

## Содержание

Введение	2
1 ОБЩАЯ ЧАСТЬ	5
1.1 Анализ предметной области	5
1.2 Средства и среда разработки	7
2 СПЕЦИАЛЬНАЯ ЧАСТЬ	14
2.1 Описание требований к информационной системе	14
2.2 Диаграммы вариантов использования	21
2.3 Диаграммы состояний	23
2.4 Схемы данных	24
2.5 Прототип спецификации API	25
3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ	37
3.1 Расчет затрат на разработку программы и решение задачи на ЭВМ	37
3.2 Расчет экономического эффекта и определение срока окупаемости	42
3.3 Экономический эффект и срок окупаемости	43
Заключение	45
Список использованной литературы	47
Приложение А	50
Приложение Б	127

					ДП.22.090207.482.06.ПЗ			
Изм	Лист	№ докум.	Подпись	Дата				
Разраб.		Кулманаков И.В.			Пояснительная записка	Лит.	Лист	Листов
Пров.						Т	2	135
						ТТИТ 482 гр.		
Н. контр.								
Утв.								

## Введение

Информационные технологии все больше и больше затрагивают сферы деятельности человека. И сейчас, под натиском информационных и телекоммуникационных технологий необходимо введение информационных систем в те области, где они не применяются или слабо развиты и которые могут уменьшить затраты, время на обработку данных и увеличить производительность труда.

Томсксофт занимается разработкой программного обеспечения.

Самые крупные проекты связаны с передачей и обработкой мультимедиа информации, формированием соцсетей для общения между пользователями этих проектов и технологическими инструментами для обработки мультимедиа. Каждый из таких проектов содержит несколько компонентов, включая, но не ограничиваясь:

1. Приложения для десктопных компьютеров
2. ВЕБ-сайты и сервисы
3. Сетевые сервисы, обслуживающие до нескольких сотен тысяч он-лайн пользователей одновременно
4. Приложения для мобильных ОС
5. Системы мониторинга и защиты от сетевых атак

Компания состоит из следующих отделов:

1. Администрация
2. Отдел сетевых разработок
3. Отдел разработки приложений

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		3

4. Отдел системного администрирования и мониторинга
5. Multimedia development department
6. Отдел дизайна
7. Отдел качества ПО и документирования
8. Рабочие группы по проектам

Каждая команда раз в неделю собирается в Zoom конференции, с участием представителя руководства компании для разбора сделанных за прошедшую неделю задач, решённых и нерешенных проблем, планирования действий на неделю. Остальные организационные принципы устанавливаются конкретной рабочей группой самостоятельно.

Объект и предмет проекта:

Объект: Информационное взаимодействие внутренних сервисов компании Томсксофт с единой внутрикорпоративной системой.

Предмет: Интеграция взаимодействия внутренних сервисов компании Томсксофт с единой внутрикорпоративной системой.

Цель проекта:

Разработка информационной системы для организации канала связи внутренних сервисов компании Томсксофт с внутрикорпоративной системой.

Задачи:

1. Изучить особенности взаимодействия сервисов с внутрикорпоративной системой.
2. Проанализировать возможности автоматизации данного процесса
3. Рассмотреть существующие варианты программных продуктов, автоматизирующие данный процесс

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		4

4. Определить основные технические и функциональные требования к разрабатываемой системе
5. Выполнить работу по проектированию ИС с учётом заявленных требований
6. Разработать спроектированную информационную систему
7. Разработать документацию к API

Практическая значимость и ожидаемые результаты

В результате выполнения проекта, будет разработана информационная система, позволяющая сторонним сервисам обращаться к системе для автоматизации комплекса используемых компании сервисов.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		5

## 1 ОБЩАЯ ЧАСТЬ

### 1.1 Анализ предметной области

В компании Томсксофт имеется множество сотрудников. Для учёта рабочего процесса и времени которых применяется внутренняя intranet-система.

Данная система разработана для

1. Размещения внутрикорпоративной информации
2. Автоматизированной рассылки данных
3. Оповещения сотрудников посредством электронной почты
4. Размещение файлов
5. Хранение учётных данных внешних корпоративных сервисов (логины и пароли)
6. Предупреждение коллег о изменении рабочего времени
7. Ведение информации о проектах
8. Ведение информации о запланированной и проделанной работе

Также система содержит в себе вывод простых отчётов на основе фильтров на web-страницу, после чего пользователь может использовать полученные данные для своих нужд:

1. Бухгалтер для расчёта зарплаты
2. Администрация для учёта рабочего времени и отпусков
3. Руководители проектов для контроля прогресса разработки

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		6

#### 4. И многие другие

С развитием сторонних систем надобность в ручном расчёте постепенно пропадает и всю работу пользователей составляет рутинный перенос данных из web-страницы в сторонний сервис.

К сторонним сервисам можно отнести:

1. 1С для расчёта заработной платы бухгалтерией
2. 1С для расчёта отпусков администрацией
3. Почтовая служба для рассылок писем
1. Планируемая служба для автоматической генерации записей и проверки рабочего времени на основе внешних источников
2. И многие другие

Исходя из всего вышеперечисленного было принято решение создать систему для интеграции данных компании со сторонними сервисами с целью автоматизации обработки и передачи данных.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		7

## 1.2 Средства и среда разработки

### 1.2.1 Теоретическая часть

Прежде чем приступить к рассмотрению средств разработки, которые были применены для создания программы, необходимо определиться с основными понятиями и терминами.

Разработка программ — сложный процесс, основной целью которого является создание, сопровождение программного кода, обеспечивающего необходимый уровень надёжности и качества. Для достижения основной цели разработки программ используются средства разработки программного обеспечения.

Средства разработки программного обеспечения — совокупность приёмов, методов, методик, а также набор инструментальных программ (компиляторы, прикладные/системные библиотеки и т.д.), используемых разработчиком для создания программного кода программы, отвечающего заданным требованиям.

Основные средства, используемые на этапах разработки программы:

- Проектирование приложения;
- Реализация программного кода приложения;
- Тестирование приложения.

Проектирование приложения — это процесс создания проекта программного обеспечения (ПО), целью которого является определение внутренних свойств системы и детализации её внешних (видимых) свойств

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		8

на основе выданных заказчиком требований к ПО. Эти требования подвергаются анализу. Средство содержит в себе наличие технического задания, эскизного и технического проекта и только потом рабочего проекта.

