении. Отображения никогда не вызываются напрямую, они должны быть загружены контроллером.

Для пользовательской части созданы экраны главной страницы, страницы выбора муниципального образования и отображения информации о налоговых ставках и льготах. Административная часть содержит экраны авторизации и экран панели администратора. Остальные экраны формируются библиотекой Grocery CRUD.

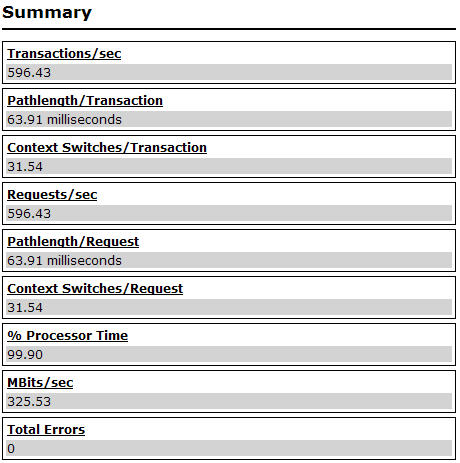
Загрузка отображения осуществляется вызовом функции $this->load->view('name'), где 'name' это имя файла отображения.

Для отображения страниц используется язык разметки HTML в связке с каскадными таблицами стилей (CSS).

**7. Апробация**

**7.1. Нагрузочное тестирование**

Нагрузочное тестирование применяется для анализа работы информационных систем на различных уровнях нагрузки. Для нагрузочного тестирования использовалась свободно распространяемая утилита wCat 6.3. При тестировании. Технические характеристики сервера: процессор Intel Xeon 3 ГГц, оперативная память 5 Гб. На рисунке 5 приведен отчет утилиты wCat 6.3 о результатах тестирования.



На рисунке видно, что сервер способен обрабатывать до 596 запросов в секунду.

**7.2. Модульное тестирование**

**7.3. Опытная эксплуатация**

С целью проверки в реальных условиях эксплуатации работоспособности программного обеспечения, была проведена опытная эксплуатация данного сервиса среди инспекций округа. В приложении Л приведена копия приказа УФНС Ханты-Мансийского автономного округа − Югры.

**8. Организационно-экономическая часть**

Экономическая эффективность − результативность экономической системы, выражающаяся в отношении полезных конечных результатов ее функционирования к затраченным ресурсам.

Следует отметить, что данный проект не направлен на достижение прямой экономической выгоды, а в первую очередь рассчитан на достижение социального эффекта в области оказания услуг населению.

**9. Безопасность жизнедеятельности**

Безопасность жизнедеятельности (БЖД) – система знаний, обеспечивающая безопасность обитания человека в производственной и непроизводственной среде, и развитие деятельности по обеспечению безопасности в перспективе с учетом антропогенного влияния на среду обитания. Как всякая наука, БЖД имеет свои цели, задачи и принципы, используемые для решения практических и теоретических задач.

Цель БЖД исходит из определения этой науки и представляет собой достижение безопасности в среде обитания. Безопасность человека определяется отсутствием производственных и непроизводственных аварий, стихийных бедствий и других природных явлений и опасных факторов, вызывающих травмы или резкое ухудшение здоровья, вредных факторов, вызывающих заболевание человека и снижение его работоспособности.

Исходя из этого, целями БЖД являются:

• создание комфортных условий обитания человека;

• идентификация воздействий факторов среды обитания на человека;

• разработка мер защиты человека и среды обитания от негативных воздействий;

• обеспечение безопасности, экологичности техники и технологических процессов при их проектировании и эксплуатации;

Охрана здоровья трудящихся, обеспечение безопасности условий труда, ликвидация профессиональных заболеваний и производственного травматизма составляет одну из главных забот человеческого общества. Обращается внимание на необходимость широкого применения прогрессивных форм научной организации труда, сведения к минимуму ручного, малоквалифицированного труда, создания обстановки, исключающей профессиональные заболевания и производственный травматизм [21].

На рабочем месте должны быть предусмотрены меры защиты от возможного воздействия опасных и вредных факторов производства. Уровни этих факторов не должны превышать предельных значений, оговоренных правовыми, техническими и санитарно-техническими нормами. Эти нормативные документы обязывают к созданию на рабочем месте условий труда, при которых влияние опасных и вредных факторов на работающих либо устранено совсем, либо находится в допустимых пределах.

Данный раздел дипломного проекта посвящен рассмотрению следующих вопросов:

* определение оптимальных условий труда,
* расчет освещенности,
* расчет уровня шума.

**9.1. Характеристика условий труда**

Научно-технический прогресс внес серьезные изменения в условия производственной деятельности работников умственного труда. Их труд стал более интенсивным, напряженным, требующим значительных затрат умственной, эмоциональной и физической энергии. Это потребовало комплексного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха.

В настоящее время компьютерная техника широко применяется во всех областях деятельности человека. При работе с компьютером человек подвергается воздействию ряда опасных и вредных производственных факторов: электромагнитных полей (диапазон радиочастот:ВЧ, УВЧ и СВЧ), инфракрасного и ионизирующего излучений, шума и вибрации, статического электричества и др. [22].

Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей позы человека-оператора.

В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

**9.2. Требования к производственным помещениям**

**9.2.1. Окраска и коэффициенты отражения**

Окраска помещений и мебели должна способствовать созданию благоприятных условий для зрительного восприятия, хорошего настроения.

Источники света, такие как светильники и окна, которые дают отражение от поверхности экрана, значительно ухудшают точность знаков и влекут за собой помехи физиологического характера, которые могут выразиться в значительном напряжении, особенно при продолжительной работе. Отражение, включая отражения от вторичных источников света, должно быть сведено к минимуму. Для защиты от избыточной яркости окон могут быть применены шторы и экраны [23].

В зависимости от ориентации окон рекомендуется следующая окраска стен и пола:

* окна ориентированы на юг: стены − зеленовато-голубого или светло-голубого цвета; пол − зеленый;
* окна ориентированы на север: стены − светло-оранжевого или оранжево-желтого цвета; пол − красновато-оранжевый;
* окна ориентированы на восток: - стены желто-зеленого цвета;
* пол зеленый или красновато-оранжевый;
* окна ориентированы на запад: - стены желто-зеленого или голубовато-зеленого цвета; пол зеленый или красновато-оранжевый.

В помещениях, где находится компьютер, необходимо обеспечить следующие величины коэффициента отражения: для потолка: 60…70%, для стен: 40…50%, для пола: около 30%. Для других поверхностей и рабочей мебели: 30…40%.

**9.2.2. Освещение**

Правильно спроектированное и выполненное производственное освещение улучшает условия зрительной работы, снижает утомляемость, способствует повышению производительности труда, благотворно влияет на производственную среду, оказывая положительное психологическое воздействие на работающего, повышает безопасность труда и снижает травматизм.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности.

Существует три вида освещения – естественное, искусственное и совмещенное (естественное и искусственное вместе) [24].

Естественное освещение – освещение помещений дневным светом, проникающим через световые проемы в наружных ограждающих конструкциях помещений. Естественное освещение характеризуется тем, что меняется в широких пределах в зависимости от времени дня, времени года, характера области и ряда других факторов.

Искусственное освещение применяется при работе в темное время суток и днем, когда не удается обеспечить нормированные значения коэффициента естественного освещения (пасмурная погода, короткий световой день). Освещение, при котором недостаточное по нормам естественное освещение дополняется искусственным, называется совмещенным освещением.

Искусственное освещение подразделяется на рабочее, аварийное, эвакуационное, охранное. Рабочее освещение, в свою очередь, может быть общим или комбинированным. Общее − освещение, при котором светильники размещаются в верхней зоне помещения равномерно или применительно к расположению оборудования. Комбинированное - освещение, при котором к общему добавляется местное освещение.

Согласно СНиП II-4-79 [] в помещений вычислительных центров необходимо применить систему комбинированного освещения.

При выполнении работ категории высокой зрительной точности (наименьший размер объекта различения 0,3…0,5мм) величина коэффициента естественного освещения (КЕО) должна быть не ниже 1,5%, а при зрительной работе средней точности (наименьший размер объекта различения 0,5…1,0мм) КЕО должен быть не ниже 1,0%. В качестве источников искусственного освещения обычно используются люминесцентные лампы типа ЛБ или ДРЛ, которые попарно объединяются в светильники, которые должны располагаться над рабочими поверхностями равномерно [23].

