

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

подпись инициалы, фамилия

« _____ » _____ 2020 г

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – “Информатика и вычислительная техника”

Веб-приложение для тайм-менеджмента

Руководитель	_____	<u>доцент, канд. техн. наук</u>	<u>Постников А.И.</u>
	подпись, дата	должность, учёная степень	инициалы, фамилия
Выпускник	_____		<u>М.М.Байгин</u>
	подпись, дата		инициалы, фамилия
Консультант	_____	<u>старший преподаватель</u>	<u>И. В. Матковский</u>
	подпись, дата	должность, учёная степень	инициалы, фамилия
Нормоконтролер	_____	<u>доцент, канд. техн. наук</u>	<u>Постников А.И.</u>
	подпись, дата	должность, учёная степень подпись, дата	инициалы, фамилия

Красноярск 2020

РЕФЕРАТ

Выпускная квалификационная работа по теме «Веб-приложение для тайм-менеджмента» содержит 48 страниц текстового документа, 33 иллюстрации, 6 таблиц, 10 использованных источников.

ТАЙМ-МЕНЕДЖМЕНТ, ВЕБ-ПРИЛОЖЕНИЕ, ГРАФИКИ, ЗАДАЧИ, ВРЕМЯ, ПРОГРАММНЫЙ ПРОДУКТ.

Объект исследования – тайм-менеджмент.

Целью работы является разработка веб-приложения для тайм-менеджмента, позволяющего отслеживать задачи и генерировать отчеты.

В результате исследования процесса тайм-менеджмента была определена структура тайм-менеджера, установлены основные функциональные модули, особенности различных тайм-менеджеров.

В итоге была разработана структура веб-приложения, его серверная и клиентская части, а также взаимодействие между ними и между базой данных. В качестве технической реализации было создано веб-приложение для тайм-менеджмента.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Исследование предметной области	7
1.1 Общие сведения о тайм-менеджерах.....	7
1.2 Структурная организация тайм-менеджеров.....	8
1.3 Анализ процесса тайм-менеджмента	9
1.4 Обзор существующего программного обеспечения	10
1.4.1 Toggl.....	11
1.4.2 Harvest.....	12
1.4.3 Hours	14
1.4.4 Timely.....	15
1.5 Анализ рассмотренных ресурсов.....	17
1.6 Требования к разрабатываемому программному продукту.....	18
2 Проектирование программного продукта.....	19
2.1 Описание структуры тайм-менеджера	19
2.2 Логическая модель программного продукта	20
2.3 Разработка структуры базы данных	22
2.4 Выбор языка разработки.....	23
2.5 Выбор системы управления базой данных	25
3 Программная реализация программного продукта.....	25
3.1 Создание базы данных	25
3.2 Клиентская часть	27
3.3 Серверная часть	29
4 Описание результатов работы.....	30
4.1 Страницы регистрации и авторизации	30
4.2 Страница с задачами	32
4.3 Страница с группами.....	36

4.4 Страница с графиками	39
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	48

ВВЕДЕНИЕ

В современном мире информационные и компьютерные технологии являются неотъемлемой частью жизни человека. Все чаще привычные предметы и процессы становятся частью компьютерного мира. Интернет, в свою очередь, позволяет объединять людей, процессы, вещи в единую сеть, в которой они плотно взаимодействуют между собой.

Развитие интернет технологий позволяет переносить повседневные дела и процессы из реального мира в виртуальный. Одним из таких процессов является контроль затраченного на выполнение различных задач времени. Перенос данной задачи в электронную среду предлагает в более удобном формате производить анализ затраченного времени. Анализ затраченного времени с целью повышения эффективности использования времени называется тайм-менеджментом.

Время – это то, из чего состоит наша жизнь. Главная цель тайм-менеджмента – отследить узкие места в работе и оптимизировать процесс [1].

Для эффективного тайм-менеджмента существуют различные тайм-менеджеры. Тайм-менеджер – некоторое устройство или приложение, помогающее отслеживать время, а после предоставлять пользователю отчеты по затраченному времени. Бумажные носители менее удобны в использовании, а также не способны выводить различные графики в удобном для просмотра виде. Отслеживание занятости группы возможно только в том случае, когда все люди из группы находятся в непосредственной близости к источнику записи данных. Для таких ситуаций в настоящее время существуют цифровые тайм-менеджеры и их обязательная составляющая – тайм-трекер. Тайм-трекер позволяет отследить время для какой-либо одной задачи. Тайм-менеджер оперирует с полученными данными и генерирует отчеты.

Несмотря на большое разнообразие существующих тайм-менеджеров, достаточно сложно определить с каким-то конкретным, ведь у всех них есть

свои особенности и отличия. Поэтому следует произвести анализ и сравнение аналогов, а затем определиться с основными функциями и критериями, которыми должна обладать программа.

1 Исследование предметной области

Для того, чтобы создать какой-либо программный продукт, нужно сперва исследовать область применения данного продукта. Затем произвести детальный обзор и анализ. Исследование области позволяет сформировать требования к создаваемому программному продукту и отвечает на вопрос целесообразности создания какого-либо продукта. Определение основных свойств продукта происходит на данном этапе. Исследование предметной деятельности формирует базовые требования к разрабатываемому программному обеспечению.

1.1 Общие сведения о тайм-менеджерах

Тайм-менеджеры чаще всего представляют собой приложение или веб-сайт. Тайм-менеджер включает себя тайм-трекер – некоторую программу, которая позволяет отслеживать начало и конец каждой задачи. Помимо тайм-трекера, тайм-менеджер должен иметь раздел с отчетами, частным случаем которых являются графики. Графики наглядно отображают занятость пользователя системы. Тайм-менеджер – целая система, состоящая из различных взаимосвязанных компонентов.

Тайм-менеджер в самом простом случае позволяет вводить имя задачи, начинать задачу, завершать ее, затем отслеживать затраченное время. Приложения для компьютеров могут отслеживать положения компьютерной мыши, делать снимки экранов, записывать данные о запущенных приложениях, после чего выводить подробную статистику. Но такой подход может быть достаточно избыточным, а также требует установки данной программы на персональный компьютер пользователя программного обеспечения. Веб-сайты не могут отслеживать запущенные приложения, но могут отслеживать открытые вкладки в браузере. Веб-сайты не нужно устанавливать на компьютер, а достаточно иметь

доступ в Интернет. Различные ситуации и требования порождают различные виды тайм-менеджеров.

Чаще всего пользователям тайм-менеджеров не требуется захват изображения экрана или отслеживание открытых приложений или вкладок, при условии, что пользователь использует тайм-менеджер для самоконтроля. И в целях безопасности своих данных не каждый пользователь захочет использовать такие возможности.

При проектировании следует учитывать внешний вид тайм-менеджера, ведь для пользователя данного продукта этот момент является важным аспектом. Следует проработать функционал и расположение элементов системы для максимального удобства пользования.

1.2 Структурная организация тайм-менеджеров

С каждым месяцем и годом появляется все больше различных программ, в том числе и тайм-менеджеров. Большинство из существующих реализаций тайм-менеджеров имеют схожие проблемы такие как ограниченный функционал в бесплатной версии или же вообще наличие платных функций. Неудобный интерфейс или отсутствие каких-либо возможностей также влияет на появление новых сервисов по отслеживанию и контролю времени.

В общем случае тайм-менеджеры имеют следующие функции и возможности:

- возможность создания задачи;
- мониторинг выполненных задач;
- вывод различных отчетов, например, статистика за неделю или за день.

Тайм-менеджеры могут быть двух видов – как отдельное приложение, устанавливаемое на целевую платформу или в виде интернет странички.

Использование интернет технологий позволяет пользователю не задумываться о выделении памяти для установки приложений, а также в дальнейшем об обновлении приложения при выходе новых версий продукта. Данное решение требует постоянного доступа к сети интернет, но в современном мире почти у каждого есть постоянный доступ к интернету. Разработка веб-приложений позволяет уйти от особенностей разработки под конкретную платформу с последующим переписыванием кода под другую платформу. Следует учитывать только размер экрана.

1.3 Анализ процесса тайм-менеджмента

Процесс тайм-менеджмента заключается в отслеживании времени, затраченного на выполнение каких-либо задач, с последующим отображением полученной статистики. Статистика в виде графиков показывает затраченное время в удобном виде и позволяет анализировать процессы.

Однако, не всегда пользователь вовремя добавляет задачу или ее завершает. Поэтому следует предусмотреть возможность ручного ввода начала и конца времени выполнения для любой задачи. Это позволяет в случае необходимости скорректировать временной промежуток.

В процессе тайм-менеджмента основным фактором является наглядность полученной статистики и информации. Графики должны отображать затраченное время по дням, неделям и месяцам. Такое разбиение по времени является самым оптимальным и привычным.

Зачастую пользователь не работает один. Групповые проекты также нуждаются в менеджменте и имеют свои особенности.

Так как целью тайм-менеджмента является наглядность, следует предусмотреть различный набор типов графиков, например круговая диаграмма или столбчатая.

На рисунке 1 изображены основные возможности пользователя тайм-менеджера.

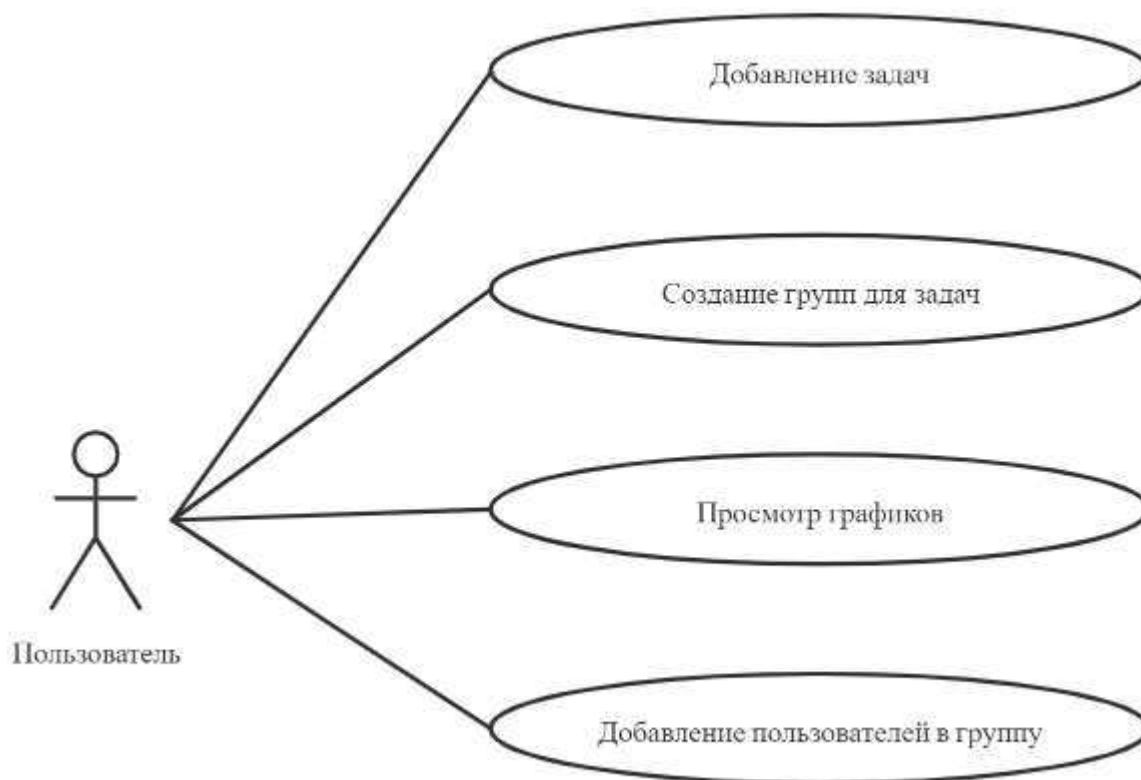


Рисунок 1 - Основные возможности пользователя

1.4 Обзор существующего программного обеспечения

В настоящее время существует обширный выбор программных продуктов. Отличаются они как по внешнему виду, так и по набору возможностей. Поэтому следует заранее оценить все плюсы и минусы аналогов. Сравнительный анализ позволяет не только учитывать слабые стороны существующих продуктов, но и принять во внимание их сильные возможности.

1.4.1 Toggl

Инструмент, главным образом, разрабатывался для фрилансеров и индивидуальных предпринимателей, хотя Toggl также позволяет отслеживать активности целой команды сотрудников. Toggl – инструмент, который предоставляет определенную гибкость в организации трекинга [2].

Toggl имеет упрощенный интерфейс, в котором есть возможность распределять задачи в рамках проектов, быстро выбирать клиентов, в том числе создавать проекты или добавлять клиентов, не выходя за рамки панели задач [3].

Приложение доступно веб-интерфейс, есть браузерные расширения, мобильное приложение и приложение для десктопа. В последнем можно автоматически отслеживать определенные активности и устанавливать напоминания.

К плюсам можно отнести реализацию системы тегов по проектам и клиентам. Создание новых проектов/тегов занимает несколько секунд и не требует перезагрузки страницы. Детальные отчеты, которые помогают визуализировать расходы, или генерировать отчеты для клиентов; также присутствуют фильтры.