Реализация программного кода приложения — это процесс создания кода компонентов программного обеспечения на выбранном языке программирования. При процедурном подходе реализация заключается в программировании функций и файлов (модулей).

Объектно-ориентированное программирование (ООП) — методология программирования, совокупности основанная объектов, на каждый представлении из которых программы является в виде экземпляром определённого класса, а классы образуют иерархию наследования.

Интегрированная среда разработки программного обеспечения— система программных средств, используемая для разработки программного обеспечения.

Обычно среда разработки включает в себя текстовый редактор, компилятор и/или интерпретатор, средства автоматизации сборки и отладчик. Иногда также содержит средства для интеграции с системами управления версиями и разнообразные инструменты для упрощения разработки приложения.

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (ЭВМ) под её управлением.

Фреймворк — это программный продукт, который упрощает создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули, а все



специфичные для проекта компоненты реализуются разработчиком на их основе. Тем самым достигается не только высокая скорость разработки, но и большая производительность и надёжность решений.

СУБД (Система управления базами данных) — комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными (вставлять, обновлять, удалять и выбирать). Система предоставляет средства для администрирования БД.

Информационная система — система, предназначенная для хранения, поиска, обработки информации и соответствующие организационные ресурсы.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		10

### 1.2.2 Выбор языка программирования

При разработке информационной системы был использован такой язык программирования, как PHP.

PHP (рекурсивный акроним словосочетания PHP: Hypertext Preprocessor) - это распространённый язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в тело ответа.

PHP крайне прост для освоения, но вместе с тем способен удовлетворить запросы профессиональных программистов.

Главная область применения PHP - написание скриптов, работающих на стороне сервера; таким образом, PHP способен обрабатывать данные форм, генерировать динамические страницы или отсылать и принимать cookies. Но PHP способен выполнять намного больше.

Существуют три основных области применения PHP:

1. Создание скриптов для выполнения на стороне сервера.
2. Создание скриптов для выполнения в командной строке.
3. Создание оконных приложений, выполняющихся на стороне клиента.

PHP доступен для большинства операционных систем, включая Linux, многие модификации Unix (такие как HP-UX, Solaris и OpenBSD), Microsoft Windows, macOS, RISC OS и многие другие. Также в PHP включена поддержка большинства современных веб-серверов, таких как Apache, IIS и многих других.

### 1.3 Среда разработки

Для реализации кода была использована такая среда разработки, как Eclipse IDE.

Преимущества среды разработки:

1. Простая в установке и использовании.
2. Настраиваемый графический интерфейс.
3. Интуитивно понятный интерфейс.
4. Работа с множеством проектов.
5. Как программное обеспечение с открытым исходным кодом доступна бесплатно (в отличие от конкурентов, таких как IntelliJ IDEA).
6. Предоставленные функции, утилиты и автодополнение облегчают написание кода.
7. Ускоряет разработку приложений и повышает эффективность работы команды программистов.
8. Благодаря разнообразию плагинов тонко настраивается и расширяется дополнительными функциями.
9. Доступна для любой платформы.
10. Из-за многолетнего существования среды в сети находится многочисленная документация, советов и хитростей по использованию.
11. Большое сообщество разработчиков помогает с решением вопросов и проблем на форумах.

12.Предоставляет встроенные локальные серверы, где разработчики развёртывают и тестируют приложения, прежде чем отправлять в другие среды.

13.Сохраняет и восстанавливает сессии.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		13

### 1.3.1 Выбор фреймворка

При разработке информационной системы был использован фреймворк Laravel.

Laravel — бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC (англ. Model View Controller — модель-представление-контроллер). Laravel выпущен под лицензией MIT.

Фреймворк отлично документирован: документация есть ко всему и на нескольких языках, в том числе на русском. Также каждый метод имеет шапку в PHPDoc. Помимо документации, есть множество руководств и форумов, помогающих разобраться с возникшими проблемами.

Можно выделить следующие плюсы:

1. Установка с помощью Composer
2. Высокая производительность
3. Встроенная валидация форм
4. Возможность подключения сторонних библиотек
5. Использование миграций баз данных
6. Поддержка автоматического модульного и интеграционного тестирования
7. CLI для автоматизации разработки

### 1.3.2 Библиотеки

Для разработки информационной системы использовались библиотеки Laravel Sanctum для авторизации и Eloquent power joins для построения запросов к БД.

Laravel Sanctum предлагает легковесную систему аутентификации для SPA (одностраничных приложений), мобильных приложений и простых API на основе токенов. Sanctum позволяет каждому пользователю вашего приложения создавать несколько токенов API для своей учетной записи. Этим токенам могут быть предоставлены полномочия / области, которые определяют, какие действия токенам разрешено выполнять.

Для запросов к БД в дополнение к Eloquent ORM использовался Eloquent power joins, являющийся строителем join-запросов для моделей, основанных на Eloquent.

В данном подходе можно выделить следующие плюсы:

1. Решение проблемы N+1: когда для одной записи данных требуется дополнительно загрузить ещё N записей.
2. Синтаксис методов предоставляет простой способ создания join запросов, которые не реализованы в Eloquent. Данные запросы реализованы в строителе запросов, что использует Eloquent, но слишком громоздки для использования.

## 2 СПЕЦИАЛЬНАЯ ЧАСТЬ

### 2.1 Описание требований к информационной системе

#### 2.1.1 Описание функций неавторизованного пользователя

##### Авторизация

##### Входные данные:

1. Ник
2. Пароль

##### Выходные данные:

1. Токен для взаимодействия с API

##### Описание поведения:

1. Считывает данные, отправленные в запросе
2. Валидирует данные. Если имеются не верно заполненные поля, то возвращается HTTP код ошибки и предупреждающее сообщение в теле ответа.
3. Если пользователь не смог войти, то возвращается HTTP код ошибки и предупреждающее сообщение в теле ответа.
4. Если пользователь успешно вошёл, то выдаётся токен пользователя для дальнейшего взаимодействия с API.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		16

## Описание функций авторизованного пользователя

Для авторизации необходимо предоставить токен в заголовке X-Auth-Key.

