МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

ОТЧЕТ ПО КУРСОВОЙ РАБОТЕ

ТЕМА:

«Разработка программного приложения "Большой Брат" для учета рабочего времени»

Выполнил:

студент 189-1(1) гр.

Костерин Сергей Олегович

Руководитель:

Доцент к. м. и е. наук, к. т. наук

Карякин Иван Юрьевич

Оглавление

[Введение 3](#_Toc43475299)

[Описание работы приложения 5](#_Toc43475300)

[Перспективы развития приложения 9](#_Toc43475301)

[Список литературы 11](#_Toc43475302)

# Введение

Тенденция по постепенному переходу от традиционного формата работы в офисе к работе удаленно, хоть и была на слуху в течении относительно долгого времени, до недавних пор имела реальное практическое применение только в весьма редких случаях, тогда, когда кандидата не было возможности найти в интересующей нанимателя локации, либо в случае работ с низкой ответственностью и высокой заменимостью исполнителей, например копирайт или разработка программного обеспечения. Однако, на фоне глобальной пандемии загнанными в условия удаленной работы оказалось множество людей самых разных профессий, и по прогнозам экспертов, далеко не все компании в итоге вернутся к привычному ранее формату работы. Это означает, что рост спроса на программное обеспечение, помогающее более эффективно работать удаленно не только, не упадет после того, как мир победит коронавирус, но наоборот, продолжит расти. Итак, одной из наиболее важных для значительной части людей, вынужденных работать удаленно, является контроль рабочего времени, переработок и недоработок. Рассмотрим популярные решения, использующиеся для этого на рынке:

1. СБИС УЧЕТ И КОНТРОЛЬ РАБОЧЕГО ВРЕМЕНИ. Решение компании «Тензор» - лидера на рынке электронной отчетности и крупнейшего оператора ЭЦП в России. Этим решением автор данной работы пользуется уже больше года в рамках профессиональной деятельности. Для функционирования приложения необходимо установить плагин в браузер и клиент для ПК. Приложение собирает данные о запущенных на ПК приложениях, активных вкладках браузера. Стоимость решения – ориентировочно 125000 рублей в месяц за 100 сотрудников.
2. Kickidler. Приложение представляет собой несколько скрытых от пользователя клиентов, которые производят запись видео с экрана, нажатий клавиатуры и запущенных приложений, а также предоставляют администраторам возможность удаленного доступа, вид на экраны сотрудников в реальном времени. Стоимость решения – 50000 рублей в месяц за 100 сотрудников.
3. TimeCamp. Решение распространяется в виде web-сервиса и клиента для ПК. Собирает и предоставляет информацию об активности пользователя в системе и времени, проведенном у компьютера. Отдельно приложение предоставляет функционал для совместного проектного трекинга. Стоимость решения – 1000 долларов за 100 пользователей в месяц.

Следует отметить, что в данном отчете приведены далеко не все возможные аналоги, однако, проведя некоторое время в изысканиях, автор смог составить общую характеристику для программных решений, заполняющих рынок:

1. Высокая для малого бизнеса и тех, кто работает индивидуально цена: как правило, больше 7-10 долларов.
2. Закрытость исходного кода, невозможность самостоятельной модернизации.
3. Невозможность проведения интеграции или масштабирования своими силами.
4. Отсутствующая или крайне ограниченная функциональность для наглядной визуализации рабочей деятельности

Значит, особенностями разрабатываемого приложения будут являться:

1. Низкая цена разработки и внедрения, использование распространенных и легко масштабируемых инструментов
2. Открытый исходный код
3. Широкий функционал по отображению затраченного на рабочую деятельность времени, времени, во время работы проведенного в различных windows-приложениях, а также Work/Life баланса.

# Описание работы приложения

Проект состоит из трех частей: приложение для операционной системы windows, занимающееся сбором данных о пользовательской активности, веб-приложение, использующееся для представления пользовательской информации, и располагающегося в облаке серверного приложения. Приложение для windows было реализовано с использованием .Net framework 4.7.2, веб приложение – с использованием интерфейсного фреймворка Vue.js, библиотеки готовых компонентов Vuetify, и typescript в качестве скриптового языка, компилируемого в javascript. Располагающееся в облаке серверное приложение было реализовано с помощью сервиса Google Firebase.