К слабой стороне можно отнести отсутствие работы с группами в бесплатной версии сайта, неадаптированную мобильную версию сайта, невозможность выбрать для задачи время начала и конца в разных днях.

На рисунке 2 изображен внешний вид странички с тайм-трекером.

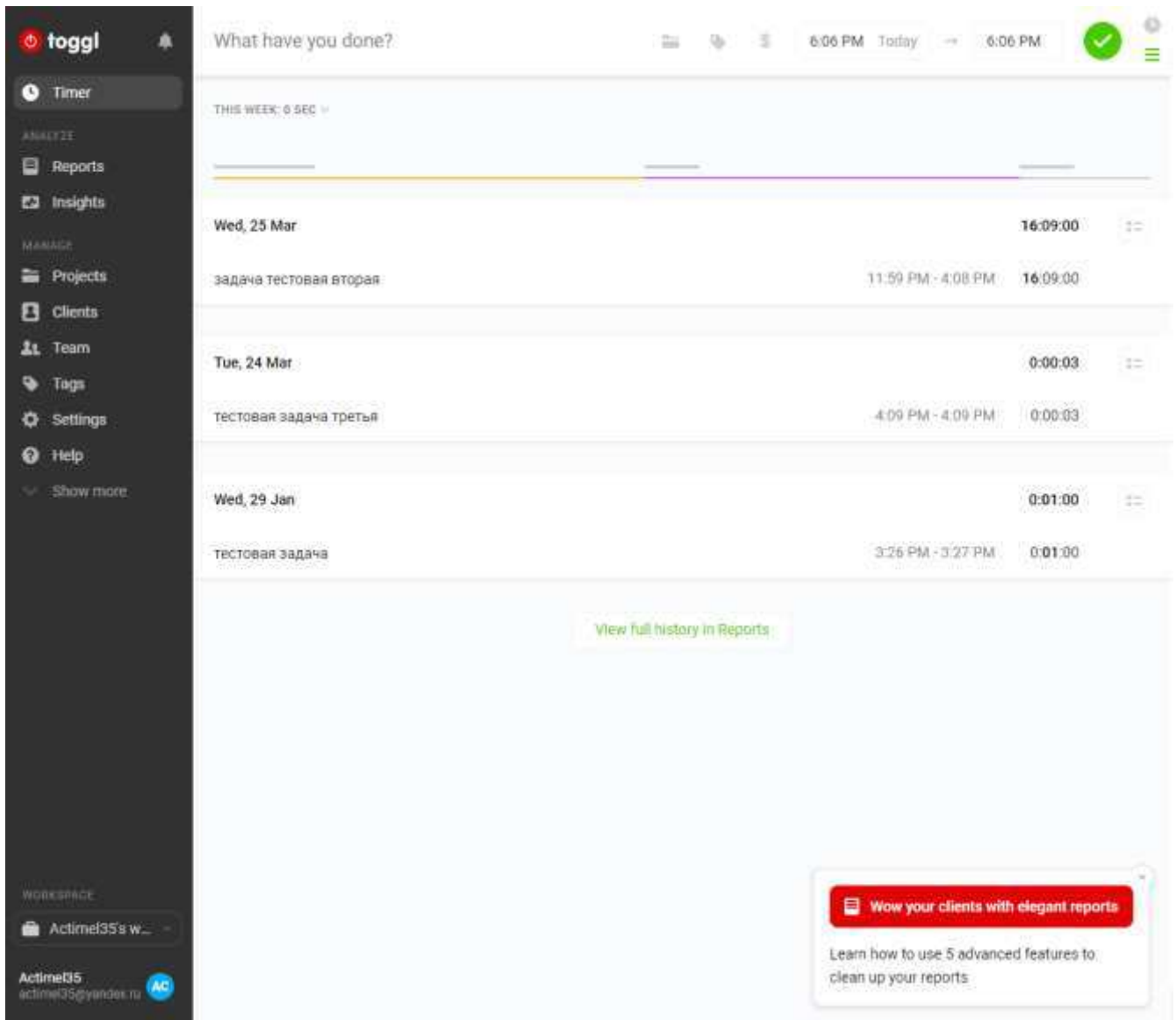


Рисунок 2 - Внешний вид Toggl

1.4.2 Harvest

Тайм-менеджер Harvest больше подходит для фрилансеров и удаленных команд. Реализована функция отслеживания расходов, создание инвойсов. Основным направлением данного тайм-менеджера является бизнес-сегмент [4]. Данное позиционирование привело к созданию тех возможностей, которые описаны ниже.

К плюсам можно отнести функцию создания счетов, включая поддержку платежей бизнес-платежи PayPal Business Payments (подходящий вариант для пользователей со счетами в американских банках). Множество интеграций со сторонними инструментами, включая управление проектами, создание счетов, CRM, клиентская поддержка, инструменты разработчика и т.д. Время отслеживается посредством веб-приложений, расширений для браузеров, приложений для десктопов и мобильных устройств. Встроенная функция отслеживания расходов, включая сканирование чеков об оплате.

Ориентация на бизнес сегмент является как плюсом, так и минусом. Бесплатная версия позволяет создавать всего два проекта и быть в них единственным лицом, работать в команде с помощью данного сервиса бесплатно не получится. Перегруженность интерфейса является следствием возможностей. Для создания новых проектов понадобится больше кликов и перезагрузок страниц, чем в аналогичных приложениях, что не очень удобно для фрилансеров, которым приходится создавать множество проектов. Интерфейс отчетов не настолько гибок, как в других инструментах, хотя большинство пользователей не жалуется.

На рисунке 3 показан внешний вид странички Harvest.

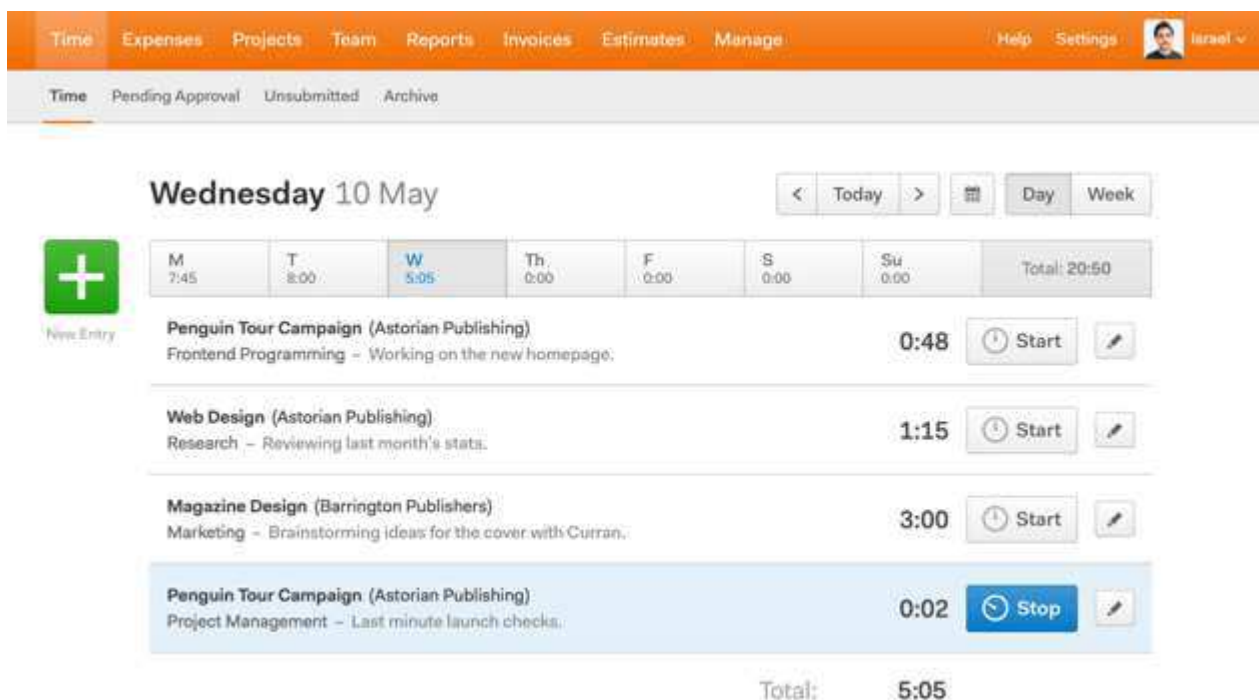


Рисунок 3 - Внешний вид Harvest

1.4.3 Hours

Hours подходит для тех пользователей, которые ищут простой тайм-трекер с минимальным интерфейсом [5].

К сильной стороне можно отнести функцию Timeline. Она позволяет мгновенно заметить пробелы и при желании заполнить их, например, если пользователь забыл добавить какую-либо задачу.

Данный тайм-трекер сложно назвать тайм-менеджером из-за очень ограниченного функционала в бесплатной версии. Отсутствие графиков, команд. Самым сильным минусом является платная веб версия, бесплатная версия только под устройства с операционной системой iOS. Данный тайм-трекер подходит для тех, кому нужно со смартфона под управлением iOS отслеживать активность в течении дня, для чего-то более крупного или брать платную подписку, или искать альтернативы.

На рисунке 4 показан внешний вид Hours.

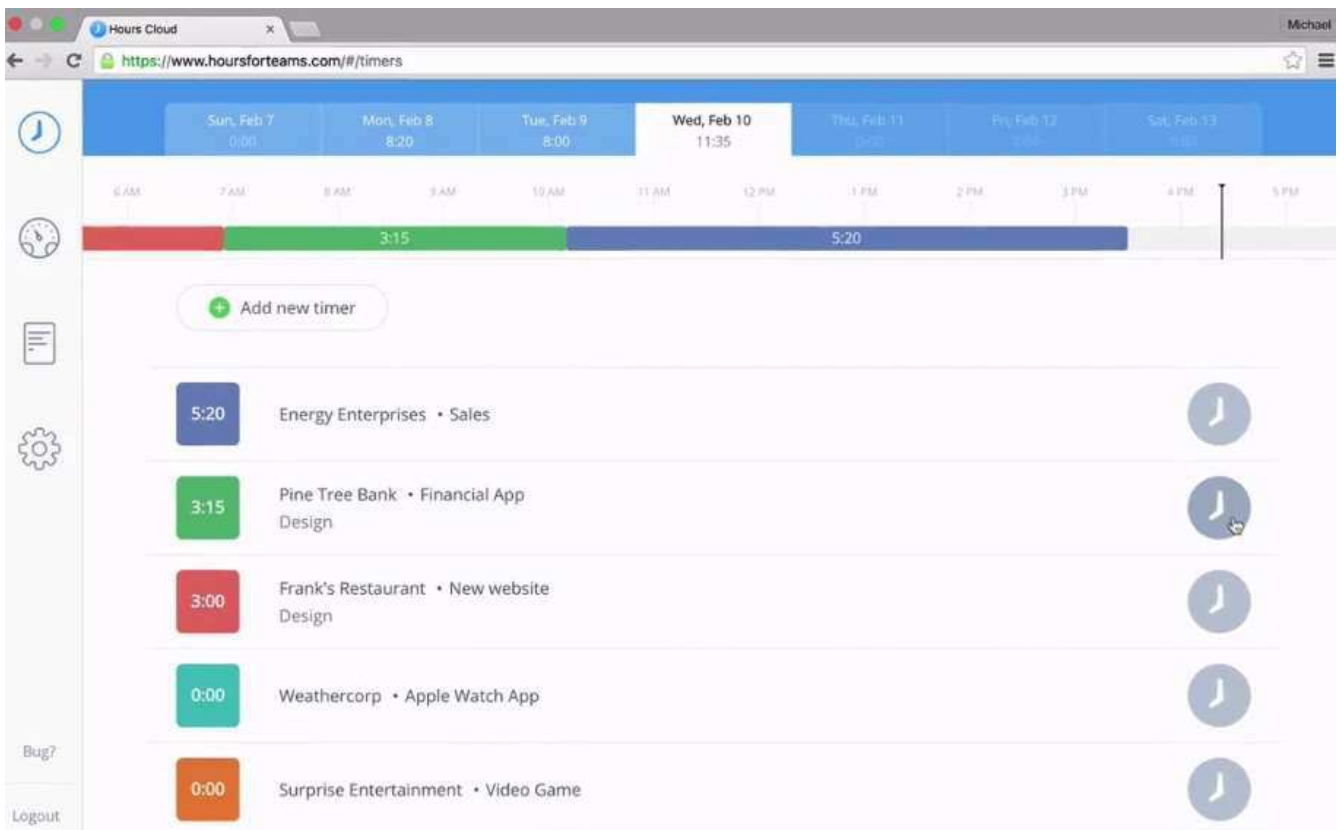


Рисунок 4 - Внешний вид Hours

1.4.4 Timely

Timely предназначен как для отдельных пользователей, так и команд, которым нужен специализированный тайм-менеджер [6]. Данный тайм-менеджер направлен на бизнес сегмент.

Уникальная особенность Timely – автоматический тайм-трекинг. Записывается любая активность, в зависимости от прошлой истории того, что делал пользователь. Затем система сама подсказывает подходящий ярлык и теги.