## Создание токенов для внешних сервисов

Входные данные:

1. Название токена
2. Предоставляемые права

Выходные данные:

1. Токен

Описание поведения:

1. Создание токена с заданными именем и правами.
2. Отправка токена пользователю.

## Просмотр созданных токенов пользователя

Входные данные: —.

Выходные данные:

1. Список имён токенов и их прав

Описание поведения:

1. Выборка токенов пользователя по данным аутентификации.
2. Отправка пользоателю.

## Просмотр токена

Входные данные:

1. Название токена

Выходные данные:

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		17



1. Имя токена
2. Права
3. Дата и время изменения
4. Дата и время последнего использования

Описание поведения:

1. Выборка токена пользователя по его имени.
2. Отправка пользователю.

Редактирование токена

Входные данные:

1. Старое название токена
2. Новое название токена
3. Список прав токена

Выходные данные:

1. Имя токена
2. Права
3. Дата и время последнего использования

Описание поведения:

1. Выборка токена пользователя по его имени.
2. Изменение токена.
3. Если пользователь не обладает правами, которые хочет назначить токену, то эти права игнорируются и выдаются соответствующие сообщения.
4. Отправка ответа пользователю.

Удаление токена

Входные данные:

1. Название токена

Выходные данные: —.

Описание поведения:

1. Выборка токена пользователя по его имени.
2. Удаление токена.
3. Отправка ответа пользователю.

Отчёт по часам

Входные данные:

1. Дата начала отчётного периода
2. Дата конца отчётного периода

Выходные данные — список, где каждый элемент содержит:

1. Ник сотрудника
2. ФИО сотрудника
3. Департамент
4. Сумма отработанных часов за заданный период

При этом один сотрудник входит в список только один раз.

Описание поведения:

1. Выборка данных для отчёта
2. Отправка результата.

Отчёт по часам, сгруппированный по проектам

Входные данные:

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		19

1. Дата начала отчётного периода

2. Дата конца отчётного периода

Выходные данные — список, где каждый элемент содержит:

1. Ник сотрудника

2. ФИО сотрудника

3. Департамент

4. Данные проекта (id и название)

5. Данные заказчика (id и название)

6. Сумма отработанных часов за заданный период

При этом один сотрудник входит в список только один раз.

Описание поведения:

1. Выборка данных для отчёта

2. Отправка результата.

Получение расписания официальных рабочих дней

Входные данные:

1. Дата начала отчётного периода

2. Дата конца отчётного периода

Выходные данные — список дней, где каждый элемент содержит:

1. Статус дня: рабочий, выходной или конфликты

2. Дополнительная информация о причине изменений

Описание поведения:

1. Выборка диапазонов выходных из БД.

2. Формирование расписания по дням

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		20

### 3. Отправка результата.

Получение расписания рабочих дней сотрудников

Входные данные:

1. Дата начала отчётного периода
2. Дата конца отчётного периода

Выходные данные — список дней, сгруппированных по сотрудникам.

Где каждый день содержит:

1. Статус дня: рабочий, выходной или конфликтный
2. Дополнительная информация о причине изменений
3. Сумма отработанных сотрудником часов за данный день

Описание поведения:

1. Получение расписания официальных рабочих дней
2. Выборка диапазонов выходных сотрудников из БД и дополнения ими расписания каждого сотрудника.
3. Формирование расписания по дням
4. Дополнение расписания данными о выполненной работе.
5. Отправка результата.

#### 2.1.2 Описание требований к среде выполнения

Требования серверной части

Операционная система:

- Debian 10

Версии ПО:

- Apache 2.4.41

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		21

- PHP 7.4.3
- MySQL 15.1

#### Аппаратное обеспечение:

- Процессор – Intel Core i3, частота 2GHz
- Оперативная память – 6 Гб
- Жесткий диск – 1ТБ

