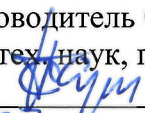


Министерство образования и науки Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт Прикладной Математики и Компьютерных Наук
Кафедра теоретических основ информатики (ТОИ)

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Руководитель ООП
д-р тех. наук, профессор
 С. П. Сущенко
« 07 » 06 2018 г.

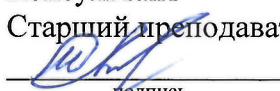
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

РАЗРАБОТКА КРОССПЛАТФОРМЕННОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ МОНИТОРИНГА РАБОТЫ СОТРУДНИКОВ

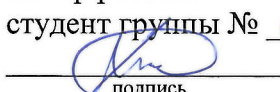
По основной образовательной программе подготовки бакалавров
Направление 02.03.03 - Математическое обеспечение и администрирование
информационных систем
Хованский Яков Александрович

Руководитель ВКР
Директор по развитию ООО «Редлайн»
 А.С. Передерей
подпись

« 7 » июня 2018 год

Консультант
Старший преподаватель НИ ТГУ
 М.С. Овсянников
подпись

« 7 » июня 2018 год

Автор работы
студент группы № 1443
 Я.А. Хованский
подпись

2018 год

АННОТАЦИЯ

Выпускная квалификационная работа, 55 с., 46 рис., 1 табл., 9 источников.

IONIC, ANGULARJS, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ,
КРОССПЛАТФОРМЕННОСТЬ, API, CSS, HTML

Цель работы – разработка кроссплатформенного мобильного приложения для мониторинга работы сотрудников.

Метод работы – теоретические исследования и практическая реализация.

Результат работы – разработано кроссплатформенное мобильное приложение для сервиса Dreel, которое внедрено в данный сервис.

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ	4
ВВЕДЕНИЕ	6
1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ	8
1.1. YAWARE TIME TRACKER	8
1.2. TIME DOCTOR.....	12
1.3. DREEL	16
1.4. СРАВНЕНИЕ СУЩЕСТВУЮЩИХ РЕШЕНИЙ	18
2. ТРЕБОВАНИЯ К СИСТЕМЕ	20
2.1. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	20
2.2. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	20
2.3. ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ.....	21
2.4. СЦЕНАРИИ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ	21
3. ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ.....	24
3.1. IONIC FRAMEWORK	24
3.2. WEBSTORM.....	26
3.3. CHART.JS	28
4. РАЗРАБОТКА ПРИЛОЖЕНИЯ.....	31
4.1. ПРОТОТИП ИНТЕРФЕЙСА	31
4.1. АРХИТЕКТУРА СИСТЕМЫ.....	32
4.3. ВЗАИМОДЕЙСТВИЕ С СЕРВЕРОМ.....	33
4.4. ПОДГОТОВКА ОКРУЖЕНИЯ	34
4.5. ВИЗУАЛЬНЫЕ КОМПОНЕНТЫ	35
4.6. СЕРВИСЫ.....	39
5. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	42
ЗАКЛЮЧЕНИЕ	52
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	53
ПРИЛОЖЕНИЕ А.....	54
ПРИЛОЖЕНИЕ Б.....	55

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

Гибридное приложение – это вебсайт встроенный в мобильное приложение, через так называемые webview. Они создаются с использованием HTML, CSS, и JavaScript, и выполняют один и тот же код, независимо от платформы, на которой они работают. При помощи фреймворков, возможно использовать нативные функции устройств, таких как камера или GPS.

SDK (от англ. software development kit) — набор средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ[1].

SASS (Syntactically Awesome Stylesheets) — модуль, включенный в Haml (язык разметки для упрощённой генерации HTML). Sass — это метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS кода и упрощения файлов каскадных таблиц стилей[1].

AngularJS – JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC-шаблона, а также упрощение тестирования и разработки.

Apache Cordova — мобильная среда разработки приложений с открытым исходным кодом. Apache Cordova позволяет программистам создавать приложения для мобильных устройств с помощью CSS3, HTML5 и JavaScript, вместо того, чтобы использовать конкретные платформы API, такие как Android, IOS или Windows Phone.

Рефакторинг (перепроектирование кода, переработка кода, равносильное преобразование алгоритмов) — процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы. В основе рефакторинга лежит последовательность небольших эквивалентных (то есть сохраняющих поведение) преобразований.

API (программный интерфейс приложения, интерфейс прикладного программирования) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах. Используется программистами при написании всевозможных приложений[1].

JSON (JavaScript Object Notation) - простой формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Он основан на подмножестве языка

программирования JavaScript, определенного в стандарте ECMA-262 3rd Edition - December 1999. JSON - текстовый формат, полностью независимый от языка реализации, но он использует соглашения, знакомые программистам C-подобных языков, таких как C, C++, C#, Java, JavaScript, Perl, Python и многих других. Эти свойства делают JSON идеальным языком обмена данными[1].

Angular CLI - интерфейс командной строки, устанавливающий необходимые зависимости и осуществляющий сборку проекта.

UI — это User Interface (дословно «пользовательский интерфейс») — то, как выглядит интерфейс и то, какие физические характеристики приобретает.

ВВЕДЕНИЕ

В современном мире уже трудно представить жизнь без мобильного телефона. Они стали неотъемлемой частью нашей жизни. Смартфоны с их огромным функционалом и возможностями делают нашу жизнь гораздо проще. А производители смартфонов и разработчики программного обеспечения всё чаще радуют нас новыми более мощными и функциональными устройствами. Теперь не обязательно идти в магазин за газетой, чтобы узнать новости, даже не обязательно вставать с постели, чтобы зайти в интернет, достаточно просто протянуть руку к смартфону и у тебя появится доступ к неограниченному объему информации.