При первой загрузке windows приложение проекта попросит пользователя зарегистрироваться или пройти аутентификацию после чего предоставит ему возможность начать вести учет своей деятельности за компьютером с использованием системы нажатием на кнопку «Начать работу». Приложение отслеживает открываемые пользователем приложения, и записывает как время общей активности в течение рабочего периода, так и время активности в каждом из приложений, что было открыто. Для того, чтобы остановить отслеживание рабочей деятельности, необходимо нажать на кнопку «Закончить работу», при этом, если пользователь не закроет приложение после нажатия на данную кнопку, он сможет продолжить работать в дальнейшем.

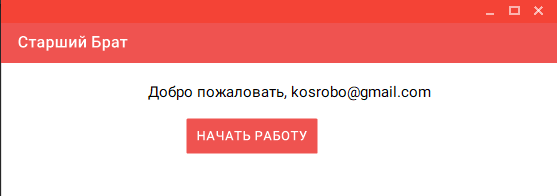


Рисунок 2 – «Основная форма windows-приложения»

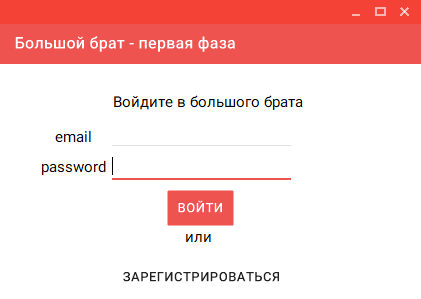


Рисунок 1 – «Форма входа/регистрации windows-приложения»

Данные об активности пользователей автоматически отправляются в облачную базу данных Firebase Realtime Database, являющуюся частью сервиса Google Firebase. Firebase Realtime Database – noSQL база данных, в основе которой – формат json. После загрузки json файла со структурой данных настройки доступа, Firebase Realtime Database генерирует REST API для взаимодействия с данными. За счет использования передовых технологий распределенного хранения Firebase Realtime Database значительно быстрее других noSQL баз данных, и, несомненно, гораздо удобнее в разработке, развертывании и масштабировании. Следует отметить, что Firebase Realtime Database не использовалась для хранения данных о пользователях в рамках данного проекта: для этого применялся сервис Firebase Authentication, предоставляющий быстрое и защищенное хранилище, оптимизированное под работу с регистрацией и аутентификаций пользователей.

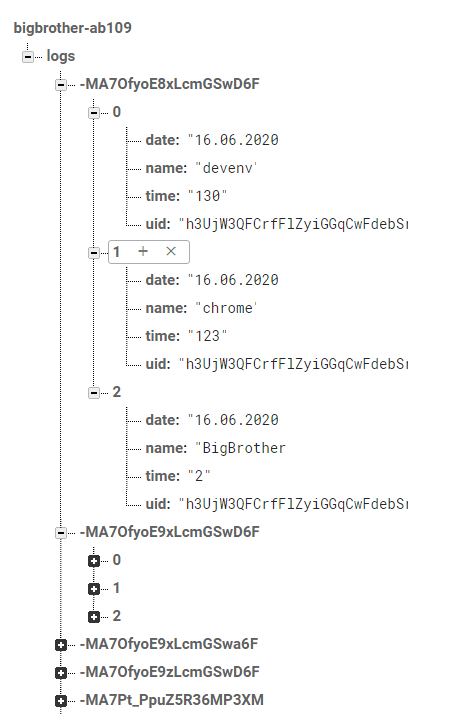


Рисунок 3 – «База данных проекта»

Следует отметить, что активность далеко не всех приложений может быть корректно захвачена: активность внутри приложений, запускаемых изнутри виртуальных машин, контейнеров docker, WSL, и иных средств виртуализации, будет распознана как активность непосредственно средств виртуализации. Справедливости ради, такой недостаток присущ всем рассмотренным во введении решениям.