#### Требования клиентской части

- Возможность отправлять HTTP запросы
- Поддержка JSON

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		22

## 2.2 Диаграммы вариантов использования

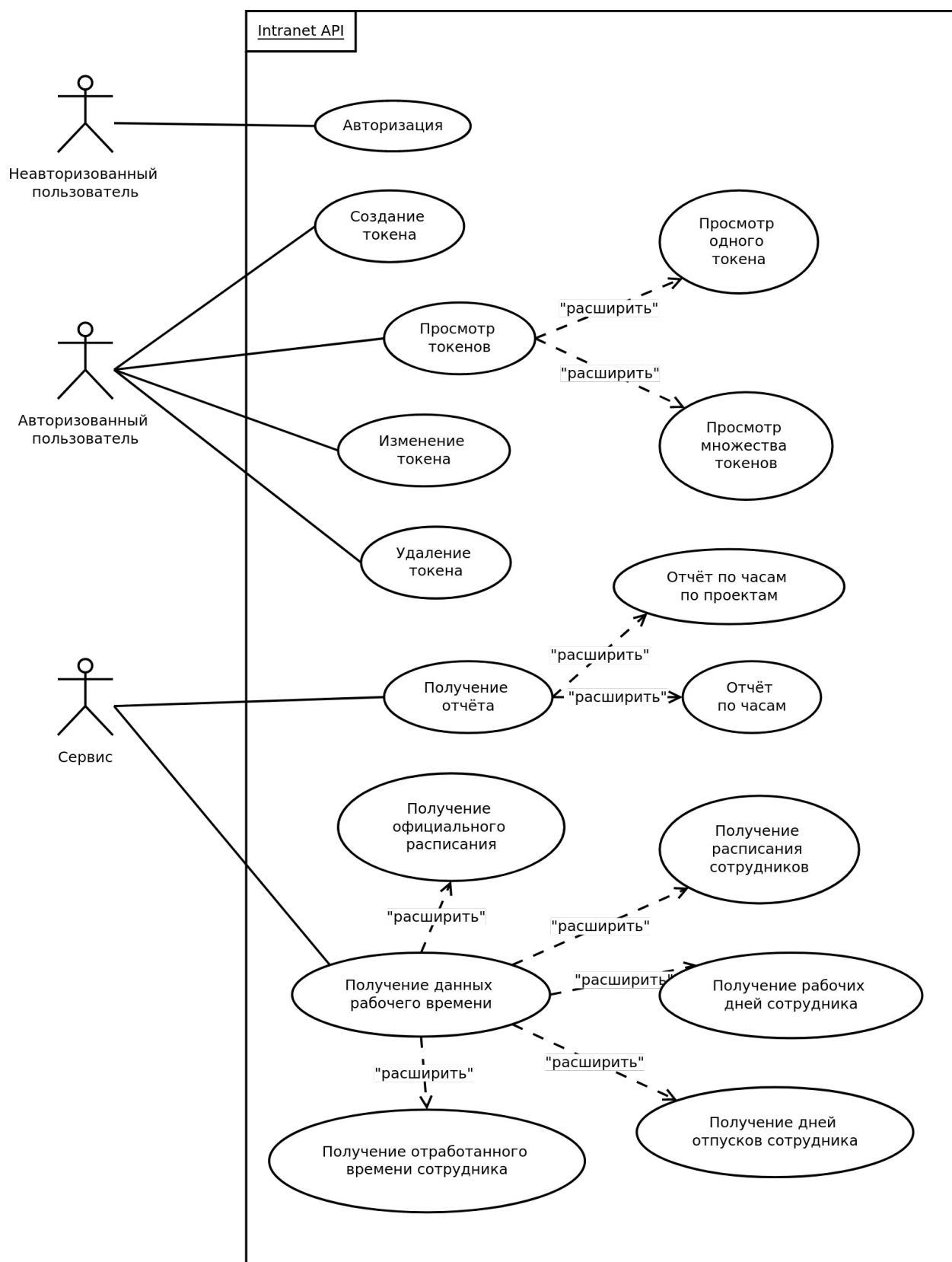


Рисунок 1. Диаграмма вариантов использования

## 2.3 Диаграммы состояний

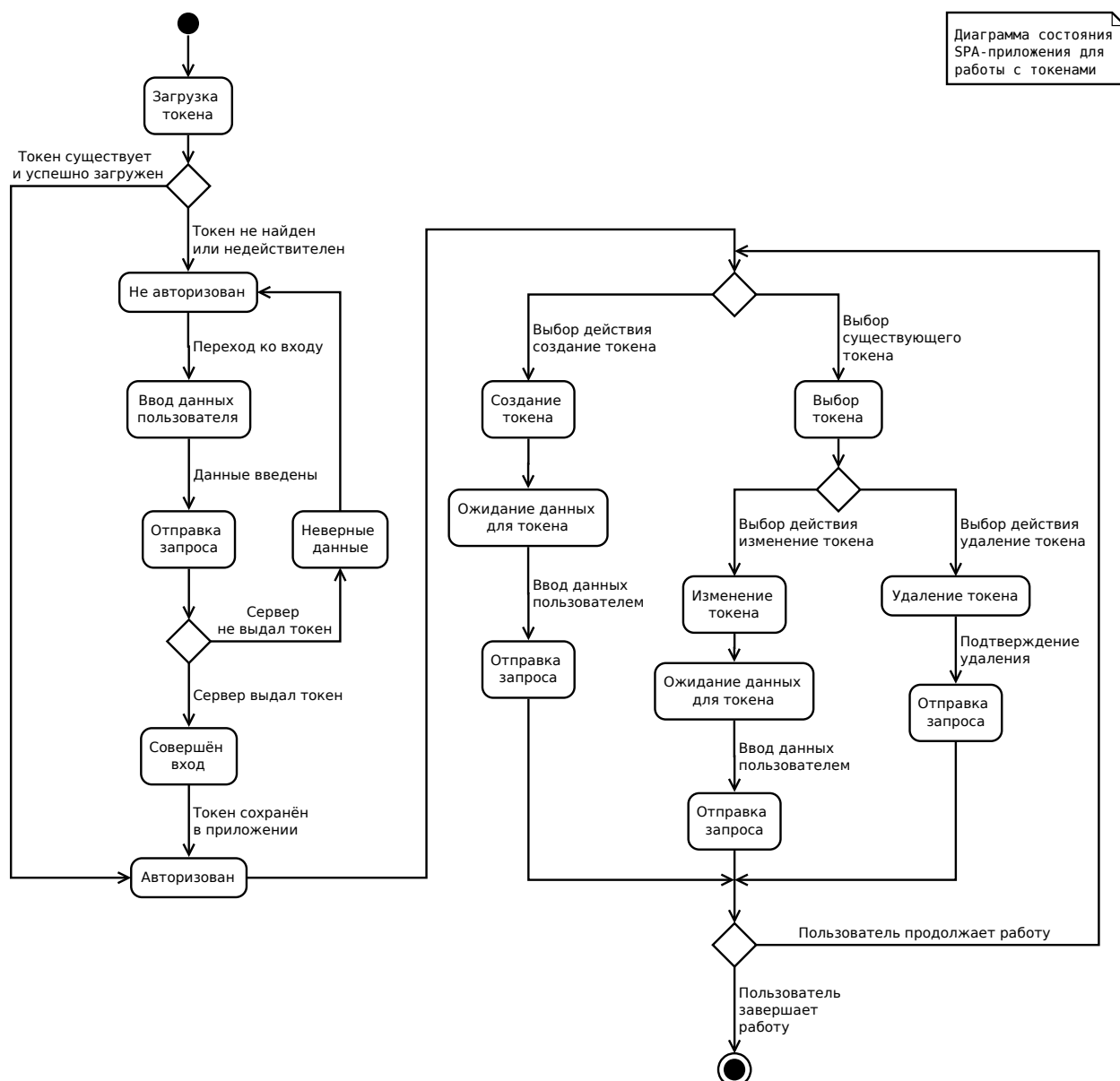


Рисунок 2. Диаграмма состояний абстрактного SPA

## 2.4 Схемы данных

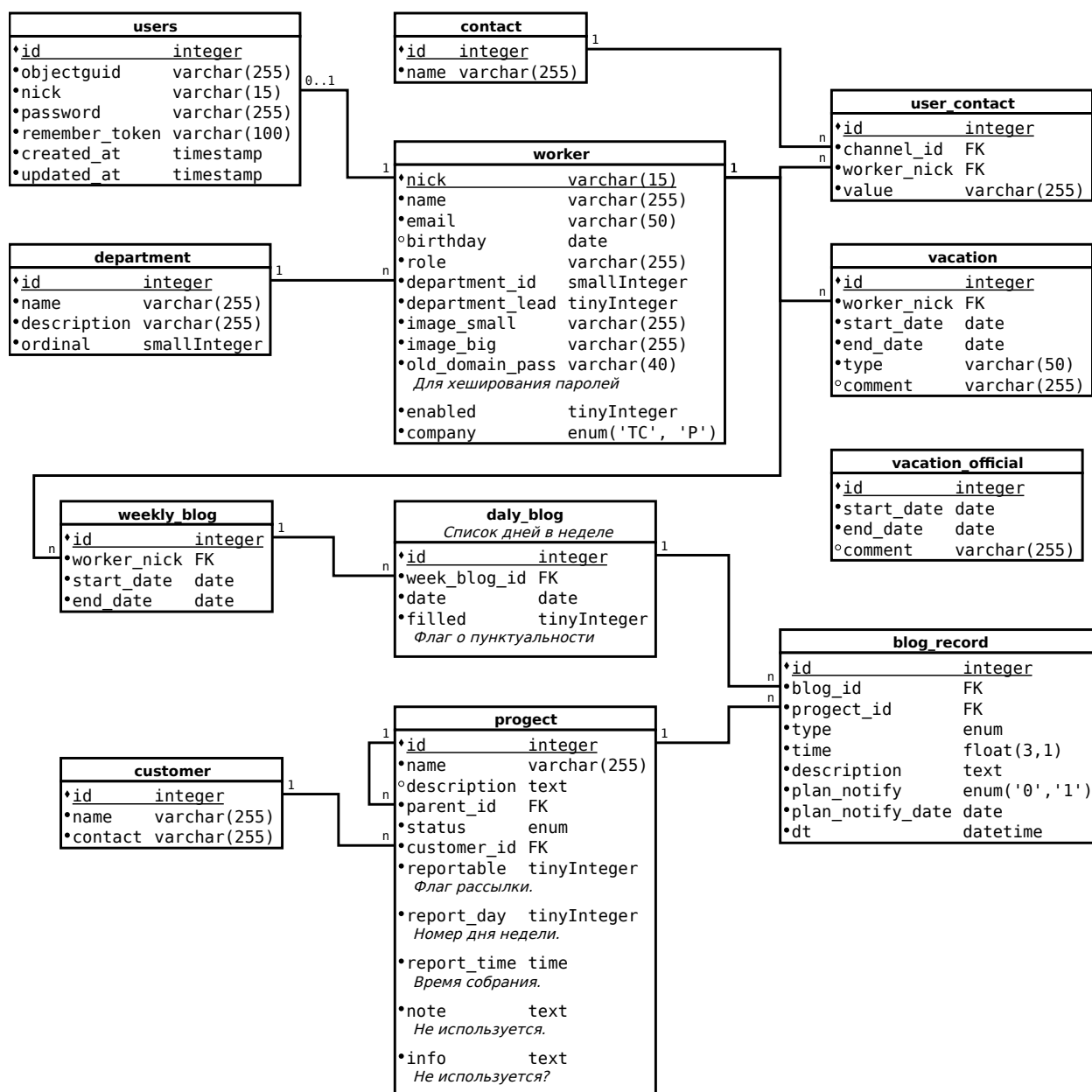


Рисунок 3. Структура данных



## 2.5 Прототип спецификации API

### 2.5.1 Запросы с аутентификацией

Запросы, требующие аутентификации, должны быть отправлены с http заголовком запроса "X-Auth-Key", содержащим токен доступа.

Все запросы, требующие токен доступа, помечены как требующий аутентификации.

Токен доступа можно получить по маршрутам POST /api/login (по нику и паролю пользователя) и POST /api/token (используя токен пользователя).

В случае если запрос пришёл без токена, токен устарел или удалён, то будет выдан следующий ответ:

Код 401 — неаутентифицирован

Тело ответа:

```
{  
  "message": "Unauthenticated."  
}
```

### 2.5.2 Токены

Создание токена

Требует аутентификации.

Создание токена для внешнего сервиса.

Запрос: POST api/token.

Параметры тела:

1. name строка — название токена

Пример ответа:

Код 200 — хорошо.

```
{  
  "data": {  
    "name": "report-service",  
    "token": "{YOUR_AUTH_KEY}"  
  }  
}
```

Чтение токенов пользователя

Требуется аутентификация.

Чтение всех токенов, что принадлежат аутентифицированному пользователю.

Запрос: GET api/token.

Пример ответа:

Код 200 — хорошо.

```
{  
  "data": {  
    "tokens": [  
      {  
        "name": "login",  
        "abilities": [  
          "*"
        ],
```

```