Можно либо принять предложения Timely, вручную отредактировать их, либо отклонить. Всякий раз, когда вы вручную вписываете данные, вы улучшаете функцию самообучения Timely. И система генерирует более точные подсказки.

Однако, такой вариант подойдет не для всех, например, при работе пользователя с документами Google Docs искусственный интеллект сервиса не сможет подсказывать ярлыки или теги. При работе над разными задачами, автоматический тайм-трекинг будет вполне уместен.

К плюсам можно отнести функцию автоматического тайм-трекинга, которая экономит время и позволяет перепроверить, что важные даты не будут пропущены. Встроенная панель Project Health Dashboard помогает контролировать учетное рабочее время, когда нужно убедиться в том, что команда не выходит за рамки установленного бюджета. Панель управления для команды предоставляет важные данные относительно продуктивности.

Минусами данного тайм-менеджера являются цена и интерфейс. Внешний вид не позволяет просматривать выполненные задачи в компактном формате, только в качестве больших плиток. Timely не имеет бесплатной версии. Абсолютно все функции платные и требуют подписки. Автоматический трекинг может быть не столь эффективным, если работать с файлами в Google Docs и подобные. В данном случае придется каждый раз обучать нейронную сеть, что занимает время.

На рисунке 5 показан внешний вид Timely.

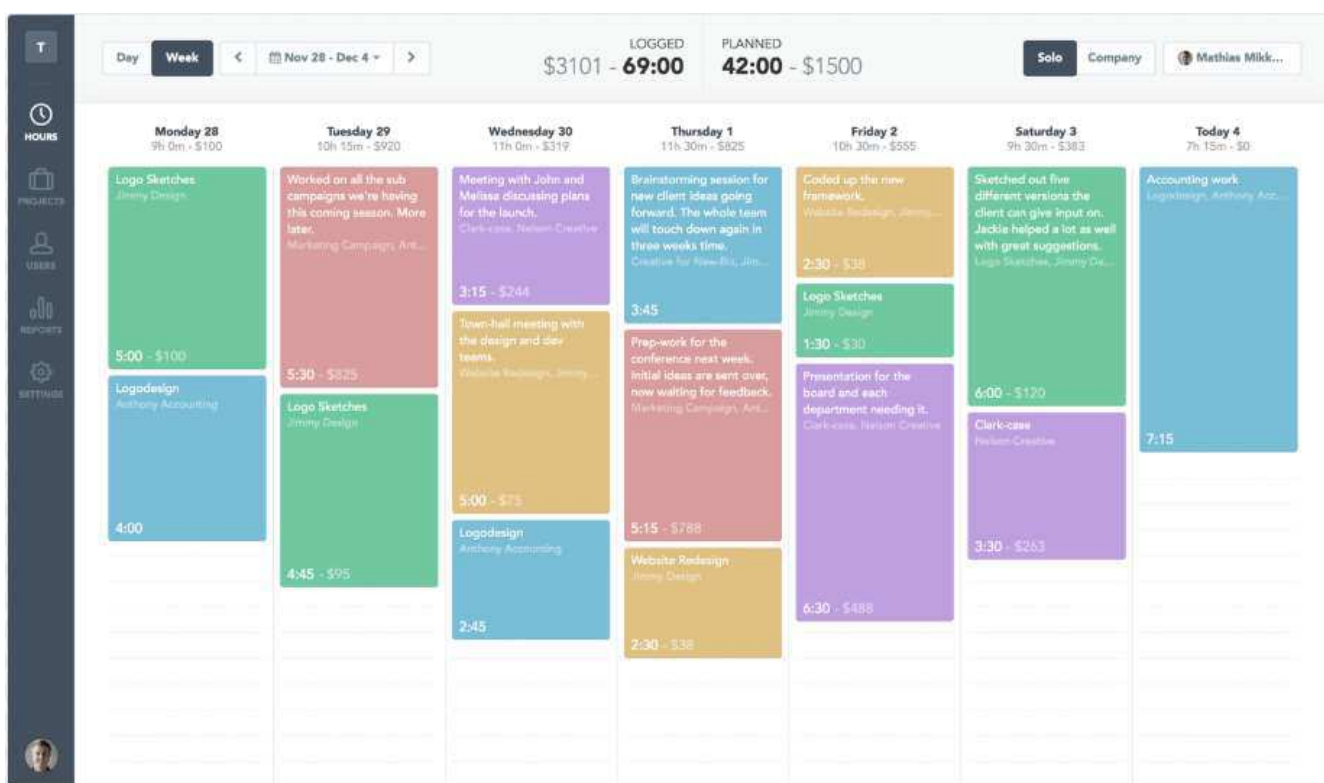


Рисунок 5 - Внешний вид Timely

1.5 Анализ рассмотренных ресурсов

В ходе обзора были проанализированы существующие программные продукты и составлена сравнительная таблица. Основным направлением сравнением аналогов является их направленность, внешний вид, список основных возможностей, стоимость. Почти все существующие тайм-менеджеры или имеют платные функции, без которых можно заниматься только самоконтролем, или полностью платные.

Таблица 1 – Сравнительная таблица аналогов

	Toggl	Harvest	Hours	Timely
Гибкость отчетов	-	+	-	+
Работа в команде	+	+	-	+
Удобный интерфейс	+	-	+	-
Наличие бесплатной версии	+	+	-	-
Веб версия	+	+	-	-
Мобильная версия сайта	+	+	-	-
Наличие графиков	+	-	-	-
Узконаправленный	-	+	-	+

Исходя из данных, отображенных в таблице 1, можно заметить, что универсальных продуктов нет, у каждого решения есть минусы. Больше всего удовлетворяют требованиям Toggl и Harvest. Timely очень узкоспециализированный и платный. Hours представляет собой простой тайм-трекер без бесплатной версии. Harvest направлен на бизнес-сегмент и имеет не очень удобный интерфейс, требующий множества действий для добавления и создания групп и задач. Toggl не имеет в бесплатной версии возможности работы с группами. Все продукты не являются полностью бесплатными.

1.6 Требования к разрабатываемому программному продукту

Проанализировав сильные и слабые стороны альтернативных тайм-менеджеров, было принято решение создать тайм-менеджер.

К программному продукту были выдвинуты следующие требования:

- удобный интерфейс;

- наличие отчетов;
- возможность исправлять для выполненных задач время начала и конца выполнения задачи;
- отсутствие лимитов на количество человек в группе;
- отсутствие лимитов на количество проектов для одного человека;
- программный продукт должен быть полностью бесплатным;
- возможность отправлять приглашения на вступление в группу пользователям;
- возможность выходить из группы;
- работа во всех современных браузерах под управлением таких операционных систем как Windows 7 и старше, Linux, MacOS.

2 Проектирование программного продукта

Проектирование позволяет на абстрактном уровне создавать структуру приложения, определиться с основными модулями и компонентами. От того, как будет спроектирован проект, зависит скорость разработки и сложность разработки, поэтому данный этап важен при разработке программных продуктов.

2.1 Описание структуры тайм-менеджера

Разрабатываемые тайм-менеджер и тайм-трекер представляют собой комплекс из четырех основных модулей.

Схема модулей представлена в таблице 2.

Таблица 2 - Модули разрабатываемого программного продукта

Название модуля	Функции
Модуль работы с задачами	Создание задач, редактирование названия, группы, времени начала и конца выполнения задачи, удаление
Модуль работы с графиками	Просмотр графиков индивидуальных и групповых
Модуль работы с группами	Создание групп, редактирование названия, цвета. Выход из группы. Отправка приглашений на вступление в группу, добавление модераторов. Детальный просмотр данных о группе, участниках и их задачах.
Модуль работы с персональными настройками	Изменение имени, смена пароля

Модули между собой обособлены и прямым образом друг на друга не влияют. Модуль работы с графиками использует данные, полученные в результате работы пользователя с модулем задачи. Их связь отображена в пункте 2.2. Модуль работы с задачами может использовать данные, полученные в результате работы пользователя с модулем группы. Связь между модулями есть, но слабая, поэтому такое разбиение структуры на модули является оптимальным. Каждый модуль является частью общей структуры без зависимости от любого другого модуля.

2.2 Логическая модель программного продукта

Разработка программного обеспечения для достижения положительных результатов требует применения специализированных методик управления, как самим процессом, так и всеми стадиями анализа, проектирования, реализации и последующего использования разработанного программного продукта. В качестве средств проектирования, моделирования, описания применяют специальные

графические языки. Самыми известными являются SADT, ERD, DFD, UML, ARIS, IDEF.

В данной работе были построены три диаграммы. Первая – диаграмма вариантов использования (рисунок 6). Данная диаграмма позволяет определить основные варианты работы с системой.

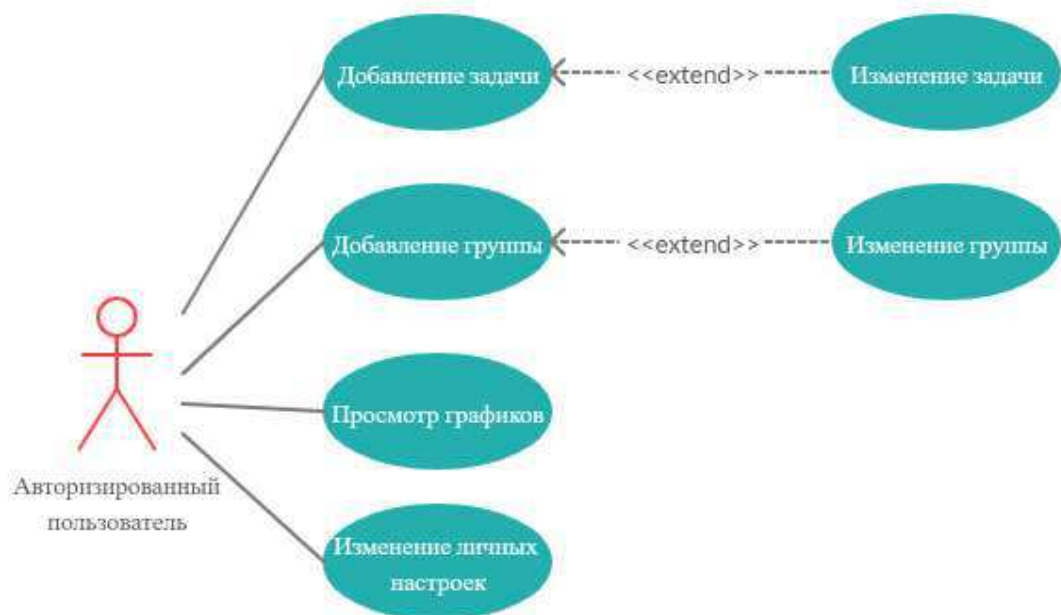


Рисунок 6 – UML-диаграмма работы пользователя с тайм-менеджером

Для описания связей между модулями используют языки моделирования потоков данных. В данной работе выбран язык DFD – Data Flow Diagrams – диаграмма потоков данных.

Потоки данных между модулями “Задачи” и “Графики” изображены на рисунке 7. Пользователь посылает запрос модулю “Задачи” на отображение данных о задачах, модуль посылает запрос в базу данных и получает данные, которые возвращает пользователю. Среди доступных действий с модулем “Задачи” присутствует изменение задач и добавление. Для визуализации задач

пользователь взаимодействует с модулем “Графики”. Пользователь посылает запросы на отображение графиков, модуль отправляет запрос в базу данных на получение нужных задач, после преобразует данные в диаграммы. Пользователь может выбирать типы диаграмм.



Рисунок 7 – DF-диаграмма модулей “Графики” и “Задачи”

2.3 Разработка структуры базы данных

Для организации работы программного продукта требуется создание базы данных. Для удобного представление структуры баз данных используют ER-диаграммы. На рисунке 8 изображена структура базы данных для хранения данных о пользователях, их задачах, группах и уведомлениях, помимо структуры в ER-диаграмме показаны связи.