Требования к освещенности в помещениях, где установлены компьютеры, следующие: при выполнении зрительных работ высокой точности общая освещенность должна составлять 300лк, а комбинированная − 750лк; аналогичные требования при выполнении работ средней точности − 200 и 300лк соответственно.

Кроме того все поле зрения должно быть освещено достаточно равномерно – это основное гигиеническое требование. Иными словами, степень освещения помещения и яркость экрана компьютера должны быть примерно одинаковыми, т.к. яркий свет в районе периферийного зрения значительно увеличивает напряженность глаз и, как следствие, приводит к их быстрой утомляемости.

**9.2.3. Параметры микроклимата**

Параметры микроклимата могут меняться в широких пределах, в то время как необходимым условием жизнедеятельности человека является поддержание постоянства температуры тела благодаря терморегуляции, т.е. способности организма регулировать отдачу тепла в окружающую среду. Принцип нормирования микроклимата – создание оптимальных условий для теплообмена тела человека с окружающей средой.

Вычислительная техника является источником существенных тепловыделений, что может привести к повышению температуры и снижению относительной влажности в помещении. В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. В санитарных нормах СН24571 установлены величины параметров микроклимата, создающие комфортные условия. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения (таблица 9.1) [22].

Объем помещений, в которых размещены работники вычислительных центров, не должен быть меньше 19,5м3/человека с учетом максимального числа одновременно работающих в смену. Нормы подачи свежего воздуха в помещения, где расположены компьютеры, приведены в таблице 9.2.

Таблица 9.1 **−** Параметры микроклимата для помещений, где установлены компьютеры

|  |  |  |
| --- | --- | --- |
| Период года | Параметр микроклимата | Величина |
| Холодный | Температура воздуха в помещении Относительная влажность  Скорость движения воздуха | 22…24°С  40…60%  до 0,1м/с |
| Теплый | Температура воздуха в помещении Относительная влажность  Скорость движения воздуха | 23…25°С  40…60%  0,1…0,2м/с |

**Таблица 9.2** − Нормы подачи свежего воздуха в помещения, где расположены компьютеры

|  |  |
| --- | --- |
| Характеристика помещения | Объемный расход подаваемого в помещение свежего воздуха, м3 /на одного человека в час |
| Объем до 20м3 на человека  20…40м3 на человека  Более 40м3 на человека | Не менее 30  Не менее 20  Естественная вентиляция |

Для обеспечения комфортных условий используются как организационные методы (рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха), так и технические средства (вентиляция, кондиционирование воздуха, отопительная система).

**9.2.4. Шум и вибрация**

Шум ухудшает условия труда оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Такие нарушения в работе ряда органов и систем организма человека могут вызвать негативные изменения в эмоциональном состоянии человека вплоть до стрессовых. Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума [выше 80 дБ(А)] на слух человека приводит к его частичной или полной потере [25].

В табл. 9.3 указаны предельные уровни звука в зависимости от категории тяжести и напряженности труда, являющиеся безопасными в отношении сохранения здоровья и работоспособности.

**Таблица 9.3**−Предельные уровни звука, дБ, на рабочих местах.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория  напряженности труда | Категория тяжести труда | | | |
| I. Легкая | II. Средняя | III. Тяжелая | IV. Очень тяжелая |
| I. Мало напряженный | 80 | 80 | 75 | 75 |
| II. Умеренно напряженный | 70 | 70 | 65 | 65 |
| III. Напряженный | 60 | 60 | - | - |
| IV. Очень напряженный | 50 | 50 | - | - |

Уровень шума на рабочем месте операторов ПК не должен превышать 50дБА, а в залах обработки информации на вычислительных машинах65дБА. Для снижения уровня шума стены и потолок помещений, где установлены компьютеры, могут быть облицованы звукопоглощающими материалами. Уровень вибрации в помещениях вычислительных центров может быть снижен путем установки оборудования на специальные виброизоляторы.

**9.2.5. Электромагнитное и ионизирующее излучения**

Большинство ученых считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Однако исчерпывающих данных относительно опасности воздействия излучения от мониторов на работающих с компьютерами не существует и исследования в этом направлении продолжаются [22].

Допустимые значения параметров неионизирующих электромагнитных излучений от монитора компьютера представлены в табл. 8.4.

Максимальный уровень рентгеновского излучения на рабочем месте оператора компьютера обычно не превышает 10мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10…100мВт/м2.

**Таблица 9.4** − Допустимые значения параметров неионизирующих электромагнитных излучений (в соответствии с СанПиН 2.2.2.542-96)

|  |  |
| --- | --- |
| Наименование параметра | Допустимые значения |
| Напряженность электрической составляющей электромагнитного  поля на расстоянии 50см от поверхности видеомонитора | 10В/м |
| Напряженность магнитной составляющей электромагнитного  поля на расстоянии 50см от поверхности видеомонитора | 0,3А/м |
| Напряженность электростатического поля не должна превышать:  для взрослых пользователей  для детей дошкольных учреждений и учащихся  средних специальных и высших учебных заведений | 20кВ/м  15кВ/м |

Для снижения воздействия этих видов излучения рекомендуется применять мониторы с жидкокристаллическим дисплеем или с пониженным уровнем излучения (MPR-II, TCO-92, TCO-99), устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

**9.3. Эргономические требования к рабочему месту**

Проектирование рабочих мест, снабженных видеотерминалами, относится к числу важных проблем эргономического проектирования в области вычислительной техники.

Рабочее место и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Большое значение имеет также характер работы. В частности, при организации рабочего места оператора ПК должны быть соблюдены следующие основные условия: оптимальное размещение оборудования, входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения.

Эргономическими аспектами проектирования видеотерминальных рабочих мест, в частности, являются: высота рабочей поверхности, размеры пространства для ног, требования к расположению документов на рабочем месте (наличие и размеры подставки для документов, возможность различного размещения документов, расстояние от глаз пользователя до экрана, документа, клавиатуры и т.д.), характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места [26].

Главными элементами рабочего места оператора ПК являются стол и кресло. Основным рабочим положением является положение сидя.

Рабочая поза сидя вызывает минимальное утомление оператора ПК. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства.

Моторное поле − пространство рабочего места, в котором могут осуществляться двигательные действия человека.

Максимальная зона досягаемости рук − это часть моторного поля рабочего места, ограниченного дугами, описываемыми максимально вытянутыми руками при движении их в плечевом суставе.

Оптимальная зона − часть моторного поля рабочего места, ограниченного дугами, описываемыми предплечьями при движении в локтевых суставах с опорой в точке локтя и с относительно неподвижным плечом.



Рисунок 9.1 − Зоны досягаемости рук в горизонтальной плоскости.

а *-* зона максимальной досягаемости;

б - зона досягаемости пальцев при вытянутой руке;

в - зона легкой досягаемости ладони;

г - оптимальное пространство для грубой ручной работы;

д *-* оптимальное пространство для тонкой ручной работы.

Оптимальное размещение предметов труда и документации в зонах досягаемости:

* дисплей размещается в зоне **а** (в центре),
* системный блок размещается в предусмотренной нише стола,
* клавиатура − в зоне **г/д,**
* «мышь» − в зоне **в** справа,
* сканер в зоне **а/б** (слева),
* принтер находится в зоне **а** (справа).

Документация, необходимая при работе − в зоне легкой досягаемости ладони – **в**, а в выдвижных ящиках стола − литература, неиспользуемая постоянно



1

2

3

5

6

4

Рисунок 9.2 −Размещение основных и периферийных составляющих ПК.

На рис. 9.2 показан пример размещения основных и периферийных составляющих ПК на рабочем столе оператора ПК.

1 – сканер, 2 – монитор, 3 – принтер, 4 – поверхность рабочего стола,

5 – клавиатура, 6 – манипулятор типа «мышь».

Для комфортной работы стол должен удовлетворять следующим условиям [26]:

* высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотник,
* нижняя часть стола должна быть сконструирована так, чтобы оператор ПК мог удобно сидеть, не был вынужден поджимать ноги,
* поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения оператора ПК,
* конструкция стола должна предусматривать наличие выдвижных ящиков (не менее 3 для хранения документации, листингов, канцелярских принадлежностей),
* высота рабочей поверхности рекомендуется в пределах 680-760мм. Высота поверхности, на которую устанавливается клавиатура, должна быть около 650мм.