        "last_used_at": "2022-04-01T11:17:50.000000Z"
    },
    {
        "name": "report-service",
        "abilities": [
            "*"
        ],
        "last_used_at": null
    }
]
}
}

```

Чтение токена пользователя

Требует аутентификации.

Чтение указанного токена у аутентифицированного пользователя.

Запрос: GET api/token/{token\_name}

Параметры URL:

token\_name строка — название токена пользователя

Пример ответа:

Код 200 — хорошо.

```

{
    "data": {
        "name": "report-service",

```

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		28

```

    "abilities": [
        "*"
    ],
    "last_used_at": null
}
}

```

Редактирование токена

Требует аутентификации.

Редактирование указанного токена у аутентифицированного пользователя.

Запрос: PUT api/token/{token\_name}.

Параметры URL:

token\_name строка — название токена пользователя

Параметры тела запроса:

1. name строка — название токена
2. abilities массив строк, опционально — список прав доступа, предоставляемых токену

Пример ответа:

Код 200 — хорошо.

```

{
  "data": {
    "name": "report-service",
    "abilities": [

```

```

        "report:getShort",
        "report:getByProject"
    ],
    "last_used_at": null
}
}

```

Удаление токена

Требуется аутентификации.

Запрос: DELETE api/token/{token\_name}.

Параметры URL:

token\_name строка — название токена пользователя

Пример ответа:

Код 204 — хорошо.

Без тела ответа.

### 2.5.3 Пользователи

#### Аутентификация

Аутентификация пользователей. По нику и паролю пользователя выдаётся токен с именем "login", который используется для дальнейшего взаимодействия с api.

Токен "login" является полноценным токеном api и может быть также отредактирован. При создании токена, ему предоставляются все права пользователя.