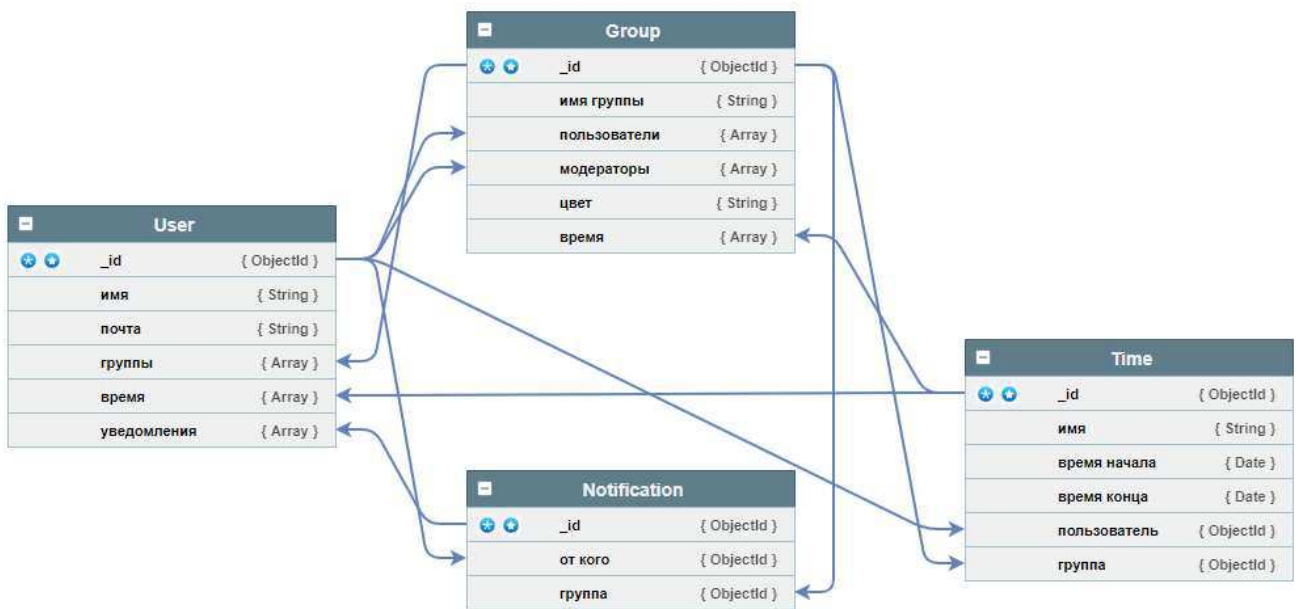


Рисунок 8 – ER-диаграмма базы данных

2.4 Выбор языка разработки

Перед разработкой программного продукта следует определиться с целевой платформой и языком программирования. По данным ресурса WebCanare [7] количество уникальных пользователей интернета на 2020 год составляет более 4.5 миллиарда пользователей (59% от всего населения планеты), в России интернетом пользуются более 118 миллионов людей (81% от всего населения страны). При этом прирост пользователей в России составляет 9 миллионов человек. Россияне сидят в интернете 7 часов 17 минут каждый день. При этом половина всего трафика приходится на мобильные устройства. По этим причинам целевой платформой для разработки программного продукта является интернет.

Сегодняшний рынок программного обеспечения для разработки приложений представляет широкий спектр языков программирования. По данным ресурса Stack Overflow [8] лидером среди языков веб-программирования является JavaScript - объектно-ориентированный язык, который отвечает за создание

интерактивных, динамических веб-страниц, включая такие функции, как анимация и масштабирование [9]. У JavaScript крупное сообщество и много поддержки, а также библиотек и фреймворков. JavaScript является языком №1 на GitHub по количеству pull request. Масса современных веб-сайтов используют JavaScript сегодня. При запуске JavaScript в браузере вы не загружаете никакое дополнительное программное обеспечение. А также язык очень доступный для новичков. Благодаря простоте и скорости многие стартапы и технологические компании сегодня используют JavaScript. JavaScript запускается в любой операционной системе с браузером, который его поддерживает.

Таким образом, языком программирования выбран JavaScript. Для данного языка существует множество различных фреймворков, позволяющих быстрее, проще, надежнее разрабатывать программы. Среди основных для frontend разработки выделяют три – Angular, React, Vue. За всеми этими фреймворками стоят огромные компании по типу Google или Facebook. Они поддерживают, разрабатывают и внедряют в свои продукты данные технологии. В данной работе выбор пал на Vue по причине работы с ним ранее.

Для серверной части программы решено также использовать JavaScript и его серверную реализацию – Node.js. Данное решение является стандартным и имеет название MEVN – акроним, содержащий в своем названии список основных используемых технологий:

- MongoDB – клиент-серверная нереляционная СУБД;
- Express – веб-фреймворк для создания API и веб-приложений;
- Vue.js – веб-фреймворк для создания клиентской части приложения;
- Node.js – фреймворк для создания серверной части приложения.

2.5 Выбор системы управления базой данных

Стек технологий MEVN предполагает использовать базу данных MongoDB. MongoDB позволяет быстро разрабатывать приложения благодаря своей структуре. У MongoDB структура данных основана на JSON-подобных документах. Данные многих веб-приложений представлены в виде JSON объектов, так как JSON является подмножеством синтаксиса языка программирования JavaScript. Хранение данных в виде ассоциированного массива легко сериализуется в JSON-документ. JSON используется при передаче данных между клиентом и сервером [10].

3 Программная реализация программного продукта

3.1 Создание базы данных

Структура базы данных представляет собой 4 схемы, которые в самой базе данных называются коллекциями (аналог таблицы в реляционных базах данных): Пользователь, Группа, Время, Уведомление. В таблицах 3-6 приведена логическая структура базы данных. В каждой таблице представлены имена полей, отражены их сущности и типы. Тип поля ObjectId представляет собой бинарную форму длиной в 12 байт, где первые четыре байта – значение времени в секундах с начала эпохи Unix, затем три байта являются идентификатором машины, после два байта состоят из идентификатора процесса, а последние три байта являются случайным значением счетчика.

Таблица 3 – Схема коллекции “Пользователь”

Наименования поля коллекции	Назначение	Тип поля
_id	Уникальный идентификатор записи	ObjectId
имя	Имя пользователя	Строковый
почта	Адрес электронной почты пользователя	Строковый
группы	Массив групп, в которых состоит пользователь	Массив ObjectId
время	Массив задач, которые принадлежат пользователю	Массив ObjectId
уведомления	Массив уведомлений пользователя	Массив ObjectId

Таблица 4 – Схема коллекции “Группа”

Наименования поля коллекции	Назначение	Тип поля
_id	Уникальный идентификатор записи	ObjectId
имя группы	Имя группы	Строковый
пользователи	Массив пользователей в группе	Массив ObjectId
модераторы	Массив модераторов в группе	Массив ObjectId
время	Массив задач, которые принадлежат пользователям в данной группе	Массив ObjectId
цвет	Цвет группы, представленный в виде трех пар шестнадцатеричных цифр	Строковый

Таблица 5 – Схема коллекции “Время”

Наименования поля коллекции	Назначение	Тип поля
_id	Уникальный идентификатор записи	ObjectId
имя	Имя задачи	Строковый
время начала	Время начала задачи	Дата
время конца	Время конца задачи	Дата
пользователь	Пользователь, которому принадлежит задача	ObjectId
группа	Группа, которой принадлежит задача	ObjectId

Таблица 6 – Схема коллекции “Уведомление”

Наименования поля коллекции	Назначение	Тип поля
_id	Уникальный идентификатор записи	ObjectId
от кого	Пользователь, от которого направлено уведомление	ObjectId
группа	Группа, от которой направлено уведомление	ObjectId

3.2 Клиентская часть

Клиентская часть нужна для отображения данных с сервера пользователю. Клиентская часть помогает пользователю взаимодействовать с тайм-менеджером. Вся клиентская часть написана на фреймворке Vue.js. Vue.js основан на шаблоне проектирования MVVM и позволяет связывать данные между собой. При изменении данных или при получении их с сервера, внешний вид мгновенно реагирует на новые данные. При загрузке страницы отправляется запрос на сервер, а сервер отправляет запрос в базу данных, после получения данных от базы данных сервер обрабатывает данные и отправляет их клиенту. Внешний вид

клиента при получении данных отображает их, но обновляет только ту часть, данные которой были обновлены. Вся страница не перезагружается и не перерисовывается.

Vue.js позволяет разрабатывать одностраничные приложения (SPA). Одностраничные приложения имеют одну общую HTML-страничку для всех веб-страниц. Взаимодействие с пользователем осуществляется с помощью динамически подгружаемых css, js html файлов.

Для навигации между различными представлениями (views) используется маршрутизация. Маршрутизация позволяет на каждый адрес сайта давать свое представление, которое будет отображено у пользователя динамически. С помощью маршрутизации можно ограничивать доступ к различным элементам приложения. Например, если пользователь не авторизован, то у него не будет возможности зайти в раздел с группами или графиками. Перед каждый переходом по новому адресу маршрутизация позволяет проверить нужные параметры и дать подходящий ответ.

Vue.js предоставляет разработчикам возможность иерархичного разделения структуры приложения. На каждый маршрут отображать какое-либо глобальное представление, само представление включает в себя переиспользуемые компоненты. Компоненты доступны для всех представлений. Например, в различных представлениях использовать одинаковые базовые кнопки, определенные в компоненте, но при этом в представлении можно поменять какие-нибудь настройки.

Для общения с сервером используется API в стиле REST. REST позволяет разрабатывать простые интерфейсы, в которые можно без проблем вносить изменения. В разрабатываемом программном продукте используется GET, POST, PUT, DELETE методы HTTP запросов. GET для получения данных, POST для отправки данных, PUT для изменения данных, DELETE для удаления данных. Использование привязки данных в данном случае позволяет при получении

данных не перезагружать всю страницу, а только обновить ту область, в которой данные отображаются.

3.3 Серверная часть

Сервер представляет собой прослойку между базой данных и клиентской частью. Клиентское и серверное приложения располагаются на разных ip серверах. Для обмена между клиентом и сервером используются HTTP запросы с различными методами. Для добавления данных от пользователя на сервер отправляется HTTP запрос с методом POST и телом сообщения, которое содержит данные, отправляемые пользователем. В случае успешного добавления данных в базу данных с сервера клиенту отправляется код состояния 200, уведомляющий о том, что данные успешно добавились. Если данные по каким-либо причинам не добавились, то клиенту отправляется код состояния 500, этот код предназначен для уведомления об ошибке сервера. Для получения данных с сервера пользователь отправляет HTTP запрос с методом GET. В данном случае сервер возвращает данные по указанному адресу в случае их наличия или возвращает код состояния 404 в том случае, если данные не найдены и код состояния 500, если произошла какая-либо ошибка на сервере. Для обновления данных используется HTTP запрос с методом PUT. PUT и POST похожи между собой, отличие между ними в разных предназначениях. POST обычно используется для добавления данных, PUT обычно используется для обновления уже существующих данных. Для удаления данных клиент посылает HTTP запрос с методом DELETE. При получении запроса сервер проверяет наличие обработки для данного запроса. Если такой обработчик найден, то будет производиться обработка данных, если не найден, то клиент получит сообщение об ошибке. Сервер общается с базой данных, получает данные и добавляет данные в базу, отправляет данные клиенту. При увеличении количества пользователей возрастают требования к серверу и к

базе данных. Node.js имеет встроенные средства для кластеризации и балансировки нагрузки. Кластеризация позволяет использовать мощности нескольких вычислительных узлов, балансировка помогает эффективно распределять ресурсы между вычислительными узлами. MongoDB имеет встроенные средства для сегментирования. Сегментирование позволяет распределить базу данных на несколько вычислительных узлов. Данные решения используются для масштабирования проектов, для увеличения производительности.

4 Описание результатов работы

При разработке интерфейса основное внимание уделялось простоте и интуитивной понятности.

4.1 Страницы регистрации и авторизации

Для того, чтобы однозначно идентифицировать пользователя, реализована система регистрации и авторизации. У неавторизованного пользователя доступа к функционалу тайм-менеджера нет. Для получения доступа нужно зарегистрироваться (рисунок 9) или авторизоваться под уже существующей учетной записью. При попытке зайти по любому другому адресу сайта, пользователь будет автоматически переброшен на авторизацию. При авторизации клиенту выдается jwt-токен, который хранится в localStorage. LocalStorage представляет собой хранилище в браузере, куда можно записывать какие-либо данные. Jwt-токен генерирует и хранит сервер. Клиент при любом обращении к серверу передает в заголовке запроса выданный токен, сервер проверяет правильность токена, а после работает с тем запросом, который направил пользователь. Если пользователь удалит в браузере в localStorage строку с

токеном, то веб-клиент не позволит пользоваться всеми функциями и перенаправит пользователя на страницу с авторизацией. Данное решение часто применяется при разработке приложений с помощью стека MEVN.