Большое значение придается характеристикам рабочего кресла. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420-550мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки - регулируемый.

Необходимо предусматривать при проектировании возможность различного размещения документов: сбоку от видеотерминала, между монитором и клавиатурой и т.п. Кроме того, в случаях, когда видеотерминал имеет низкое качество изображения, например заметны мелькания, расстояние от глаз до экрана делают больше (около 700мм), чем расстояние от глаза до документа (300-450мм). Вообще при высоком качестве изображения на видеотерминале расстояние от глаз пользователя до экрана, документа и клавиатуры может быть равным.

Положение экрана определяется:

* расстоянием считывания (0,6…0,7м);
* углом считывания, направлением взгляда на 20° ниже горизонтали к центру экрана, причем экран перпендикулярен этому направлению.

Должна также предусматриваться возможность регулирования экрана:

* по высоте +3 см;
* по наклону от -10° до +20° относительно вертикали;
* в левом и правом направлениях.

Большое значение также придается правильной рабочей позе пользователя. При неудобной рабочей позе могут появиться боли в мышцах, суставах и сухожилиях. Требования к рабочей позе пользователя видеотерминала следующие:

* голова не должна быть наклонена более чем на 20°,
* плечи должны быть расслаблены,
* локти - под углом 80°…100°,
* предплечья и кисти рук − в горизонтальном положении.

Причина неправильной позы пользователей обусловлена следующими факторами: нет хорошей подставки для документов, клавиатура находится слишком высоко, а документы − низко, некуда положить руки и кисти, недостаточно пространство для ног.

В целях преодоления указанных недостатков даются общие рекомендации: лучше передвижная клавиатура; должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и экрана, а также подставка для рук [26].

Существенное значение для производительной и качественной работы на компьютере имеют размеры знаков, плотность их размещения, контраст и соотношение яркостей символов и фона экрана. Если расстояние от глаз оператора до экрана дисплея составляет 60…80 см, то высота знака должна быть не менее 3мм, оптимальное соотношение ширины и высоты знака составляет 3:4, а расстояние между знаками – 15…20% их высоты. Соотношение яркости фона экрана и символов − от 1:2 до 1:15 [22].

Во время пользования компьютером медики советуют устанавливать монитор на расстоянии 50-60 см от глаз. Специалисты также считают, что верхняя часть видеодисплея должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем когда он смотрит вниз. За счет этого площадь обзора значительно увеличивается, вызывая обезвоживание глаз. К тому же если экран установлен высоко, а глаза широко открыты, нарушается функция моргания. Это значит, что глаза не закрываются полностью, не омываются слезной жидкостью, не получают достаточного увлажнения, что приводит к их быстрой утомляемости.

Создание благоприятных условий труда и правильное эстетическое оформление рабочих мест на производстве имеет большое значение как для облегчения труда, так и для повышения его привлекательности, положительно влияющей на производительность труда.

**9.4. Режим труда**

Как уже было неоднократно отмечено, при работе с персональным компьютером очень важную роль играет соблюдение правильного режима труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках [22].

В табл. 9.5 представлены сведения о регламентированных перерывах, которые необходимо делать при работе на компьютере, в зависимости от продолжительности рабочей смены, видов и категорий трудовой деятельности с ВДТ (видеодисплейный терминал) и ПЭВМ (в соответствии с СанПиН 2.2.2 542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ» []).

**Таблица 9.5** − Время регламентированных перерывов при работе на компьютере

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Категория работы  с ВДТ или ПЭВМ | Уровень нагрузки за рабочую смену при видах работы с ВДТ | | | Суммарное время регламентированных перерывов, мин | |
| Группа А, количество знаков | Группа Б, количество знаков | Группа В, часов | При 8-часовой смене | При 12-часовой смене |
| I | до 20000 | до 15000 | до 2,0 | 30 | 70 |
| II | до 40000 | до 30000 | до 4,0 | 50 | 90 |
| III | до 60000 | до 40000 | до 6,0 | 70 | 120 |

Примечание: Время перерывов дано при соблюдении указанных Санитарных правил и норм. При несоответствии фактических условий труда требованиям Санитарных правил и норм время регламентированных перерывов следует увеличить на 30%.

В соответствии со СанПиН 2.2.2 546-96 все виды трудовой деятельности, связанные с использованием компьютера, разделяются на три группы:

группа А: работа по считыванию информации с экрана ВДТ или ПЭВМ с предварительным запросом;

группа Б: работа по вводу информации;

группа В: творческая работа в режиме диалога с ЭВМ.

Эффективность перерывов повышается при сочетании с производственной гимнастикой или организации специального помещения для отдыха персонала с удобной мягкой мебелью, аквариумом, зеленой зоной и т.п.

**9.5 Расчет освещенности**

Расчет освещенности рабочего места сводится к выбору системы освещения, определению необходимого числа светильников, их типа и размещения. Исходя из этого, рассчитаем параметры искусственного освещения.

Обычно искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые по сравнению с лампами накаливания имеют ряд существенных преимуществ [24]:

* по спектральному составу света они близки к дневному, естественному свету;
* обладают более высоким КПД (в 1,5-2раза выше, чем КПД ламп накаливания);
* обладают повышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);
* более длительный срок службы.

Расчет освещения производится для комнаты площадью 15м2 , ширина которой 5м, высота − 3 м. Воспользуемся методом светового потока [23].

Для определения количества светильников определим световой поток, падающий на поверхность по формуле:

 , где (9.1)

***F*** − рассчитываемый световой поток, Лм;

***Е*** − нормированная минимальная освещенность, Лк (определяется по таблице). Работу оператора ПК, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная освещенность будет**Е**= 300Лк;

***S*** − площадь освещаемого помещения (в нашем случае **S** = 15м2);

***Z*** − отношение средней освещенности к минимальной (обычно принимается равным 1,1…1,2 , пусть **Z** = 1,1);

***К*** − коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае ***К***= 1,5);

***n*** − коэффициент использования, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен (РС) и потолка (РП)), значение коэффициентов РС и РП были указаны выше: РС=40%, РП=60%. Значение **n** определим по таблице коэффициентов использования различных светильников. Для этого вычислим индекс помещения по формуле:

 , где (9.2)

***S*** − площадь помещения, **S** = 15 м2;

***h*** − расчетная высота подвеса, **h** = 2.92 м;

***A*** − ширина помещения, **А** = 3 м;

***В*** − длина помещения, **В** = 5 м.

Подставив значения получим:



Зная индекс помещения ***I***, по таблице 7 [23] находим ***n***= 0,22

Подставим все значения в формулу для определения светового потока ***F***:



Для освещения выбираем люминесцентные лампы типа ЛБ40-1, световой поток которых **F** = 4320 Лк.

Рассчитаем необходимое количество ламп по формуле:

 (9.3)

***N*** − определяемое число ламп;

***F*** − световой поток, **F** = 33750 Лм;

***Fл***− световой поток лампы, **Fл** = 4320 Лм.



При выборе осветительных приборов используем светильники типа ОД. Каждый светильник комплектуется двумя лампами.

**9.6 Расчет уровня шума**

Одним из неблагоприятных факторов производственной среды является высокий уровень шума, создаваемый печатными устройствами, оборудованием для кондиционирования воздуха, вентиляторами систем охлаждения в самих ЭВМ.

Для решения вопросов о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников [25]:

 (9.4)

где *Li*– уровень звукового давления i-го источника шума;n – количество источников шума.

Полученные результаты расчета сравнивается с допустимым значением уровня шума для данного рабочего места. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в табл. 9.6.

**Таблица 9.6** Уровни звукового давления различных источников.

|  |  |
| --- | --- |
| **Источник шума** | **Уровень шума, дБ** |
| Жесткий диск | 40 |
| Вентилятор | 45 |
| Монитор | 17 |
| Клавиатура | 10 |
| Принтер | 45 |
| Сканер | 42 |

Обычно рабочее место оператора оснащено следующим оборудованием: винчестер в системном блоке, вентилятор(ы) систем охлаждения ПК, монитор, клавиатура, принтер и сканер.

Подставив значения уровня звукового давления для каждого вида оборудования в формулу, получим:

L∑=10·lg(104+104,5+101,7+101+104,5+104,2)=49,5 дБ

Полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ (ГОСТ 12.1.003-83). И если учесть, что вряд ли такие периферийные устройства как сканер и принтер будут использоваться одновременно, то эта цифра будет еще ниже. Кроме того при работе принтера непосредственное присутствие оператора необязательно, т.к. принтер снабжен механизмом автоподачи листов.