При запуске web-составляющей проекта, пользователя встретит меню аутентификации, которое позволит зарегистрироваться или войти в систему. После аутентификации будет отображена главная страница проекта: т.н. дэшборд, предоставляющий собранную windows-приложением информацию в трех представлениях:

1. Время, потраченное за неделю. На диаграмме отображается количество времени, потраченное в течение каждого рабочего дня на протяжении последней недели.
2. Время по приложениям за день. На диаграмме отображается количество времени, проведенное пользователем в каждом из запущенным им приложений. День, по которому отображается информация, можно выбрать из выпадающего меню.
3. Баланс жизнь/работа. На диаграмме отображается соотношение времени, потраченного на работу в выбранный из выпадающего меню день, к времени, которое было проведено вне работы.

Возможность выйти из аккаунта предоставляет кнопка «Выйти», если пользователь не нажмет на нее, закрыв страницу, то при следующей загрузке будет аутентифицирован автоматически.

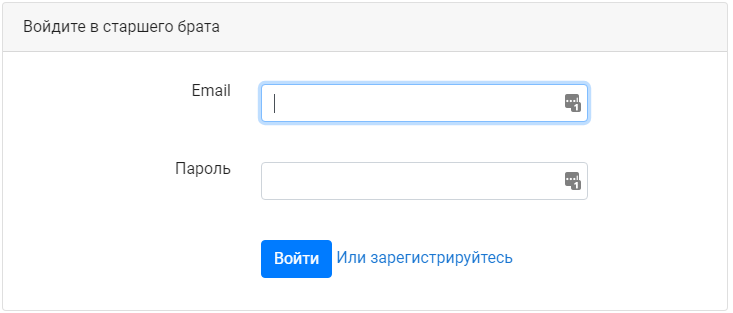


Рисунок 4 – «Форма для аутентификации в web-приложении»