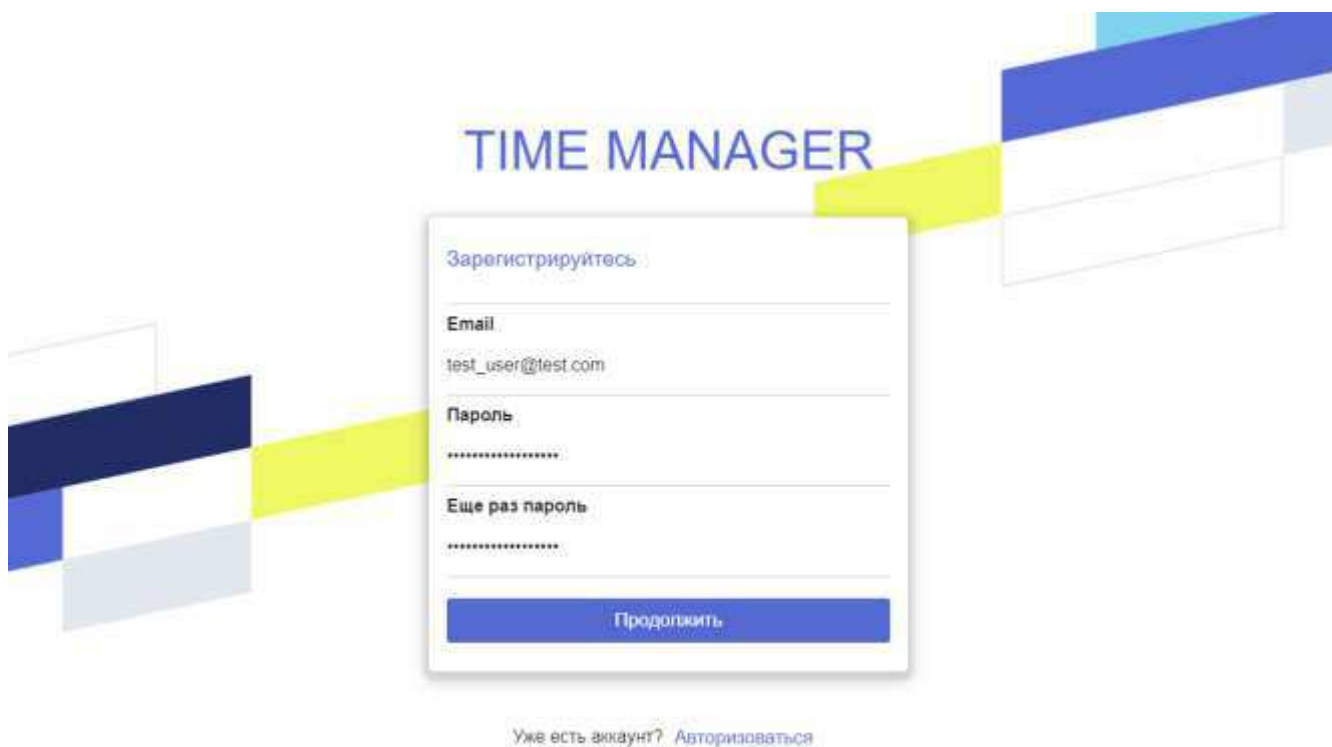


Рисунок 9 – Окно регистрации пользователя

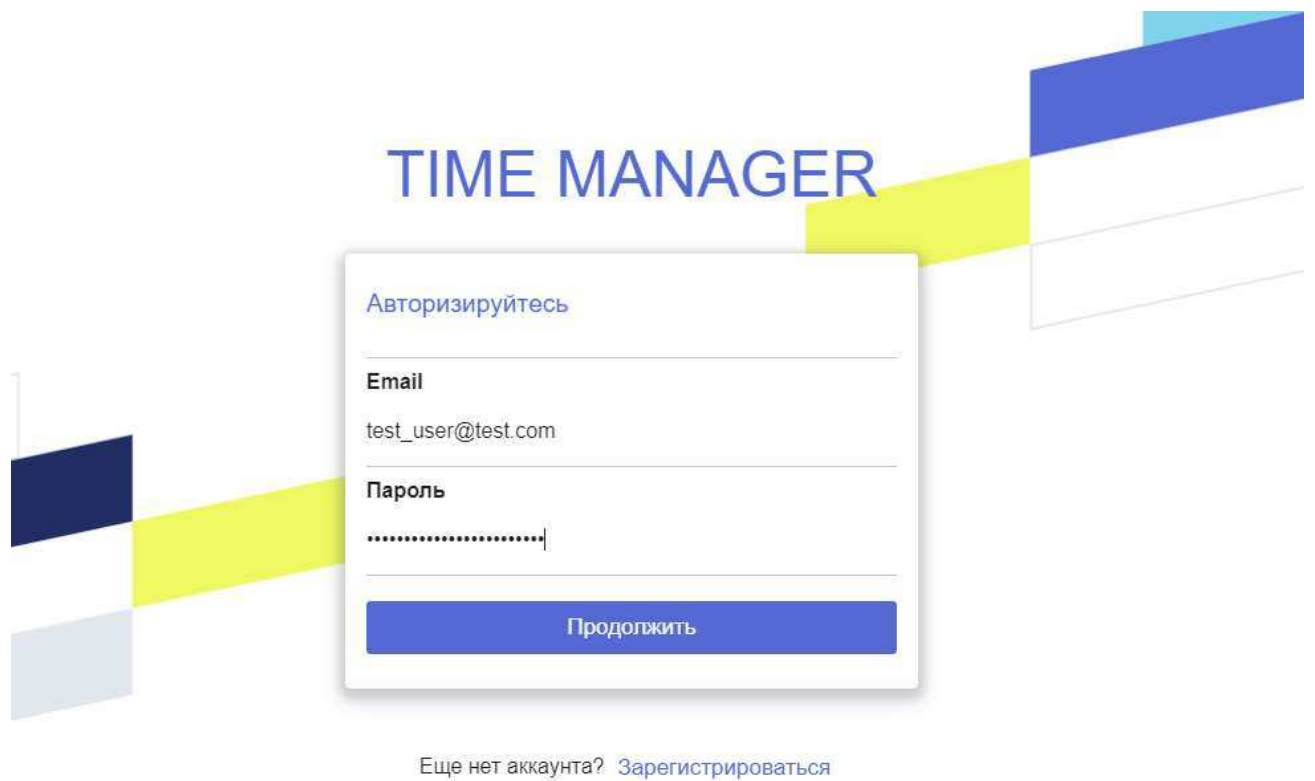


Рисунок 10 – Окно авторизации пользователя

4.2 Страница с задачами

После авторизации или регистрации для пользователя автоматически открывается главная страница тайм-менеджера. Главная страница (рисунок 11) разделена на три смысловых блока. В самом верху страницы располагается тайм-трекер (рисунок 12). Тайм-трекер позволяет давать задаче имя, выбирать группу, начинать задачу и заканчивать. При этом имя задачи и группа не являются обязательными и могут быть добавлены при желании после создания задачи.

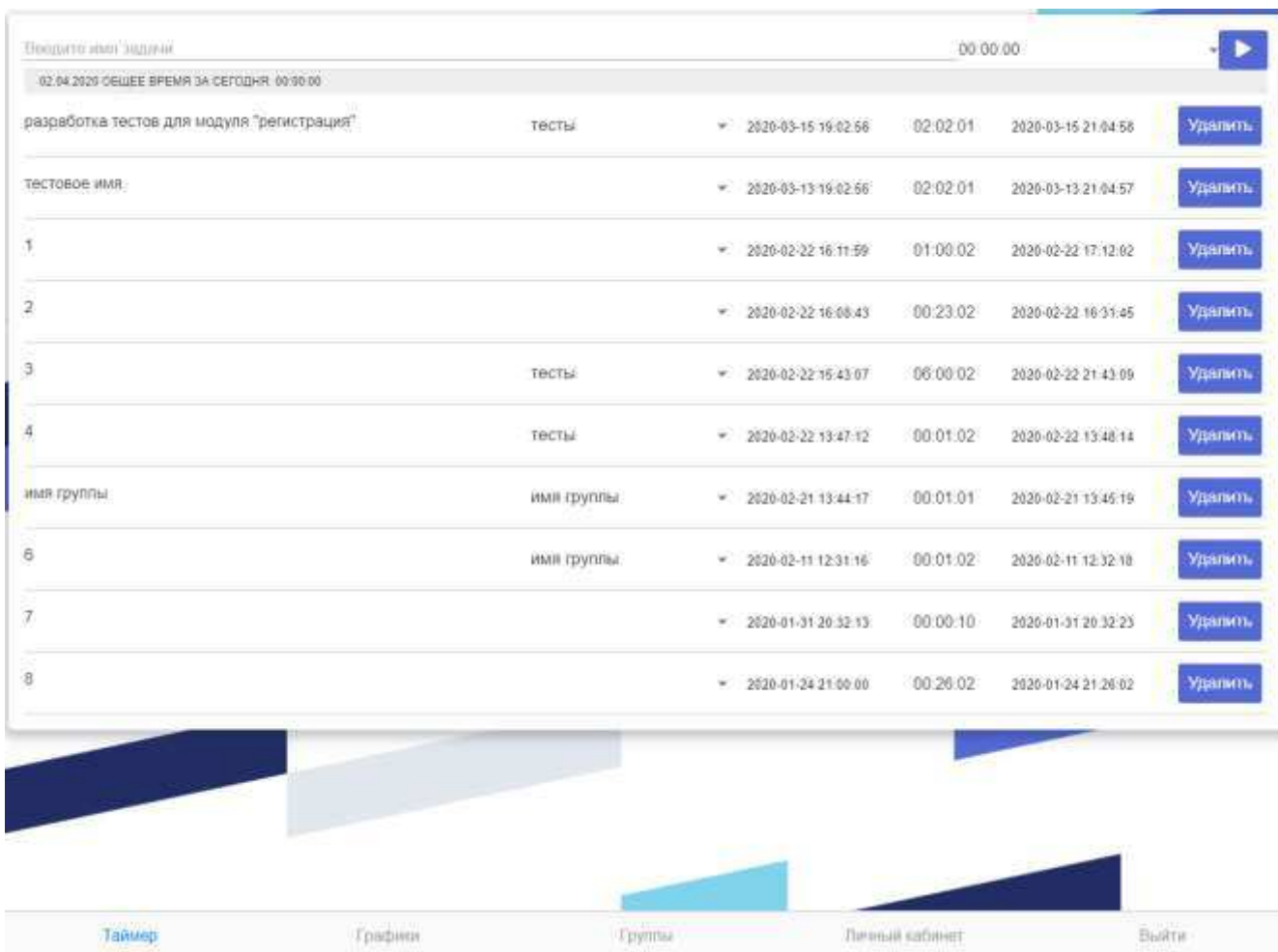


Рисунок 11 – Внешний вид страницы с задачами

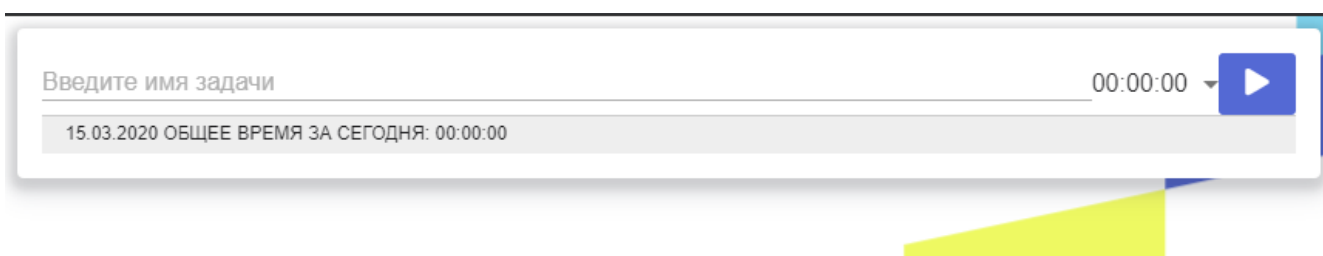


Рисунок 12 – Тайм-трекер пользователя

В самом низу страницы располагается меню (рисунок 13), с помощью которого можно быстро переходить в нужный раздел. Пять кнопок внизу расположены по причине популярности данной техники при проектировании интерфейсов.

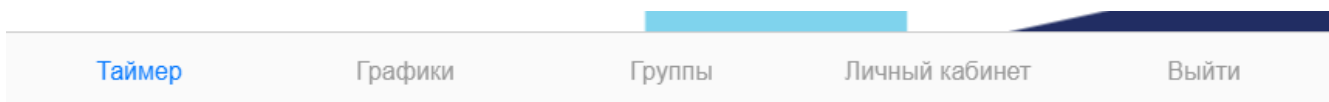


Рисунок 13 – Меню пользователя

Третьим функциональным блоком является список задач (рисунок 14). В этом блоке содержатся все задачи пользователя. Отображаются они в виде списка, самые поздние задачи выше в списке. Максимальное количество задач на одной странице – 12, затем появляется разделение на страницы, между которыми можно переключаться и просматривать более ранние задачи.



Рисунок 14 – Список задач

У каждой задачи имеется возможность задания имени. Если требуется изменить имя, то его можно легко изменить, просто введя новое (рисунок 15). Все изменения сохраняются автоматически.

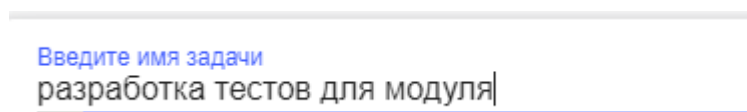


Рисунок 15 – Изменение имени

Изменение имени не единственное, что можно поменять у задачи. Изменение время начала (рисунок 16) и время конца доступно для любой задачи. Это сделано для тех случаев, когда нужно скорректировать временной интервал. Сначала выбирается дата, затем выбирается время. Следует отметить, что задаче нельзя задать время начала позже времени конца, как и время конца нельзя

определить раньше времени начала. Задаче можно изменить группу, задачу можно удалить.