**9.7. Пожарная безопасность**

**9.7.1. Содержание помещений**

Во всех помещениях после окончания работы электроосвещение, множительная и оргтехника должна быть выключена, а электросеть обесточена.

Все окна в помещениях по окончании рабочего дня должны быть плотно закрыты, двери заперты и опечатаны.

Производить ежедневный противопожарный осмотр помещения после окончания рабочего дня, осмотр регистрировать в специальном журнале с росписью ответственного лица.

Запасной комплект ключей от запасных выходов должен находиться у дежурного поста.

В коридорах на доступных местах должны быть вывешены планы эвакуации и стрелки-указатели запасных выходов для эвакуации людей в случае пожара.

Запрещается устанавливать в коридорах, на лестничных клетках, в лифтовых холлах, на выходах из помещений какие-либо предметы, препятствующие движению людей, сужающие расчетную ширину эвакуационных проходов и выходов.

В служебных помещениях возле выхода из помещения должна находиться памятка о мерах противопожарной безопасности.

**9.7.2. Содержание средств пожаротушения**

Запрещается использовать средства пожаротушения для хозяйственных нужд.

Все средства пожаротушения должны постоянно содержаться в исправном состоянии, в полной комплектации и годности к действию для ликвидации пожара.

Системы оповещения о пожаре должны находиться в исправном состоянии и обеспечивать передачу сигнала оповещения о пожаре одновременно по всему зданию.

**9.7.3. Подготовка и действия сотрудников при возникновении пожара**

Все сотрудники должны знать места расположения средств пожаротушения, правильно и эффективно их применять, проходить инструктаж и периодически проходить обучение приемам пользования средствами пожаротушения.

При обнаружении пожара каждый сотрудник обязан:

Немедленно сообщить об этом в городскую пожарную охрану по телефону 01 (при этом необходимо назвать адрес объекта, место возникновения пожара, а также сообщить свою фамилию);

Принять меры по вызову к месту пожара руководящего должностного лица.

Приступить к тушению очага пожара имеющимися средствами пожаротушения.

Принять по возможности меры по эвакуации людей, тушению пожара и сохранности материальных ценностей.

Воздержаться от открытия окон и дверей, а также не разбивать стекол. Покидая помещение, необходимо закрыть окна и двери, чтоб поток свежего воздуха не способствовал быстрому распространению огня.

Эвакуацию проводить без паники, не забывать про верхнюю одежду в холодное время года, не образовывать пробки на путях эвакуации, эвакуировать материальные ценности в соответствии с разработанным планом эвакуации.

Лица, виновные в нарушении Инструкции, в зависимости от характера нарушений и их последствий, несут ответственность в дисциплинарном, административном или уголовном порядке.

**Заключение**

Результатом выпускной квалификационной работы является интернет-сервис «Налоговый помощник».

В ходе выполнения дипломной работы был проведено обследование объекта автоматизации. По результатам анализа было выявлено, что налогоплательщик тратит свои временные ресурсы для получения информации о налоговых ставках и льготах. Для налоговых сотрудников имеется проблема контролирования актуальной информации о налоговых ставах и льготах, так как информация рассредоточена по налоговым инспекциям округа. В качестве решения было принято разработать интернет-сервис «Налоговый помощник».

При принятии решения о выборе инструментов разработки выбор сделан в пользу каркасных систем (фреймворков), которые позволяли использовать опыт наработок в сфере web-приложений и ускоряли процесс создания приложений.

При обзоре аналогов были рассмотрены такие системы как Zend Framework, CodeIgniter, Yii и CakePHP. При анализе каждой системы были выявлены особенности и возможности каждой каркасной системы, которые были использованы при оценке значимых показателей для разработки интернет-сервиса. В результате сравнения суммарных оценок показателей был выбран CodeIgniter, так имеет высокую производительность, быструю возможность инсталляции и минимальной настройки, широкие начальные возможности, подробную документацию и большое русское сообщество, что явилось определяющим фактором выбора данного фреймворка.

Фреймворк CodeIgniter реализует шаблон проектирования Модель-Представление-Контроллер, который разделяет приложение на функциональные слои, что позволяет обеспечить независимость одного слоя от другого.

Для слоя домена используется типовой шаблон проектирования модель предметной области. Модель предметной области образует сеть взаимосвязанных объектов, в которой каждый объект представляет собой отдельную значащую сущность. В CodeIgniter слой источника данных реализует шаблон архитектуры источников данных активная запись.

В слое служб платформой CodeIgniter используется шаблон проектирования контроллер приложения. Вместе с ним для слоя представления был выбран паттерн представления по шаблону.

Следовательно, интернет-сервис решил задачи совершенствования администрирования имущественных налогов и повышения эффективности информирования налогоплательщиков по применению налоговых ставок и льгот.

**Список литературы**

1. Административный регламент №9 федеральной налоговой службы по исполнению государственной функции по бесплатному информированию (в том числе в письменной форме) налогоплательщиков, плательщиков сборов и налоговых агентов о действующих налогах и сборах, законодательстве о налогах и сборах и принятых в соответствии с ним нормативных правовых актах, порядке исчисления и уплаты налогов и сборов, правах и обязанностях налогоплательщиков, плательщиков сборов и налоговых агентов, полномочиях налоговых органов и их должностных лиц, а также представлению форм налоговых деклараций (расчетов) и разъяснению порядка их заполнения.
2. Лебедев К. С. Разработка метода и инструментальных средств создания приложений для системы управления веб-сайтами: Автореферат диссертации на соискание ученой степени кандидата технических наук.
3. Что такое CMF, CMS. Сравнительный анализ популярных CMS для разработки сайтов: [Электронный ресурс]: Максимальный эффект, − Режим доступа: http://www.maxeff.org/node/53.
4. CodeIgniter: [Электронный ресурс]: Википедия – свободная энциклопедия, - Режим доступа: http://ru.wikipedia.org/wiki/CodeIgniter/.
5. Zend Framework: [Электронный ресурс]: Русскоязычное Zend Framework сообщество, - Режим доступа: http://zendframework.ru/.
6. Полное руководство по Yii – Что такое Yii: [Электронный ресурс]: Русскоязычное сообщество Yii, - Режим доступа: http://zendframework.ru/.
7. Возможности Yii Framework: [Электронный ресурс]: Yii Framework − Russia, − Режим доступа: http://yii.devgroup.ru/tiki-index.php?page= Возможности+Yii+Framework.
8. CakePHP: [Электронный ресурс]: Википедия – свободная энциклопедия, - Режим доступа: http://ru.wikipedia.org/wiki/CakePHP/.
9. IEEE 830-1993. Спецификация требований к ПО (Software Requirements Specification – SRS).

**Приложение А**

**Структура УФНС по ХМАО – Югре**

**Приложение Б**

**Структурно-функциональная модель системы «AS-IS» (IDEF0)**

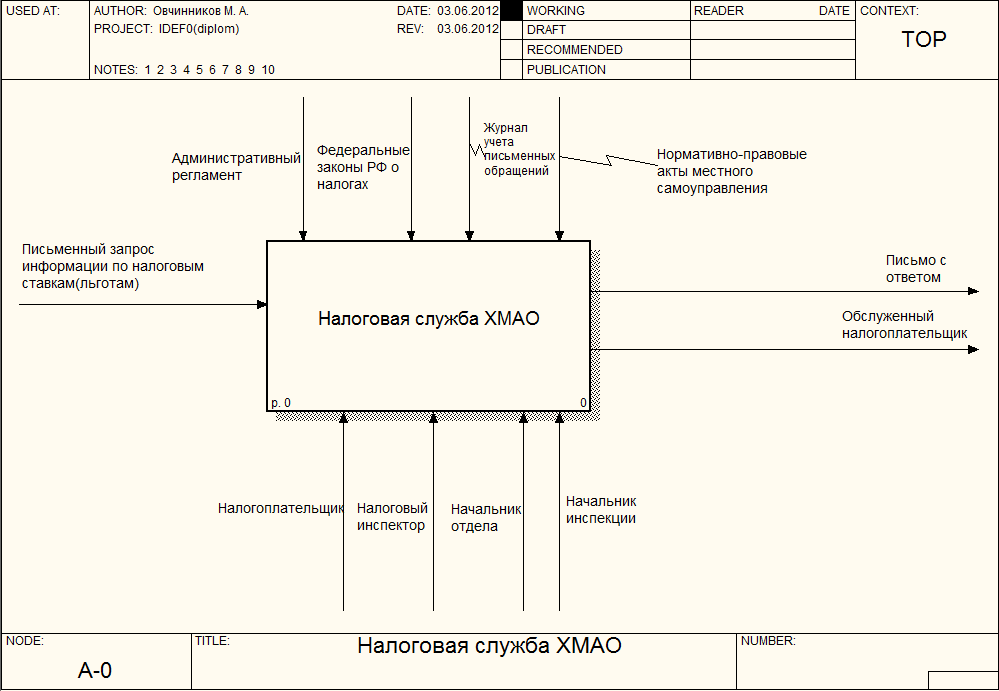


Рисунок Б.1 − IDEF0 – диаграмма «AS-IS»(контекстная диаграмма)