Если токен "login" был ранее создан, то он будет удалён и создан заново.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		30

Запрос: POST api/login.

Пример ответа:

Код 200 — хорошо.

```
{  
  "data": {  
    "name": "login",  
    "token": "{YOUR_AUTH_KEY}"  
  }  
}
```

#### 2.5.4 Отчёт по часам

Короткий отчёт

Требуется аутентификации.

Получение короткого отчёта о часах сотрудников за определённый период дней.

Простая сумма времени, указанного в блогах, сгруппированные по сотрудникам.

Запрос: GET api/report/hours/short.

Get параметры запроса:

1. start\_date строка, опционально — Дата начала периода отчёта. Включительно. Должен быть корректной датой.
2. end\_date строка, опционально — Дата конца периода отчёта. Включительно. Должен быть корректной датой.

Пример ответа:

Код 200 — хорошо.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		31

```

{
  "data": {
    "count_reports": 2,
    "reports": [
      {
        "nick": "nherman",
        "name": "Myrna Toy II",
        "company": "TC",
        "total_hours": 4.7
      },
      {
        "nick": "sterling03",
        "name": "Oliver Daniel",
        "company": "TC",
        "total_hours": 8.2000000000000001
      }
    ]
  }
}

```

Отчёт по проектам

Требуется аутентификации.

Получение отчёта о часах сотрудников по проектам за определённый период дней.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		32

Если на одном проекте работало несколько сотрудников, то создаётся несколько записей в массиве "reports" с одинаковым значением полей, относящихся к проекту.

Запрос: GET api/report/hours/project.

Get параметры запроса:

1. start\_date строка, опционально — Дата начала периода отчёта. Включительно. Должен быть корректной датой.
2. end\_date строка, опционально — Дата конца периода отчёта. Включительно. Должен быть корректной датой.

Пример ответа:

Код 200 — хорошо.

```
{
  "data": {
    "count_reports": 4,
    "reports": [
      {
        "nick": "nherman",
        "name": "Myrna Toy II",
        "company": "TC",
        "customer_id": 24,
        "customer_name": "est explicabo nesciunt",
        "project_id": 26,
        "project_name": "dolores neque quae",
        "total_hours": 4.7
      }
    ]
  }
}
```



```

    },
    {
      "nick": "sterling03",
      "name": "Oliver Daniel",
      "company": "TC",
      "customer_id": 28,
      "customer_name": "aut et voluptatem",
      "project_id": 12,
      "project_name": "dicta reiciendis asperiores",
      "total_hours": 4.4
    },
  ]
}
}

```

## 2.5.5 Расписания

Общее расписание

Требует аутентификации.

Получение расписания рабочих дней компании.

Запрос: GET api/schedule/official.

Get параметры запроса:

1. start\_date строка, опционально — Дата начала периода отчёта. Включительно. Должен быть корректной датой.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		34

2. end\_date строка, опционально — Дата конца периода отчёта.  
Включительно. Должен быть корректной датой.

Пример ответа:

Код 200 — хорошо.

```
{
  "data": {
    "2010-10-07": {
      "status": "holiday",
      "info": [
        "Official holiday"
      ]
    },
    "2010-10-08": {
      "status": "holiday",
      "info": [
        "Official holiday"
      ]
    },
    "2010-10-09": {
      "status": "work",
      "info": []
    }
  }
}
```

}

Расписание сотрудников

Требует аутентификации.

Получение расписания рабочих дней сотрудников.

Запрос: GET api/schedule/worker.

Get параметры запроса:

1. start\_date строка, опционально — Дата начала периода отчёта. Включительно. Должен быть корректной датой.
2. end\_date строка, опционально — Дата конца периода отчёта. Включительно. Должен быть корректной датой.

Пример ответа:

Код 200 — хорошо.

{

  "ilene.wolf": {

    "2010-10-07": {

      "status": "holiday",

      "info": [

        "Error possimus maiores eius est sit molestiae qui qui."

    ]

  },

  "2010-10-08": {

    "status": "work",

    "info": [

"Error possimus maiores eius est sit molestiae qui qui.",

"Provident delectus et atque sequi."

],

"total\_hours": 5

},

"2010-10-09": {

"status": "work",

"info": []

}

}

}

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		37

### 3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

#### 3.1 Расчет затрат на разработку программы и решение задачи на ЭВМ

Основными компонентами затрат на разработку программы и решение задачи на ЭВМ являются затраты, связанные с оплатой труда специалистов на разработку программы, обслуживание и эксплуатацию ЭВМ в период отладки программы и решения задачи, то есть рассчитываются прямые и косвенные затраты.

При определении полной себестоимости программы учтены расходы по заработной плате, отчисления в социальные статьи и составлена калькуляция затрат в следующей последовательности:

- основная заработная плата персонала;
- дополнительная заработная плата персонала;
- отчисления во внебюджетные фонды;
- стоимость работ на ЭВМ;
- расчет косвенных затрат на разработку программы.

При расчете всех экономических показателей была составлена таблица 1, в которой указаны все этапы работы по разработке программы и решению задачи, исполнитель каждого этапа, трудоемкость и стоимость исполнения.

Стоимость каждого этапа определена, исходя из оклада исполнителей и времени выполнения этапа.

Количество рабочих часов в месяце равно 168 часов, то есть 21 рабочий день в месяце по 8 часов.

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		38

Стоимость часа работы определяется по следующей формуле 1.

$$Ст.ч. = \frac{Оклад.}{К.р.ч} \quad (1)$$

где Оклад. – оклад, руб.;

Ст.ч. – стоимость часа работы;

К.р.ч. – количество рабочих часов в месяце, час.

Стоимость часа работы руководителя

Ст.ч.р = 58800/ 168 = 350 руб./час.

Стоимость часа работы программиста

Ст.ч.п = 15000 / 168 = 89,29 руб./час.