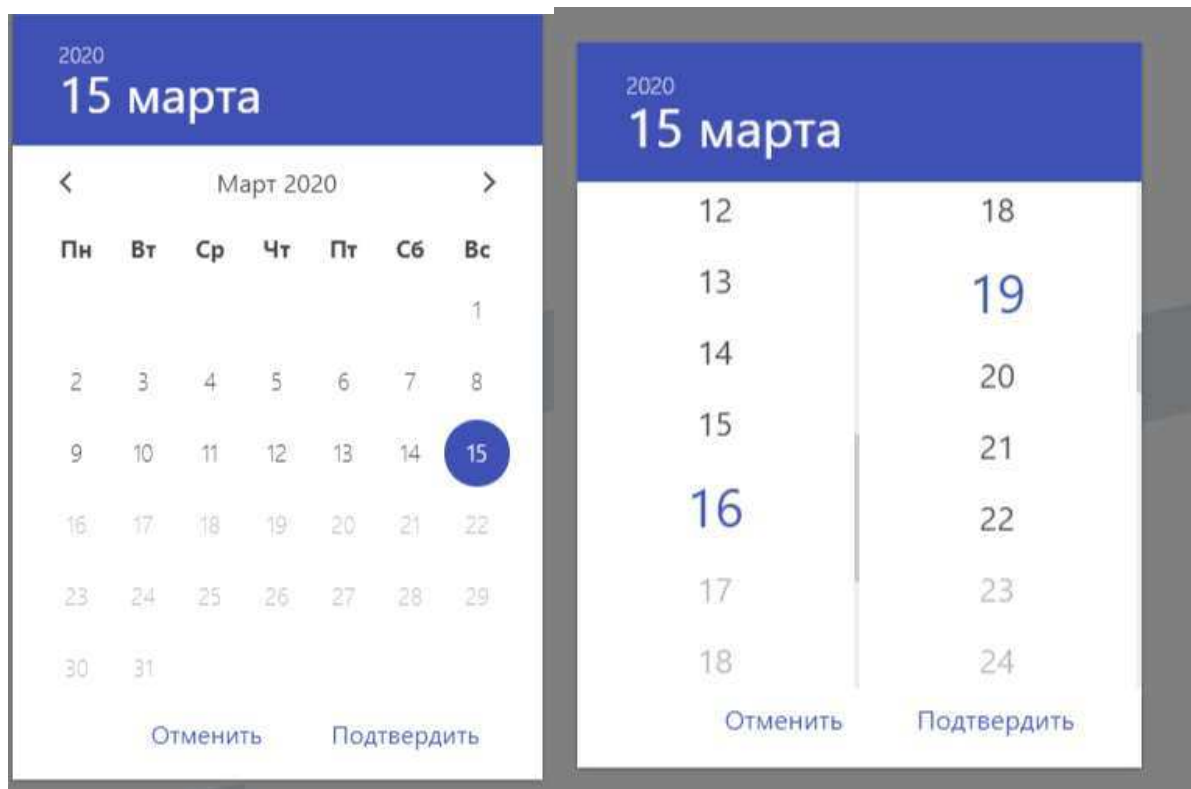


Рисунок 16 – Выбор даты и времени

На рисунке 17 изображена часть задачи, отображающая время начала и конца, а также разницу в часах, минутах и секундах между концом и началом задачи.

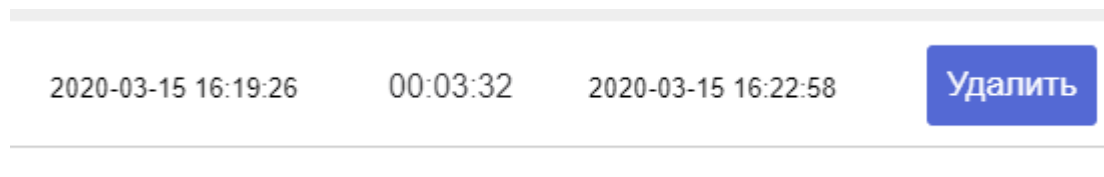


Рисунок 17 – Время задачи

4.3 Страница с группами

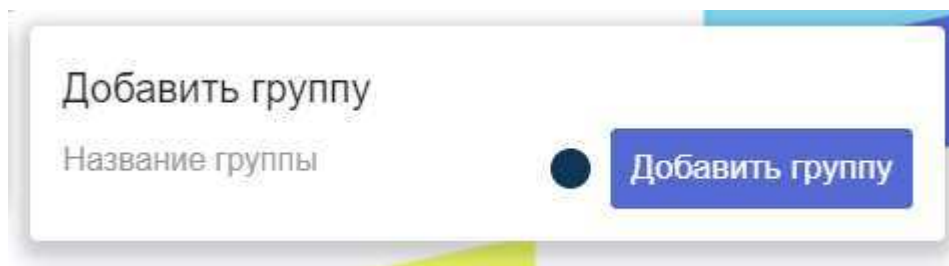
Помимо страницы с тайм-менеджером, авторизованному пользователю доступна страница с группами. Внешний вид страницы показан на рисунке 18. Суть данной страницы в создании и управлении группами. Каждая группа представляет какое-то абстрактное объединение. Например, под группой может подразумеваться проект в какой-либо организации. Или под группой определяют разделение времени на учебу и домашние дела. Страница с группами позволяет посмотреть количество человек в каждой группе, посмотреть общее время работы в группе, создать группу и просмотреть приглашения в другие группы.



На главную		
● тесты	2	09:15:36
● проект "кубок"	1	
● тесты "кубок"	1	

Рисунок 18 – Внешний вид страницы с группами

Интерфейс создания группы показан на рисунке 19. Для группы вводится имя и выбирается цвет. После чего следует нажать кнопку, и группа будет создана.



Добавить группу

Название группы ●

Рисунок 19 – Интерфейс добавления группы

Задачам можно добавлять группы. Так, на рисунке 20 показана задача, для которой выбрали группу “тесты” и цвет этой группы - желтый.

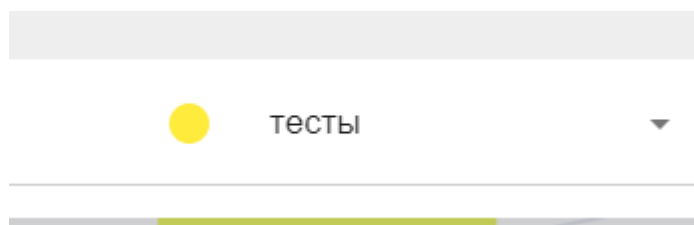


Рисунок 20 – Выбранная группа для задачи

Если нажать на группу в списке групп, то откроется страничка с редактированием группы (рисунок 21). Данный раздел позволяет просмотреть список всех пользователей в группе. Общее время всех пользователей данной группы отображается над списком пользователей. При нажатии на пользователя раскрывается выпадающий список с задачи пользователя для данной группы.

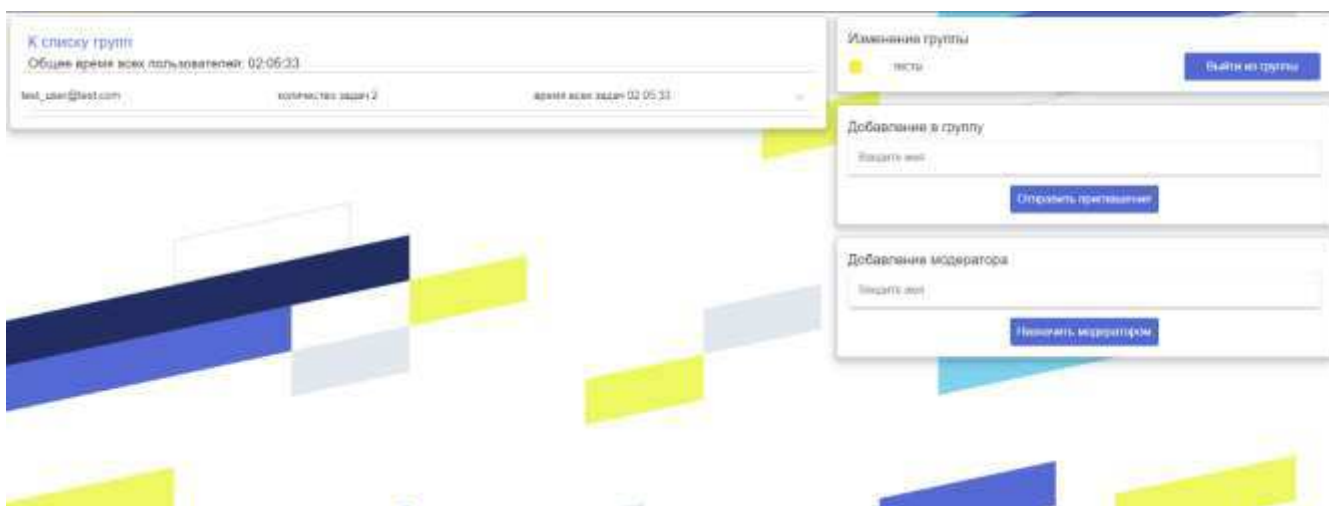


Рисунок 21 – Страница с редактированием группы

В группу можно добавлять других пользователей. Для этого следует вписать email пользователя и выбрать его, затем отправить приглашение. Внешний вид приглашения изображен на рисунке 22. Пользователь, которому направлено приглашение, может принять приглашение, после чего в его список групп будет добавлена данная группа, сам пользователь добавится к списку участников. Пользователь может отменить приглашение, тогда он не будет в списке пользователей у данной группы.

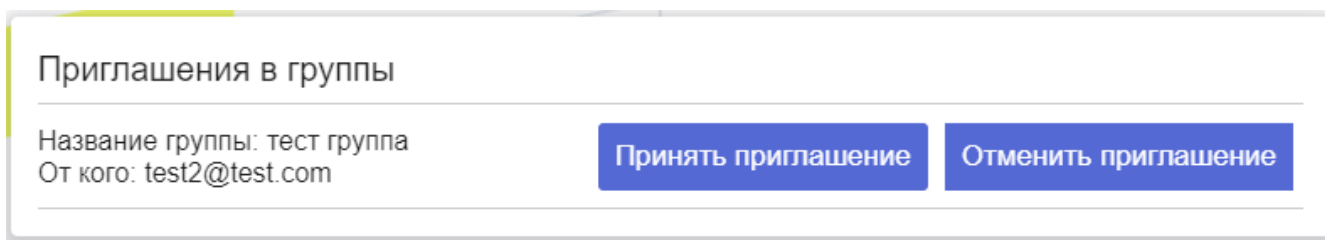


Рисунок 22 – Приглашения в группы

В группах имеется два типа пользователь: обычный и модератор. Обычный пользователь не может менять название и цвет группы (рисунок 23). Не может добавлять других модераторов. У обычного пользователя эти блоки отсутствуют. Модератор может поменять цвет и название группы. Добавлять других модераторов. Отправку приглашений в группу может осуществлять любой пользователь. Просмотр о задачах других пользователей также открыты для всех.

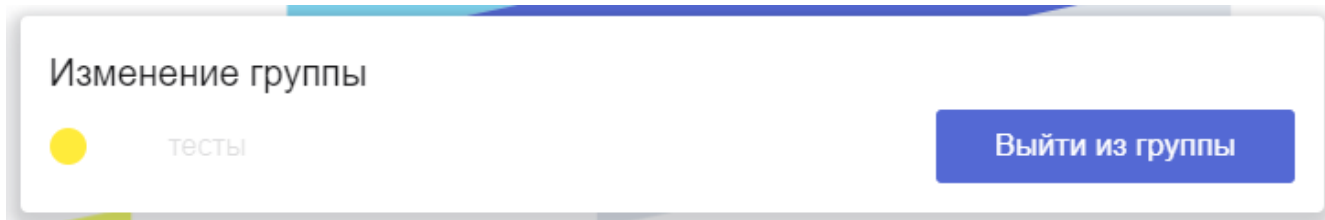


Рисунок 23 – Возможности обычного пользователя

4.4 Страница с графиками

Еще одним местом, куда может попасть авторизованный пользователь – страница с графиками. Графики помогают наглядно отследить затраченное время. Для графиков пользователя предлагается список из пяти видов (рисунок 24). Помимо просмотра графиков, присутствует иная возможность взаимодействия с данными. Пользователь может скачать нужный график в формате PDF. Кроме скачивания графика, можно работать с данными в текстовом виде в формате CSV, данный формат нужен для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми. Пользователь может как добавить свои данные о задачах, в данном случае сервер обработает полученный файл и добавит задачи в базу данных, так и скачать файл в формате CSV данные о своих задачах. Данная функция позволяет мигрировать от одного тайм-менеджера к другому. В разных тайм-менеджерах используются разные наименования столбцов, разное количество столбцов, разный формат времени для задач, по этой причине может возникнуть нужда в правке файлов перед миграцией от одного тайм-менеджера к другому. Так как CSV формат представляет собой строку текста, в которой для разделения полей используется запятая, то пользователь может скорректировать данные, полученные путем экспорта из тайм-менеджеров в табличный вид или привести к такому виду, который понимает другой тайм-менеджер.