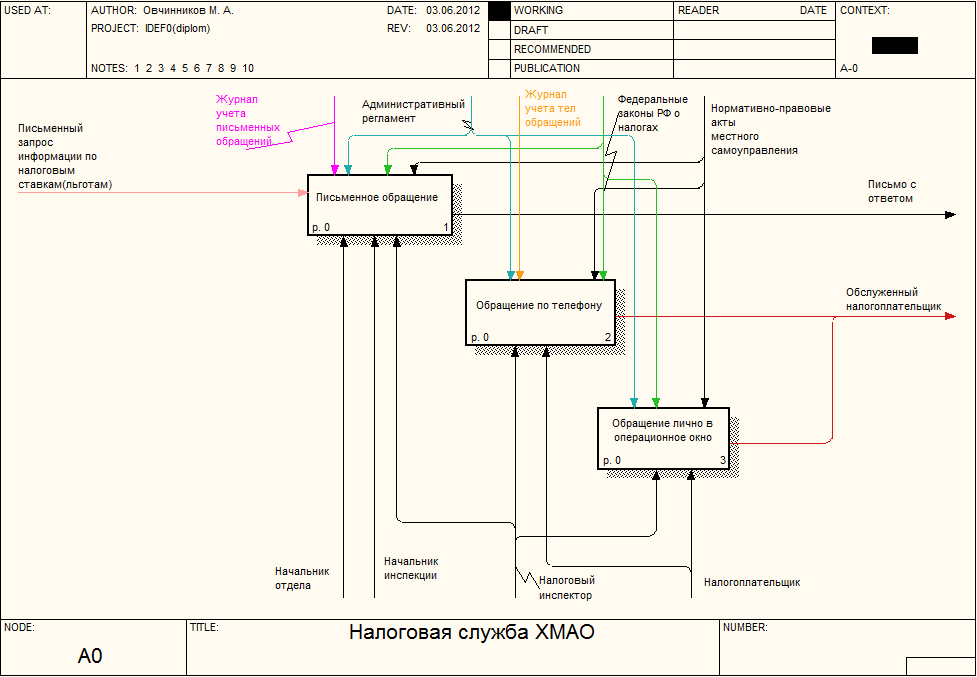


Рисунок Б.2 - IDEF0 – диаграмма «AS-IS»(декомпозиция первого уровня)

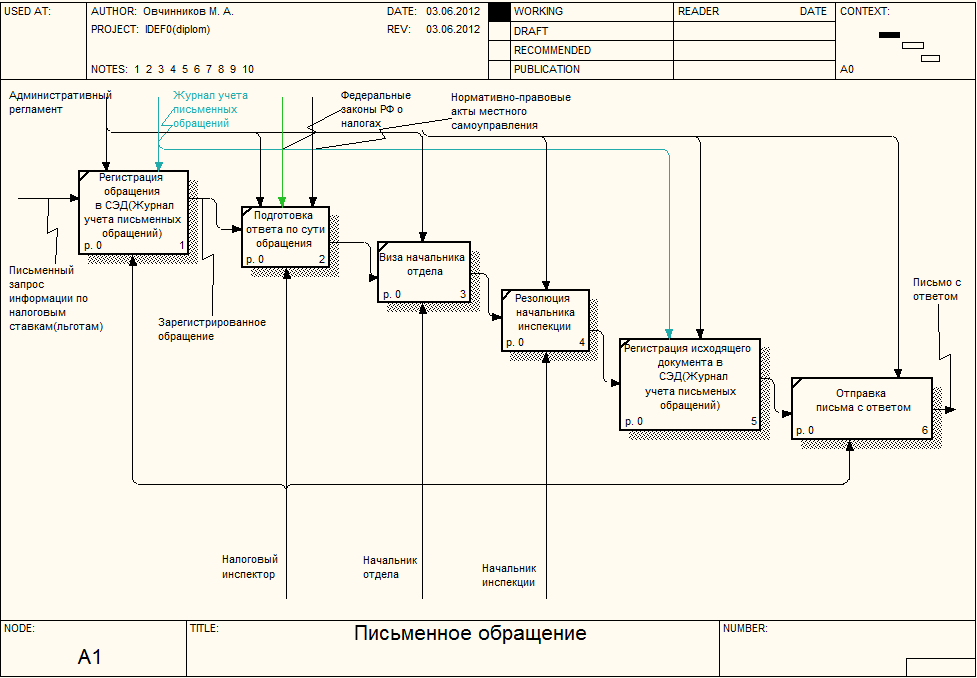


Рисунок Б.3 - IDEF0 – диаграмма «AS-IS»(декомпозиция блока «Письменное обращение»)

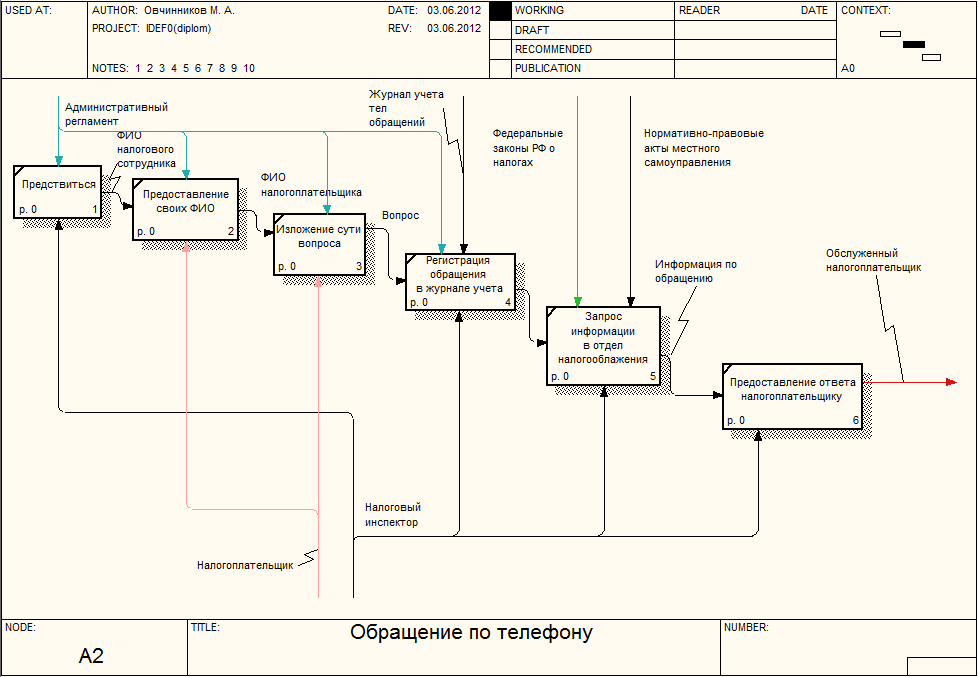


Рисунок Б.4 - IDEF0 – диаграмма «AS-IS»(декомпозиция блока «Обращение по телефону»)