Таблица 1 — Этапы разработки

Наименование этапов работ	Исполнитель	Трудоемкость, час	Плата за час, руб./час	Стоимость исполнения, руб.
Постановка задачи	Руководитель	9	350	3150
	Программист	13	89,29	1160,71
Изучение литературы	Программист	19	89,29	1696,43
Технический проект	Программист	23,5	89,29	2098,21
	Руководитель	3	350	1050
Эскизный проект	Программист	10	89,29	892,86
	Руководитель	3	350	1050
Написание кода	Программист	85	89,29	7589,29
Отладка программы Тестирование	Программист	29	89,29	7589,29
	Руководитель	8	350	2800
Оптимизация программы	Программист	50	89,29	4464,29
Оформление сопроводительной документации	Программист	20	89,29	1785,71
	Руководитель	1	350	350
Итого	Программист	249	89,29	22276,79
	Руководитель	24	350	8400

Основная заработная плата персонала рассчитывается по формуле 2:

$$ЗП = Сти * РК \quad (2)$$

где ЗП – основная заработная плата персонала;

Сти – стоимость исполнения из таблицы 1 для каждого исполнителя, руб.;

РК – районный коэффициент (1,3).

Основная заработная плата руководителя

$$ЗП_{рук.} = 8400 \times 1,3 = 10920 \text{ руб.}$$

Основная заработная плата программиста

$$ЗП_{пр.} = 22276,79 \times 1,3 = 28959,82 \text{ руб.}$$

Дополнительная заработная плата персонала рассчитывается по формуле 3

$$ЗП_{доп} = ЗП * 0,10 \quad (3)$$

где ЗП<sub>доп</sub> – дополнительная заработная плата персонала;

ЗП – основная заработная плата персонала, руб.

Дополнительная заработная плата руководителя

$$ЗП_{доп} = 10920 \times 0,1 = 1092 \text{ руб.}$$

Дополнительная заработная плата программиста

$$ЗП_{доп} = 28959,82 \times 0,1 = 2895,98 \text{ руб.}$$

Отчисления во внебюджетные фонды рассчитываются по формуле 4

$$О_{сн} = (ЗП + ЗП_{доп}) \times 0,302 \quad (4)$$

где ЗП<sub>доп</sub> - дополнительная заработная плата;

ЗП – основная заработная плата;

О<sub>сн</sub> - отчисления во внебюджетные фонды (30,2%).

Отчисления во внебюджетные фонды составляют 30,2%, из них:

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		40

- в пенсионный фонд( $O_{пф}$ ) отчисляется 22%;
- в фонд социального страхования( $O_{сстр}$ ) – 2,9%;
- в фонд медицинского страхования( $O_{мс}$ ) – 5,1%;
- страхование от несчастных случаев на производстве – 0,2%

Отчисления во внебюджетные фонды от заработной платы руководителя рассчитывается по формуле 4,

где  $O_{пф}$  – отчисления в пенсионный фонд;

$O_{сстр}$  – отчисления в фонд социального страхования;

$O_{мс}$  – отчисления в фонд медицинского страхования;

$O_{сс}$  – отчисления во внебюджетные фонды..

$$O_{пф} = (10920 + 1092) \times 0,22 = 2642,64 \text{ руб.};$$

$$O_{сстр} = (10920 + 1092) \times 0,029 = 348,35 \text{ руб.};$$

$$O_{мс} = (10920 + 1092) \times 0,051 = 612,61 \text{ руб.};$$

$$O_{сс} = (10920 + 1092) \times 0,302 = 3627,62 \text{ руб.}$$

Отчисления во внебюджетные фонды от заработной платы программиста рассчитывается по формуле 4:

$$O_{пф} = (28959,82 + 2895,98) \times 0,22 = 7008,28 \text{ руб.};$$

$$O_{сстр} = (28959,82 + 2895,98) \times 0,029 = 923,82 \text{ руб.};$$

$$O_{мс} = (28959,82 + 2895,98) \times 0,051 = 1624,65 \text{ руб.};$$

$$O_{сс} = (28959,82 + 2895,98) \times 0,302 = 9620,45 \text{ руб.}$$

Для расчёта стоимости работ на ЭВМ учтём амортизацию ЭВМ на период написания программы и расходы на электроэнергию, используемую при разработке программы.

Стоимость работ на ЭВМ рассчитывается по формуле 5

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		41



$$C_{pm} = C_{mч} \times T_m \quad (5)$$

где  $C_{mч}$  – стоимость машинного часа в рублях;

$C_{pm}$  – стоимость работ на ЭВМ;

$T_m$  – общее время работы ЭВМ (час).

Стоимость работ на ЭВМ

$$C_{pm} = 0,3 \times 273,5 = 82,05 \text{ руб.}$$

Расчет косвенных расходов на разработку программы рассчитывается по формуле 6

$$P_k = ЗП * K_{нр} \quad (6)$$

где  $P_k$  – косвенные расходы на разработку программы;

$ЗП$  – основная заработная плата персонала;

$K_{нр}$  – коэффициент накладных расходов (5-10%).

$$P_k = 39879,82 \times 0,05 = 1993,99 \text{ руб.}$$

Полная себестоимость программы приведена в таблице 2.

Таблица 2 — Смета затрат на разработку

Наименование статей расходов	Стоимость работ (руб.)
Основная заработная плата	39879,82
Дополнительная заработная плата	3987,98
Отчисления во внебюджетные фонды	13248,08
Стоимость работ на ЭВМ	82,05
Косвенные расходы	1993,99
Итого	59191,92

### 3.1.1 Расчет годовых затрат на эксплуатацию программы

Стоимость одного непосредственного решения на ЭВМ определяется по формуле 7

$$C_{p.m} = C_{мч} \times T_p + 3П_{o.п.} \times Q \times K_p \times K_{кр} \quad (7)$$

где  $C_{p.m}$  – стоимость одного непосредственного решения на ЭВМ;

$C_{мч}$  – стоимость работы на ЭВМ за час (руб./час);

$T_p$  – время решения задачи на ЭВМ (час);

$Q$  – трудоемкость исполнителя (час);

$K_p$  – районный коэффициент (1,3);

$K_{кр}$  – коэффициент косвенных расходов (1,05);

$3П_{o.п.}$  – заработная плата за час работника (руб./час).

Стоимость одного непосредственного решения на ЭВМ

$$C_{p.m} = 2,31 \times 0,0025 + 0,01 \times 178,57 \times 1,3 \times 1,05 = 2,44 \text{ руб.}$$

Расчёт годовых затрат на эксплуатацию программы необходимо провести для последующего анализа эффективности данного программного продукта.