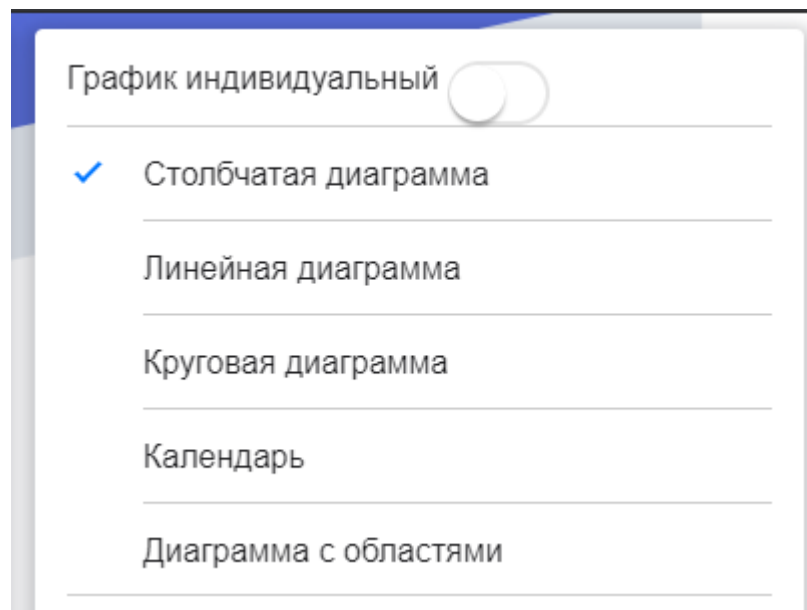


Рисунок 24 – Список доступных графиков

Первым среди графиков является столбчатая диаграмма (рисунок 25), представляющая собой столбцы одинаковой ширины с различной высотой от количества времени. Столбцов всегда 7 – по количеству дней в недели. По горизонтальной оси располагаются дни недели и дата в виде дней месяца. По вертикальной оси отображается время в часах. Высота столбцов формируется в процентах от максимального значения за неделю. Если задачи располагаются в разных неделях, месяцах, годах, то появляются кнопки управления, с помощью которых можно переключаться на другие недели. При переключении недели без задач пропускаются, таким образом можно быстро отследить только те недели, когда была какая-либо активность.

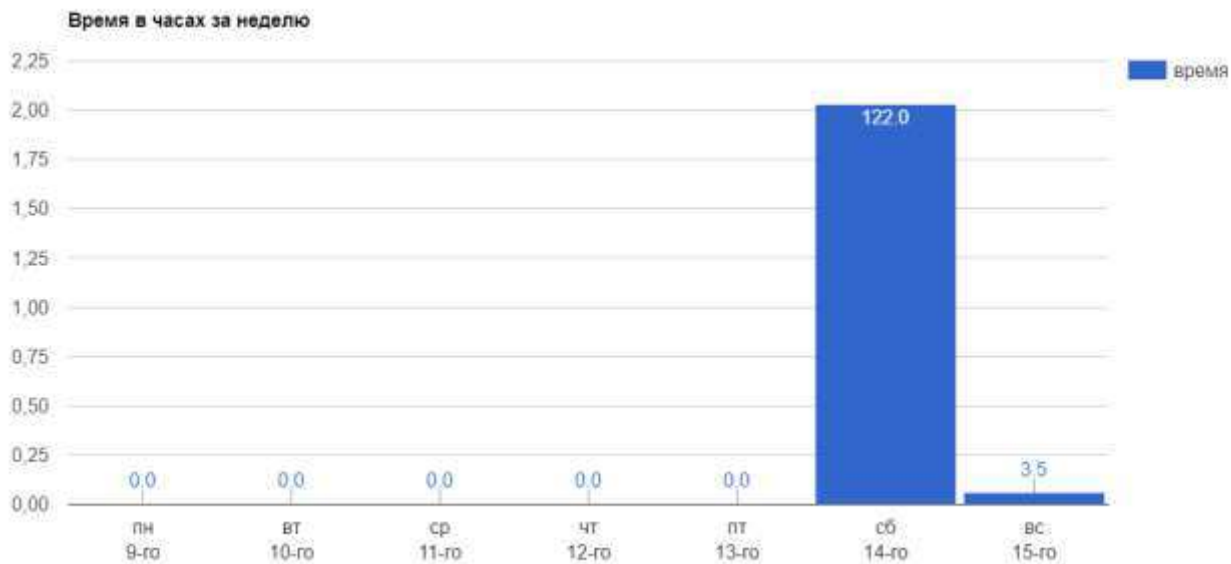


Рисунок 25 – Столбчатая диаграмма

Вторым типом графиков является линейная диаграмма (рисунок 26). Она во многом похожа на столбчатую диаграмму, но линейная диаграмма показывает динамику изменения, а не просто численные значения.

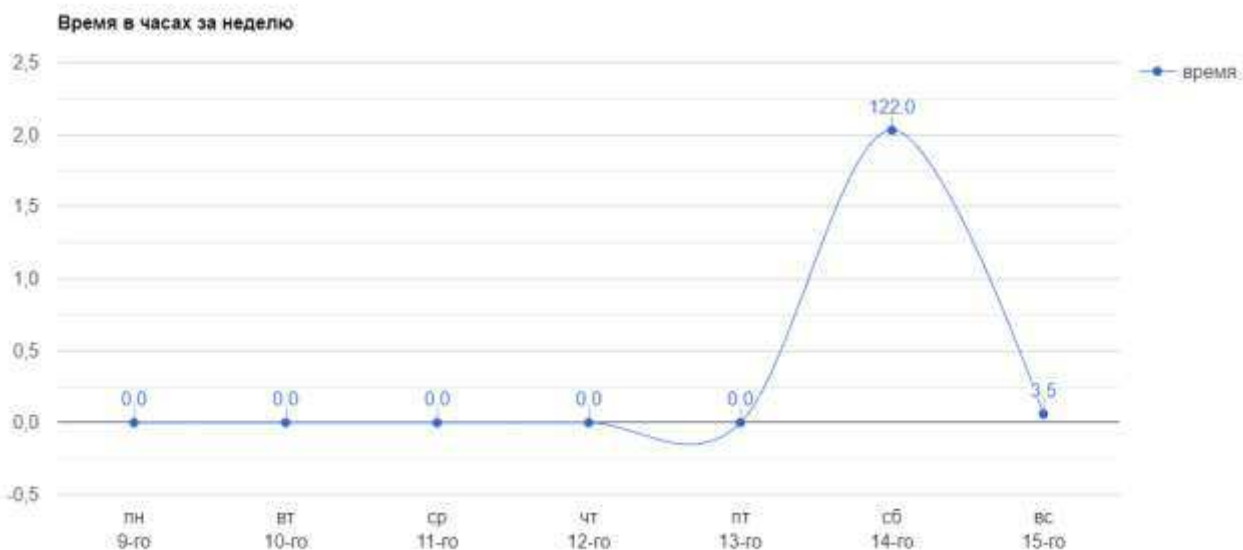


Рисунок 26 – Линейная диаграмма

Третий тип графиков – круговая диаграмма (рисунок 27). Круговые диаграммы позволяют отобразить вклад каждого дня относительно общего времени за неделю. Каждый день окрашивается в свой цвет и площадь сектора каждого дня пропорциональна времени за неделю. Позволяет наглядно увидеть разницу по дням и определить узкие места в работе. На графике отображаются только те дни, в которые была какая-либо активность, так как в противном случае площадь сектора была бы равна 0 и не отображалась бы на графике.

Время в часах за неделю

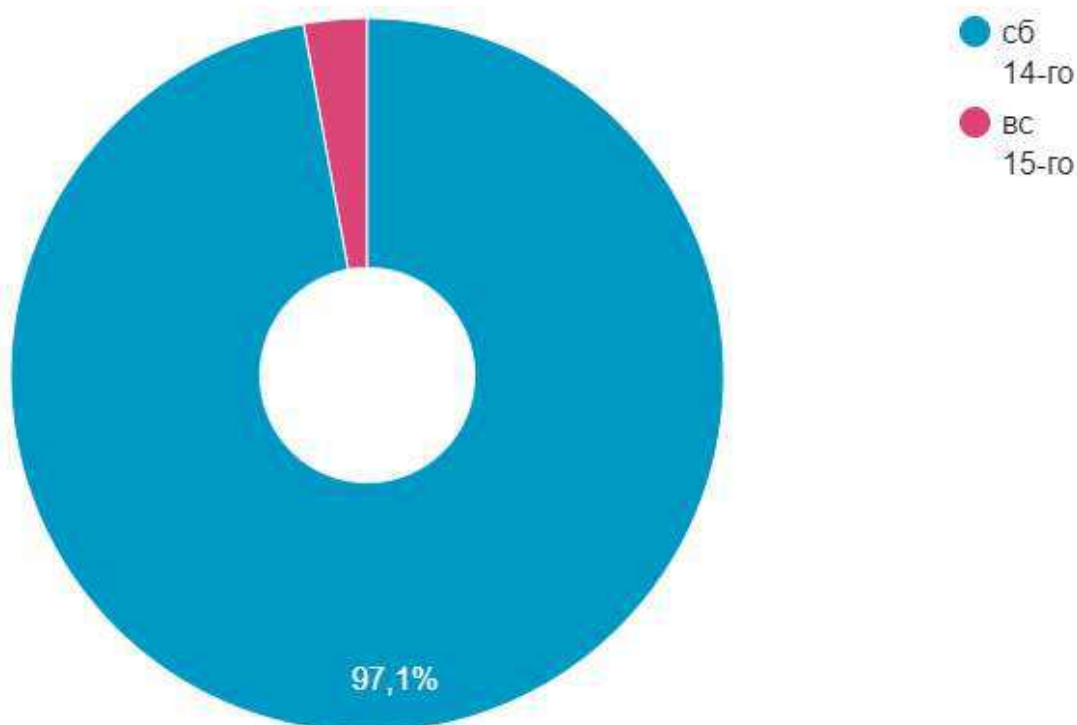


Рисунок 27 – Круговая диаграмма

Следующим типом графиков является календарь (рисунок 28). Данный тип графиков отображает сразу весь год, строками которого являются дни недели. В легенде диаграммы показан цветовой градиент. От белого к синему, где белый –

минимальное значение, а синий – максимальное. Каждый день в году окрашивается в свой цвет. Чем больше активности, тем темнее цвет. Если в какой-либо день не было активности, то этот день окрашивается в серый штрих. Данный график позволяет оценить продуктивность за целый год.

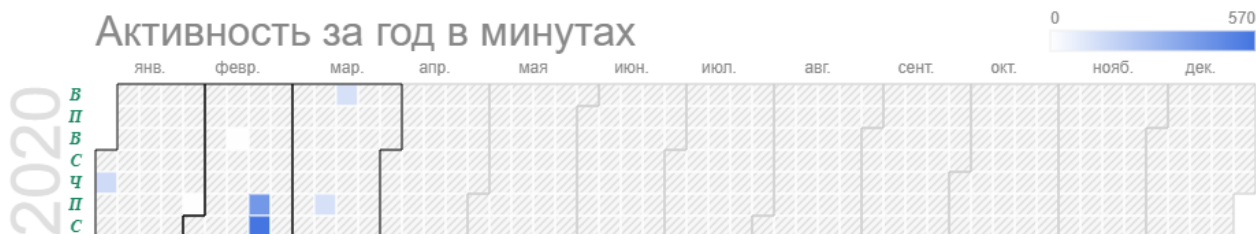


Рисунок 28 – Календарь

Последний в списке, но не по значению тип графиков – диаграмма с областями (рисунок 29). Диаграмма позволяет отследить тенденцию по времени за неделю.

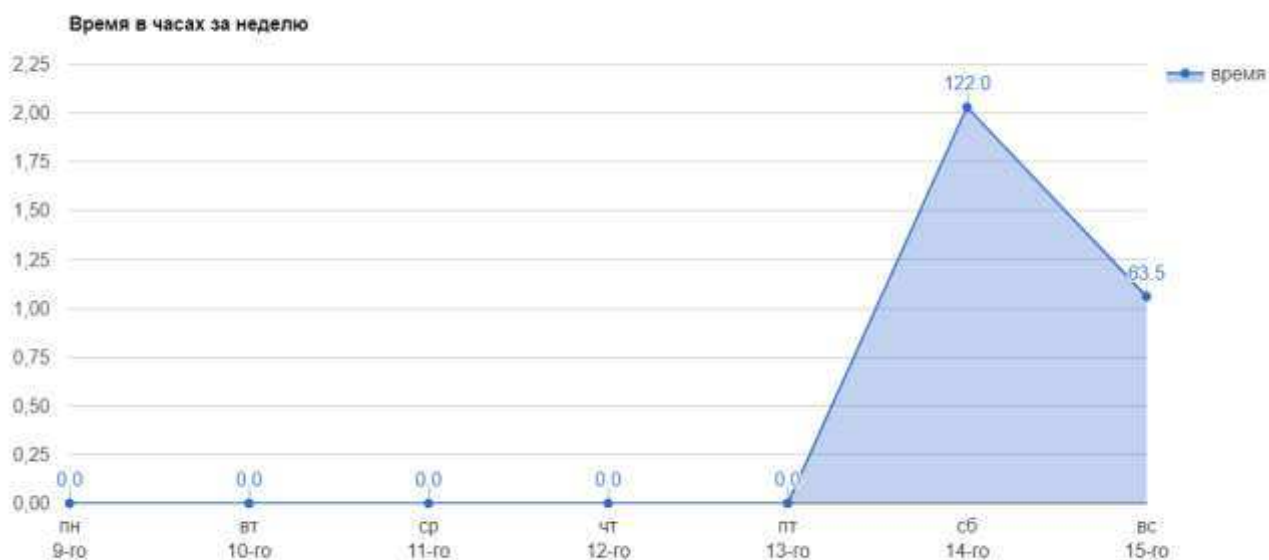


Рисунок 29 – Диаграмма с областями

Параллельно с графиками для одного пользователя существуют графики для групп. Их список ограничен тремя вариантами: столбчатая диаграмма, линейная и диаграмма с областями (рисунок 30). Для выбора группы присутствует поле ввода, где следует вписать нужную группу. Групповые графики позволяют отследить вклад каждого участника.