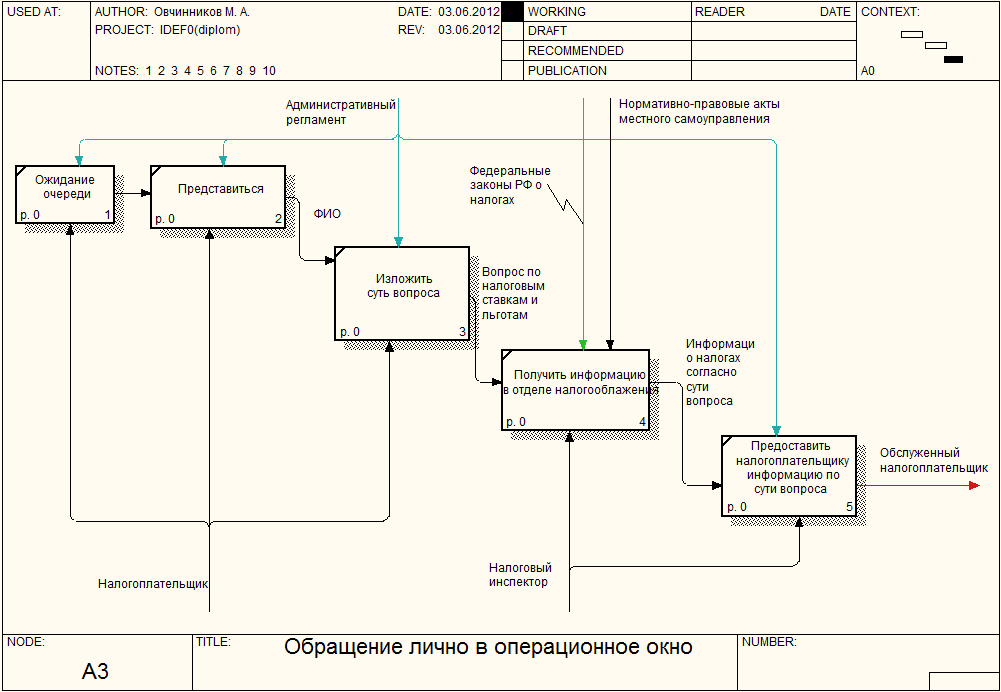


Рисунок Б.4 - IDEF0 – диаграмма «AS-IS»(декомпозиция блока «Обращение лично в операционное окно»)