Готовые затраты на эксплуатацию программы рассчитываются по формуле 8

$$C_{p.m.год} = N \times C_{p.m} + E_n \times C \quad (8)$$

где  $N$  – плотность потока заявок (заявок в год);

$C_{p.m.год}$  – годовые затраты на эксплуатацию программы;

$C_{p.m}$  – стоимость одного непосредственного решения на ЭВМ;

$E_n$  – нормальный коэффициент сложности (0,2-0,6);

$C$  – себестоимость разработки программы (итог таблицы 2).

Расчёт годовых затрат на эксплуатацию программы

$$C_{\text{р.м.год}} = 150 \times 2,44 + 0,2 \times 58697,93 = 12106,08 \text{ руб.}$$

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		44

## 3.2 Расчет экономического эффекта и определение срока окупаемости

Экономический эффект достигается при эксплуатации и характеризуется экономией времени работы специалиста, повышением производительности труда.

Для того чтобы определить экономическую эффективность проекта необходимо рассчитать затраты на эксплуатацию ранее употреблявшимся образом.

### 3.2.1 Расчёт годовых затрат на выполнение работ ранее употреблявшимся способом

Расходы на выполнение работ ранее употреблявшимся способом рассчитываются по формуле 9

$$C_{p.cn} = 3П_{cn} \times T_{cn} \times K_{кр} \times K_p \quad (9)$$

где  $3П_{cn}$  – заработная плата специалиста за час (руб./час);

$T_{cn}$  – затраты времени специалиста на выполнение работ ранее употреблявшимся способом (ч);

$K_p$  – районный коэффициент (1,3);

$K_{кр}$  – коэффициент косвенных расходов (1,05).

Расходы на выполнение работ ранее употреблявшимся способом:

$$C_{p.cn} = 125 \times 5 \times 1,05 \times 1,3 = 853,125 \text{ руб.}$$

Зная стоимость всех работ по выполнению одной задачи, определим годовые расходы ранее употреблявшимся способом:

$$C_{p.cn.god} = N \times C_{p.cn} \quad (10)$$

где  $N$  – плотность потока заявок (заявок в год);

$C_{р.сп.год}$  – годовые расходы ранее употреблявшимся способом.

Расчёт годовых затрат на выполнение работ ранее употреблявшимся способом

$$C_{р.сп.год} = 150 \times 853,13 = 127968,75 \text{ руб.}$$

					ДП.22.09.02.07.482.06.ПЗ	Лист
Изм	Лист	№ докум.	Подпись	Дата		46

### 3.3 Экономический эффект и срок окупаемости

Экономия рассчитывается по формуле

$$\mathcal{E}_{\text{год}} = C_{\text{р.сп.год}} - C_{\text{р.м.год}} \quad (11)$$

где  $\mathcal{E}_{\text{год}}$  – экономия в год;

$C_{\text{р.сп.год}}$  – годовые затраты на выполнение работ ранее употреблявшимся способом;

$C_{\text{р.м.год}}$  – годовые затраты на эксплуатацию программы.

Определение коэффициента экономической эффективности программы.

Коэффициента экономической эффективности показывает сколько на 1 руб. вложенных затрат в разработку и эксплуатацию, получаем экономии. Чем больше данное значение, тем эффективнее проект.

Данный коэффициент рассчитывается по формуле 12

$$E_p = \frac{\mathcal{E}_{\text{год}}}{C + C_{\text{р.м.год}}} \quad (12)$$

Рассчитаем экономию, связанную с использованием разработки:

$$\mathcal{E} = C_{\text{р.сп.год}} - C_{\text{р.м.год}} = 127968,75 - 12204,88 = 115763,87$$

Рассчитаем экономическую эффективность программы:

$$E_p = \frac{\mathcal{E}_{\text{год}}}{C + C_{\text{р.м.год}}} = \frac{115763,87}{127968,75 + 59191,92} = 0,62$$

Экономический эффект показывает, что на 1 вложенный рубль в разработку и эксплуатацию программы, получаем 0,62 рублей экономии. Так как проект не предполагает коммерциализации, мы не можем посчитать его коммерческую эффективность, но в результате внедрения программы

облегчается труд специалиста, снижаются затраты времени на решение задач.

Срок окупаемости программы рассчитываем исходя из экономии. То есть благодаря экономии за какой период времени окупятся затраты на разработку и внедрение программы.

$$T_{ок} = \frac{1}{E_p} \quad (13)$$

Рассчитаем срок окупаемости программы:

$$T_{ок} = \frac{1}{0,62} = 1,62$$

Таким образом, программа окупится через 1,62 года.

Выводы. На основании проведённых расчётов себестоимости и экономического эффекта можно сделать следующие выводы. Результаты технико-экономического обоснования свидетельствуют об экономической эффективности проекта. За счёт снижения эксплуатационных затрат проект окупится через 19,4 месяца.

## Заключение

В ходе выполнения преддипломной практики было разработано приложение для интеграции сервисов с данными компании Томсксофт.

Данное приложение является первой версией и закладывает основу для развития.

### Реализованный функционал

В ходе выполнения работы был разработан следующий функционал системы:

1. Авторизация
2. CRUD токенов
3. Формирование отчётных данных по часам сотрудников и по часам сотрудников по проектам
4. Получение расписания рабочих дней компании
5. Получение расписания рабочих дней сотрудников компании
6. Автоматические тесты разрабатываемой системы



## Список использованной литературы

1. Официальная документация Laravel [Электронный ресурс] URL: <https://laravel.com/> (дата обращения: 10.04.2022)
2. Документация Laravel от сообщества [Электронный ресурс] URL: <https://laravel.su/docs/8.x/> (дата обращения: 10.04.2022)
3. Документация Laravel Sanctum от сообщества [Электронный ресурс] URL: <https://laravel.su/docs/8.x/sanctum> (дата обращения: 10.04.2022)
4. Официальная документация php [Электронный ресурс] URL: <https://www.php.net> (дата обращения: 10.04.2022)
5. README.md eloquent-power-joins [Электронный ресурс] URL: <https://github.com/kirschbaum-development/eloquent-power-joins> (дата обращения: 16.05.2022)
6. Laravel — Википедия [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/Laravel> (дата обращения: 10.04.2022)

					Лист
ДП.22.09.02.07.482.06.ПЗ					
Изм	Лист	№ докум.	Подпись	Дата	