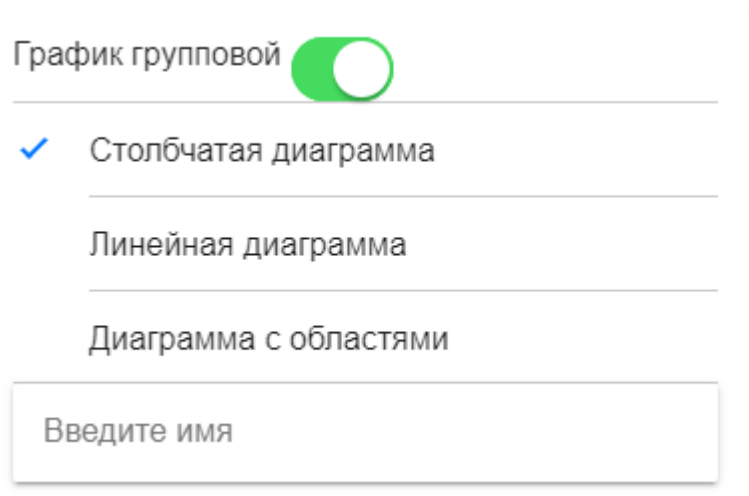


Рисунок 30 – Список графиков для группы

Первым среди графиков для групп является столбчатая диаграмма (рисунок 31). Отличительной особенностью является группировка столбцов по числам. В каждый день столбец каждого пользователя отображает свою информацию. Если пользователь в какой-то день не имеет задач, то его столбец будет отсутствовать. Данная диаграмма позволяет отследить занятость каждого пользователя в каждый день недели. К данной диаграмме также доступны кнопки перелистывания недель при их наличии. С помощью них можно быстро просмотреть статистику по каждому пользователю по неделям. График с легкостью масштабируется до 10 человек. Ограничений на количество пользователей нет, но при большем количестве человек наглядность становится неудобной. В данном случае рекомендуется разбиение группы на подгруппы.

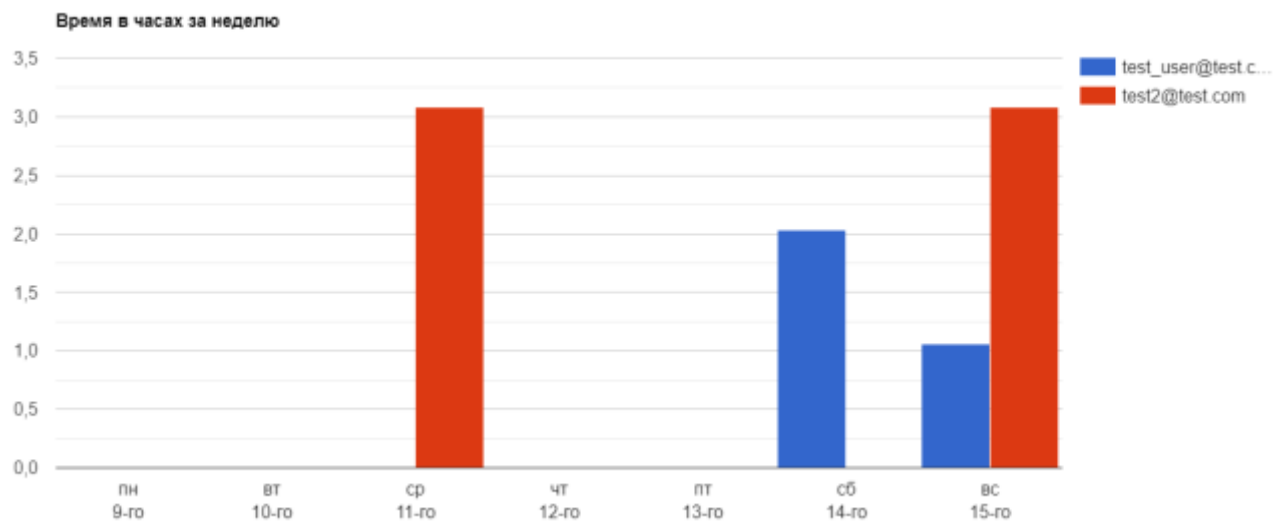


Рисунок 31 – Групповая столбчатая диаграмма

Вторым типом графиков для групп является линейный график (рисунок 32). Данный тип графика отражает динамику изменения количества времени по дням.

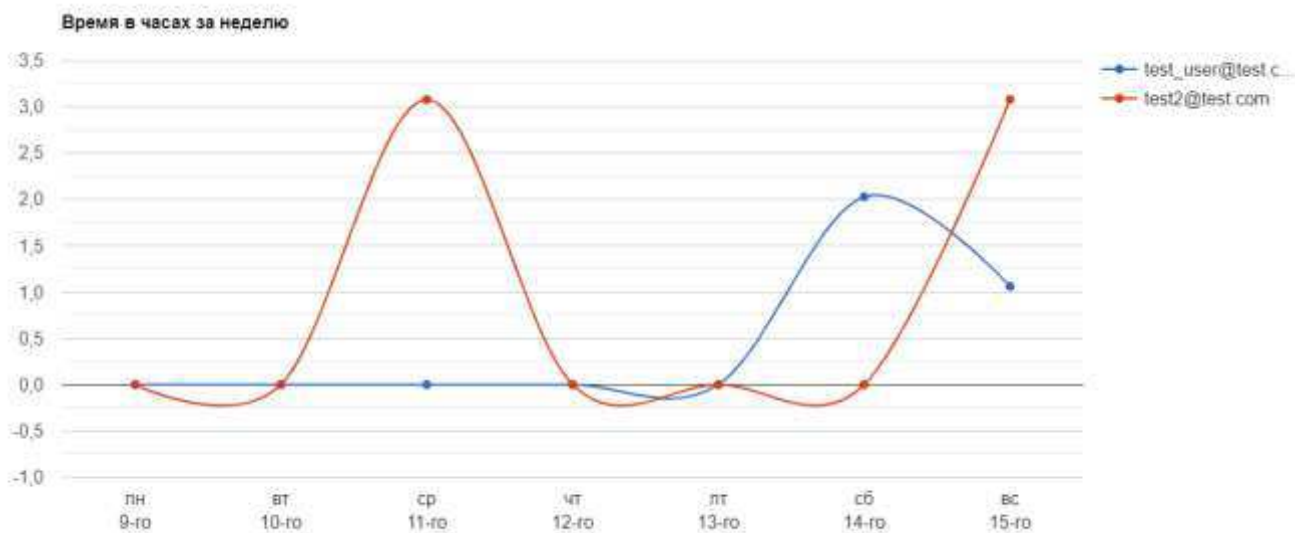


Рисунок 32 – Групповая линейная диаграмма

Последним типом графиков является диаграмма с областями (рисунок 33). Данная диаграмма показывает сумму по времени всех задач всех пользователей группы в данный день. Таким образом можно оценить вклад и занятость каждого пользователя в конкретный день.

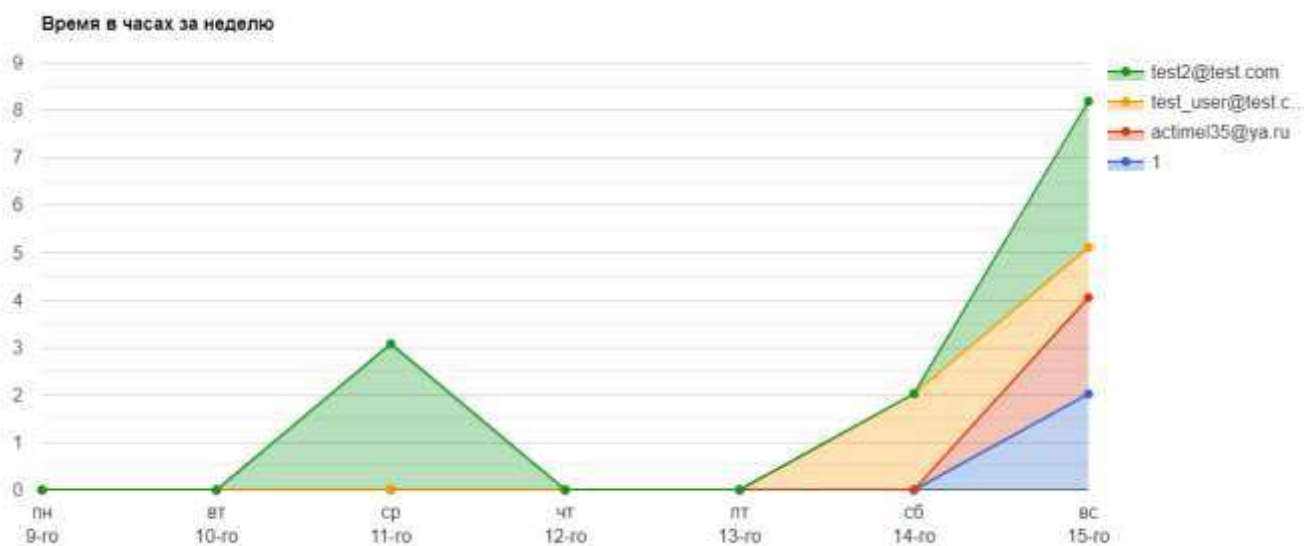


Рисунок 33 – Групповая диаграмма с областями

ЗАКЛЮЧЕНИЕ

На начальном этапе разработки веб-приложения для тайм-менеджмента была проанализирована необходимость разработки данного программного продукта. Были проанализированы существующие аналоги, в результате анализа разработан список требований к разрабатываемому программному продукту.

На следующем этапе разрабатывалась общая структура проекта и его основных модулей. Также была разработана структура базы данных, выбран язык программирования.

После была создана база данных, разработаны пользовательский интерфейс и сервер, общающийся с пользователем и базой данных.

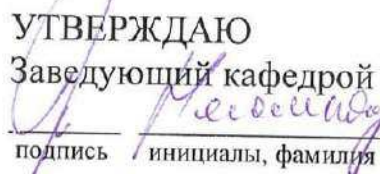
В результате написания выпускной квалификационной работы разработано веб-приложение для тайм-менеджмента. Данное приложение может быть полезно как при самоорганизации, так и при групповых работах, требующих отслеживания затраченного времени.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. STIMULAS – мотивация и развитие личности [Электронный ресурс]: Для чего нужен тайм-менеджмент? – Режим доступа: <http://stimulas.ru/timemanagement/dlya-chego-nuzhen-taym-menedzhment.html>.
2. Kickidler [Электронный ресурс]: Топ-8 лучших тайм-трекеров для учета рабочего времени, обзор и сравнение 2020 года. – Режим доступа: <https://www.kickidler.com/ru/for-it/methods-of-working/8-luchshix-tajm-trekerov-dlya-ucheta-rabochego-vremeni,-obzor-i-sravnenie-2020-goda.html>
3. Toggl – free time tracking software [Электронный ресурс]: – Режим доступа: <https://toggl.com/>
4. Harvest – simple online time tracking software [Электронный ресурс]: – Режим доступа: <https://www.getharvest.com/>
5. Hours – time tracking app online [Электронный ресурс]: – Режим доступа: <https://hourstimetracking.com/>
6. Timely appointment booking software [Электронный ресурс]: – Режим доступа: <https://www.gettimely.com/>
7. Webcanape [Электронный ресурс]: Вся статистика интернета на 2020 год — цифры и тренды в мире и в России – Режим доступа: <https://www.webcanape.ru/business/internet-2020-globalnaya-statistika-i-trendy/>
8. Stack overflow developer survey 2019 [Электронный ресурс]: Developer Survey Results 2019 – Режим доступа: <https://insights.stackoverflow.com/survey/2019#overview>
9. Proglib [Электронный ресурс]: 7 языков программирования, которые ты должен знать в 2020 году – Режим доступа: <https://proglib.io/p/7-yazykov-programmirovaniya-kotorye-ty-dolzhen-znat-v-2020-godu-2019-11-17>
10. Habr [Электронный ресурс]: MySQL и MongoDB — когда и что лучше использовать – Режим доступа: <https://habr.com/ru/post/322532/>

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

подпись / инициалы, фамилия
« ____ » ____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – "Информатика и вычислительная техника"
код – наименование направления

Веб-приложение для тайм-менеджмента
тема

Руководитель	 подпись, дата	доцент, канд. техн. наук должность, ученая степень	А. И. Постников инициалы, фамилия
Выпускник	 подпись, дата		М. М. Байгин инициалы, фамилия
Консультант	 подпись, дата	старший преподаватель должность, ученая степень	И. В. Матковский инициалы, фамилия
Нормоконтролер	 подпись, дата	доцент, канд. техн. наук должность, ученая степень	А. И. Постников инициалы, фамилия

Красноярск 2020