**Приложение В**

**Диаграмма потоков данных (DFD) – TO-BE**

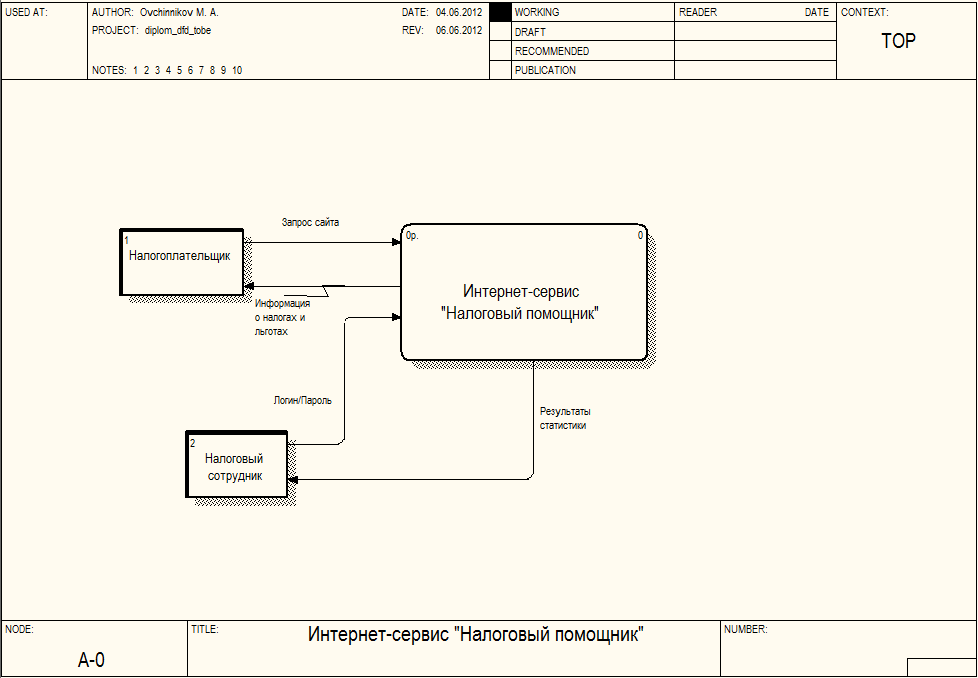


Рисунок В.1 – К

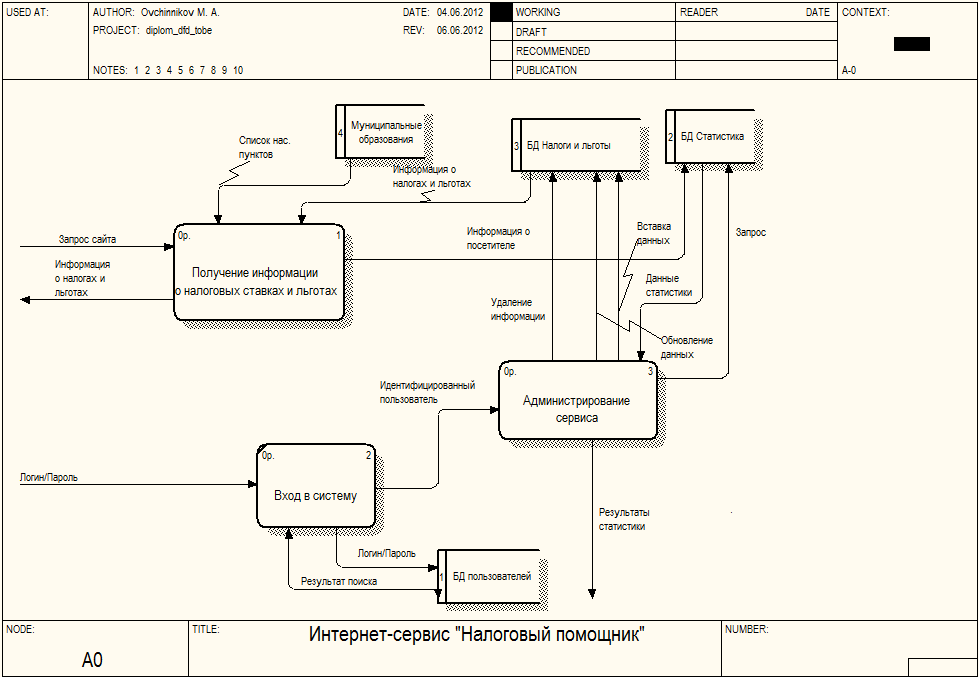


Рисунок В.2 −

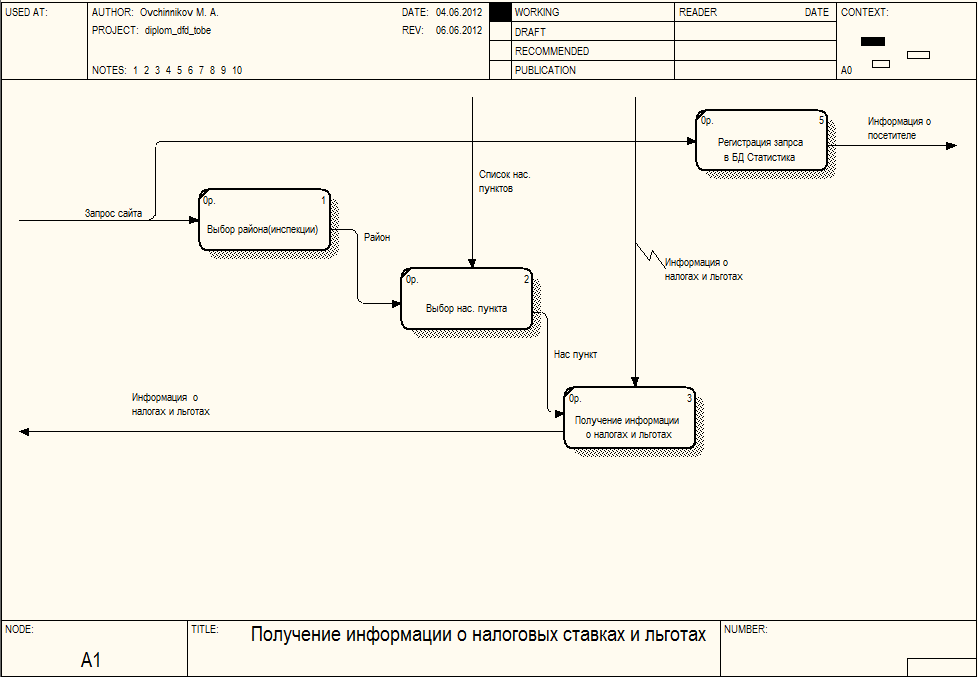


Рисунок В.3 –

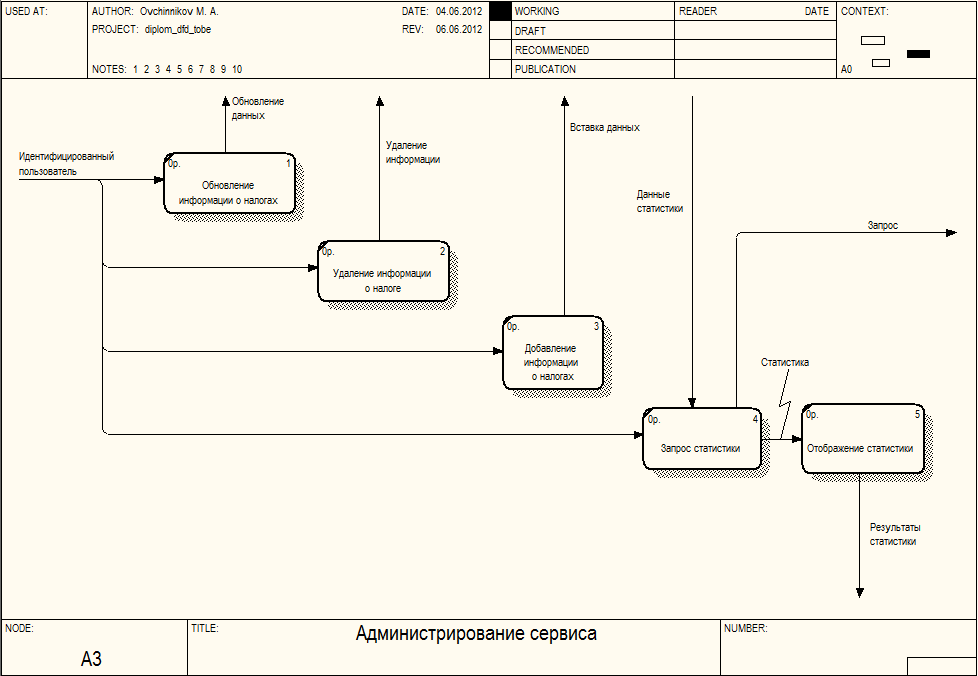


Рисунок В.4 −

**Приложение Г**

**Техническое задание на создание интернет-сервиса «Налоговый помощник»**

* 1. Общие сведения

Данное техническое задание описывает требования и порядок создания информационного портала «Налоговый помощник», в соответствии с которым проводится разработка портала и ее приемка при вводе в эксплуатацию. ТЗ включает в себя системное описание расширенных требований к разрабатываемому изделию.

Заказчиком системы является УФНС России по ХМАО-ЮГРЕ.

Система создается на основании обследования предприятия при прохождении преддипломной практики и задания на дипломный проект.

Сроки проведения работ: январь 2012 г. – июнь 2012 г.

2. Назначение и цели создания информационного портала

2.1. Назначение системы:

1. Предоставление информации о налоговых ставках по месту расположения налогоплательщика

2. Предоставление информации о налоговых льготах по месту расположения налогоплательщика

3. Предоставление информации о порядке уплаты налогов

4. Предоставление нормативных документов, форм налоговой отчетности, форм квитанций об уплате налогов посредством возможности скачивания файлов

5. Доступ к предоставляемым интернет-сервисам

6. Предоставление информации о работе межрайонных инспекциях ФНС России в ХМАО

7. Возможность отправки вопроса в налоговые органы ХМАО

8. Администрирование информационного портала и разграничение прав между межрайонными инспекциями

2.2. Цели создание системы:

1) Выполнение направления стратегии развития ФНС России как клиент-ориентированной организации в рамках всероссийского проекта «Электронная Россия»

2) Предоставление максимально возможной информации для физических лиц в доступной форме

3) Разгрузка операторов налоговых органов

3. Характеристика объекта автоматизации

Объектом автоматизации является деятельность по предоставлению информации на.

4. Требования к системе

4.1 Требования к системе в целом

* + 1. Требования к структуре и функционированию системы

1. Бесперебойная работа информационного портала 24 часа;
2. Информационный портал имеет 2 части:

* Пользовательская часть: получение информации, отправка вопросов, скачивание файлов;
* Администраторская часть: редактирование информации, загрузка файлов, просмотр статистики посещения портала;

1. Система должна быть расширяемой (возможность подключения дополнительных модулей)

4) Обеспечение эффективности работы конечного пользователя портал в виде использования средств навигации, меню, онлайновых подсказок и всплывающих окон.

5) Система должна легко инсталлироваться (пошаговая инсталляция с предоставлением инструкций на каждом шаге и вводом параметров установки)

6) Обеспечение легкости эксплуатации информационного портала (Наличие автоматизированных процедур запуска, копирования и восстановления работоспособности приложения)

* + 1. Требования к численности и квалификации персонала системы и режиму его работы

1. Пользователь – базовые знания ПК, умение пользоваться любым интернет-браузером, знание правил использования системы.
2. Сотрудник - углубленные знания ПК, умение пользоваться любым интернет-браузером, знание инструкции по эксплуатации системы.
3. Администратор – опыт администрирования СУБД MySQL; знание языков SQL, HTML, CSS, Ajax; опыт программирования на PHP; понимание реализации системы в виде шаблона проектирования Model-View-Controller и способность применять ее в реальном проекте; умение построения систем на основе PHP-фреймворков; понимание и умение применять объектно-ориентированную модель; умение инсталляции и настройки веб-серверов.
   * 1. Показатели назначения

Должно достигаться изменение следующих показателей:

1. количество хранимых записей в таблицах базы данных (до 10-12 тыс.);
2. формат вывода отчетов.
   1. Требования к функциям (задачам), выполняемым системой
      1. Перечень функций, задач, подлежащих автоматизации
3. отправка обращений в налоговые органы;
4. функции администрирования портала;
5. мониторинг посещения статистики сайта;
6. отображение информации из БД.
   * 1. Требования к форме представления выходной информации
7. выходная информация представляется в виде таблиц представленных в виде html-тегов;
8. таблицы должны иметь возможность экспортироваться в документ Word;
   * 1. Перечень отказов системы
        1. Сбои в работе используемой СУБД;
        2. Неисправность в аппаратных средствах;

4.3. Требования к видам обеспечения

4.3.1. Требования к информационному обеспечению

4.3.1.1. Все данные организованы в таблицы и хранятся в базе данных. В качестве базы данных для хранения используется MySQL. СУБД выступает инструмент PhpMyAdmin.

4.3.1.2. В информационное обеспечение должны быть включены физическая и логическая модель данных.

4.3.2. Требования к лингвистическому обеспечению

4.3.2.1. Язык разметки документов HTML5 (определяет структуру отображения веб-страниц)

4.3.2.2. Язык описания внешнего вида документа CSS3 (задает цвет, шрифт, расположение отдельных блоков и других аспектов представления)

4.3.2.3. Java-script - объектно-ориентированный скриптовый язык программирования (Создание интерактивности веб-страниц)

4.3.2.4. AJAX - подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером (обновление данных происходит без перезагрузки страниц, что делает веб-приложение быстрым и удобным)

4.3.2.5. PHP (*PHP: Hypertext Preprocessor*) - скриптовый языкпрограммирования общего назначения, интенсивно применяемый для разработки веб-приложений (обеспечивает динамическое содержание сайта).

4.3.2.6. SQL – cструктурированный язык манипулирования данными.

4.3.2.7. Библиотеки, плагины, модули, готовые инструменты (обеспечение принципа повторяемости):

* PHP-фреймворк CodeIgniter (фреймворк - структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта)
* jOuery - библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API по работе с Ajax.
* HTML5 Boilerplate - базовый HTML/CSS/JS-шаблон, который позволяет полноценно использовать последние технологии, не обращая внимания на различия в браузерах. Шаблоны содержат: кросс-браузерную нормализацию, оптимизацию быстродействия, кроссбраузерный Ajax, .htaccess файл для кэширования статического контента и gzip'а и т.п.
* CSS-фреймворк 960 GridSystem (подготовленная css-библиотека, созданная для упрощения работы верстальщика, быстроты разработки и исключения максимально возможного числа ошибок вёрстки)
  1. Требования к программному обеспечению

1. информационная система должна функционировать под операционными системами семейства Windows, Unix подобных систем;
2. в качестве платформы размещения информационной системы выступает связка IIS-PHP-MSSQL.
3. Клиентам для доступа к информационной системе является веб-браузер (Google Chrome 10.0+, Opera 6.0+, Mozilla Firefox 3.5+, Internet Explorer 6+)
   * 1. Требования к техническому обеспечению

Рабочее место пользователя (Минимальные требования):

* клавиатура, мышь;
* монитор;
* материнская плата;
* жесткий диск;
* процессор AMD/Intel с тактовой частотой не менее 2000 МГц;
* видеокарта с объемом видеопамяти не менее 32 Мб;
* оперативная память не менее 512 Мб;
* сетевая плата с выходом в интернет.