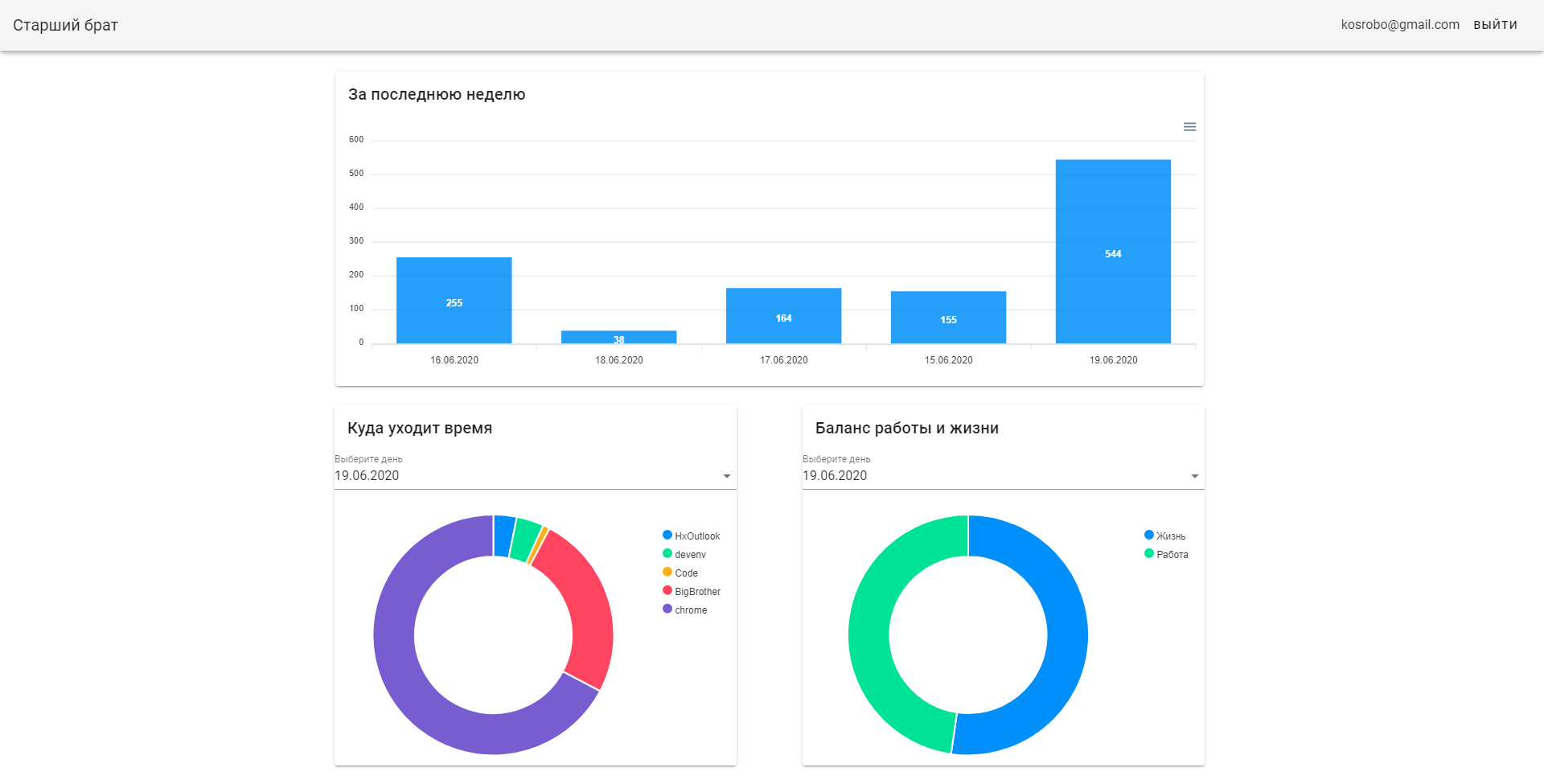


Рисунок 5 – «Основной экран web-приложения»

# Перспективы развития приложения

Так как приложение позиционируется, по понятным причинам, как некоммерческий open-source проект для энтузиастов и небольших компаний, желающих улучшить качество своей удаленной работы путем сбора и изучения статистики использования персонального компьютера, наиболее важным этапом в развитии приложения будет создание расширений для популярных браузеров, которые позволят анализировать не только запущенные локально приложения, но и открытые сайты. Следующей важной доработкой будет интеграция приложения с системами учета задач, имеющими открытое API, такими, как TodoIst, Jira, Trello. Это даст пользователям возможность отмечать, сколько времени они потратили на выполнение той или иной задачи, и на что они отвлекались в процессе. Другая важная веха в развитии продукта – предоставление пользователям возможности создавать черные и белые списки приложений, используемых для той или иной задачи. Автор работы считает должным заметить, что благодаря выбранному стеку технологий доработка приложения и его масштабирование значительно упрощены по сравнению с приложениями, использующими собственный веб-сервер и базу данных.

Таким образом, в результате курсовой работы было создано легковесное, легко дорабатываемое и масштабируемое решение, опирающееся на передовые облачные технологии и свободное программное обеспечение.

# Список литературы

1. Документация Vue,JS [Электронный ресурс]. – Режим доступа: <https://vuejs.org/v2/api/>, свободный. – (Дата обращения: 10.05.2020);
2. Документация Vuetify [Электронный ресурс]. – Режим доступа: [https://vuetifyjs.com/en/,](https://vuetifyjs.com/en/,%20) свободный. – (Дата обращения: 10.05.2020);
3. Документация Vuex [Электронный ресурс]. – Режим доступа: <https://vuex.vuejs.org/>, свободный. – (Дата обращения: 20.05.2020);
4. Документация Vue.Router [Электронный ресурс]. – Режим доступа: <https://router.vuejs.org>, свободный – (Дата обращения: 10.05.2020);
5. Документация Firebase [Электронный ресурс]. – Режим доступа: <https://firebase.google.com/docs/>, свободный – (Дата обращения: 13.05.2020);