1. Состав и содержание работ по содержанию системы
   1. Перечень стадий работ по созданию системы
      1. формирование требований к АС;
      2. разработка технического задания;
      3. разработка концепции АС;
      4. создание детального проекта.
   2. Сроки выполнения

Сроки выполнения январь 2012 г. – июнь 2012 г.

1. Порядок контроля и приемки системы

Промежуточный контроль над ходом проектирования ИС осуществляется научным руководителем проекта еженедельно на протяжении всего периода разработки системы.

Прием проекта будет осуществлена государственной аттестационной комиссией.

1. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.

Не обозначены

1. Требования к документированию

Перечень подлежащих разработке комплектов и видов документов:

1. техническое задание;
2. проектная документация программного обеспечения.
3. Источники разработки

Документы и информационные материалы, на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании системы:

1. ГОСТ 34.602-89 «Техническое задание на создание АС»;
2. задание на дипломный проект;
3. проектная документация программного обеспечения.

**Приложение Д**

**Диаграммы последовательностей детального проектирования**

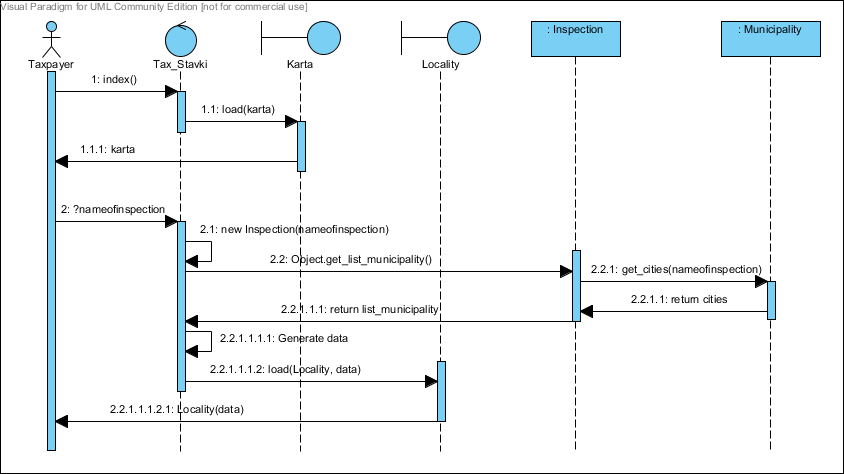
****

Рисунок Д.1 – Диаграмма последовательности выбора муниципального образования

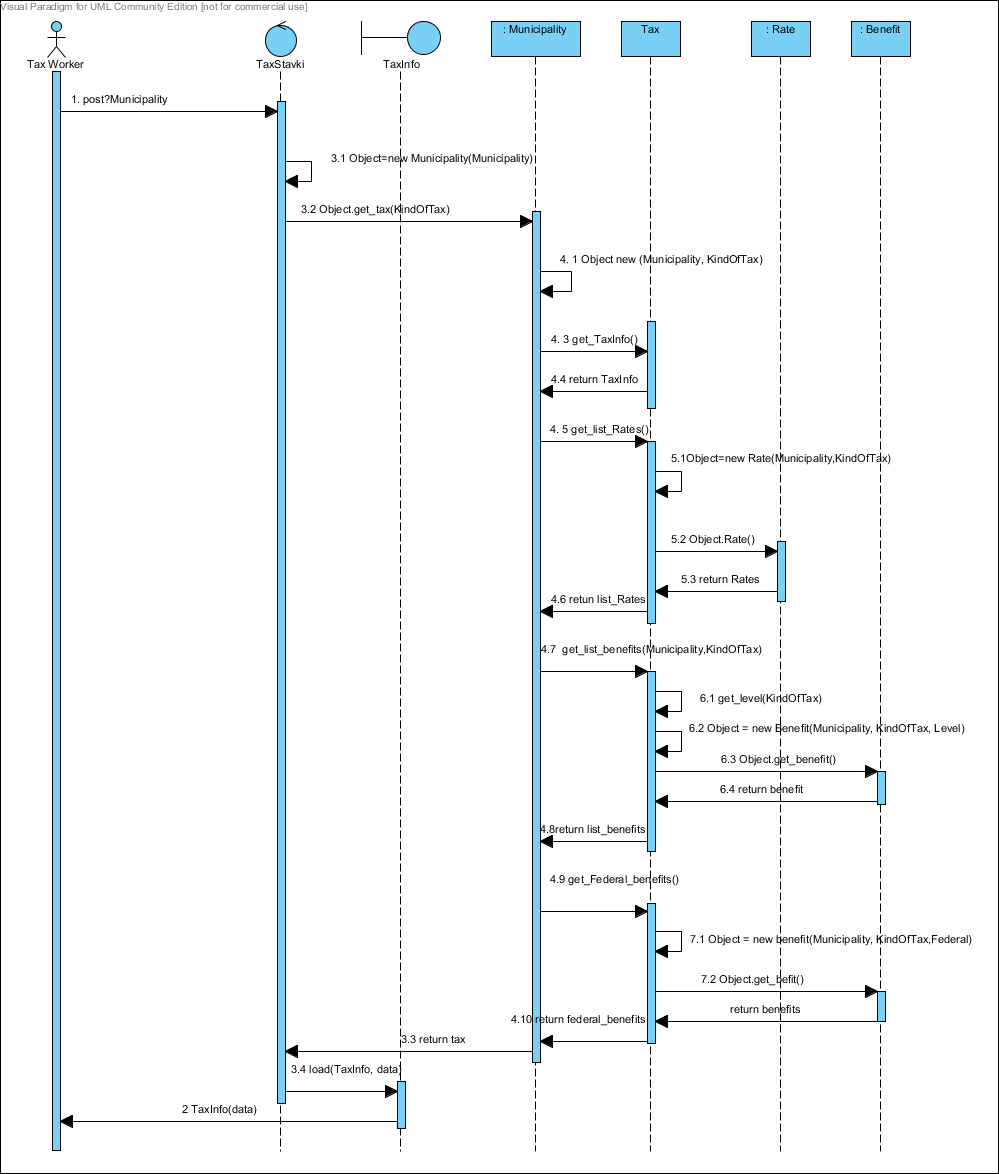
****

Рисунок Д.2 – Диаграмма последовательности варианта использования «Просмотр информации о налоговых ставках»

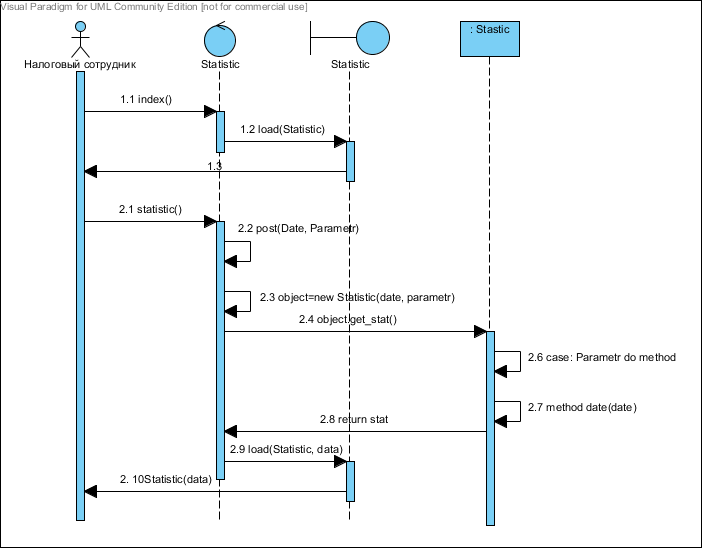
****

Рисунок Д.3 − Диаграмма последовательности варианта использования «Просмотр статистики»