В связи с популярностью смартфонов в наше время, разработка программного обеспечения для них, стала одним из востребованных направлений на рынке программного обеспечения. На данный момент на рынке существует огромное количество различных мобильных приложений, выполняющих совершенно разные функции. Перед разработчиками открываются огромные возможности для разработки своих приложений, ведь они получают доступ к различным функциям устройств, которых нет на компьютере, которые упрощают жизнь пользователя смартфона и делают её более интересной, такие как: гироскоп, компас, акселерометр и т.д. Одна только камера, установленная на мобильном устройстве, открывает огромный простор для фантазии разработчика, как и датчик GPS.

Разработка мобильного приложения не такая простая задача, так как приложение должно привлекать пользователя. Для этого надо продумать большое количество требований: дружелюбный и понятный интерфейс, быстроедействие и т.д.

На рынке существует класс программного обеспечения, который позволяет отслеживать рабочее время сотрудников. Представителями данного класса программного обеспечения являются Time Doctor, Yaware, Dreel.

Dreel – программа для контроля и анализа рабочего времени сотрудников, которым в настоящее время ежедневно пользуются десятки компаний России. В связи с большой конкуренцией в данном классе программного обеспечения, возникла необходимость создания мобильного приложения для сервиса Dreel, дающая возможность руководителю оперативно получать о работе своих сотрудников. Также после проведения опроса у пользователей данного сервиса, было выяснено, что потребность в мобильном приложении существует.

Чтобы повысить конкурентоспособность сервиса Dreel на рынке программного обеспечения и устранить данную проблему - отсутствие мобильного приложения, в рамках

данной выпускной квалификационной работы было решено разработать кроссплатформенное приложение для мониторинга сотрудников с помощью отчётов и анализа данных сотрудников для руководителей.

Основными задачами, поставленными в рамках данной выпускной квалификационной работы, являются:

- Анализ существующих решений
- Проектирование вариантов использования
- Разработка архитектуры приложения
- Проектирование интерфейса приложения
- Разработка приложения

1. Обзор существующих решений

1.1. Yaware TimeTracker

Yaware TimeTracker – автоматическая онлайн-система учета рабочего времени и оценки эффективности работы сотрудников за компьютером [4].

Возможности:

- Автоматический контроль рабочего времени – во сколько часов начинается и заканчивается рабочий день сотрудников, как часто они покидают свое рабочее время и сколько на самом деле они отрабатывают [4] (рисунок 1).

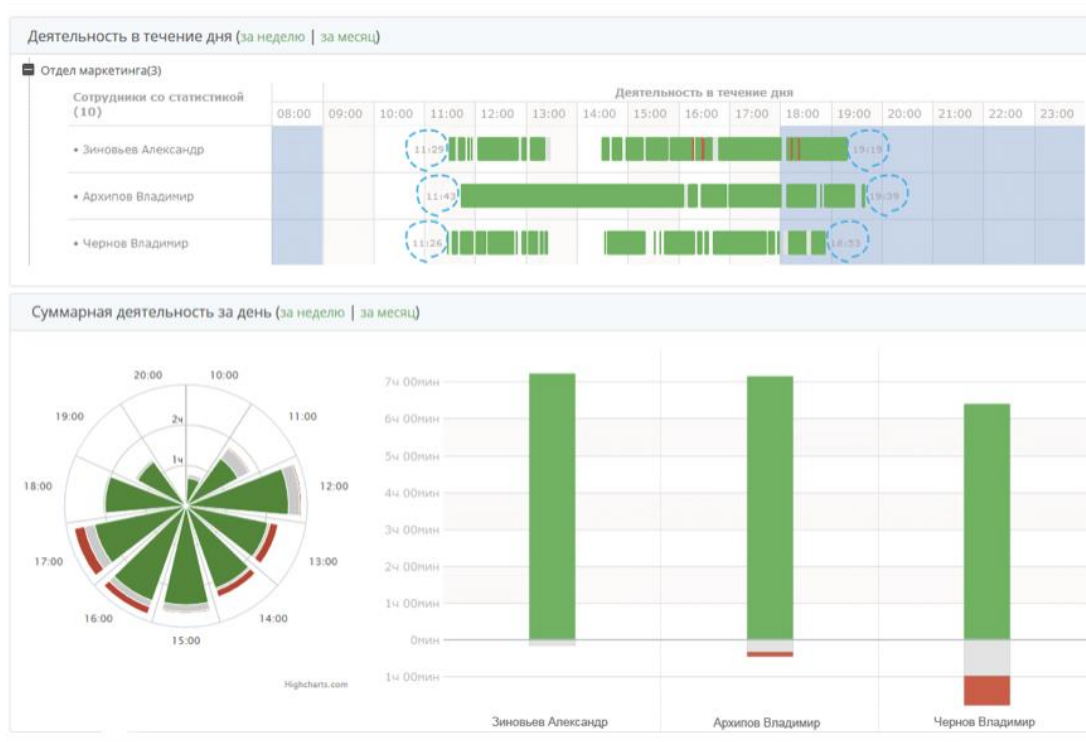


Рисунок 1 – Изображение экрана сервиса Yaware

- Мониторинг программ и сайтов, анализ продуктивности – какими программами пользуются сотрудники, также есть категории программ и сайтов по продуктивности, с возможностью их изменений [4] (рисунок 2).

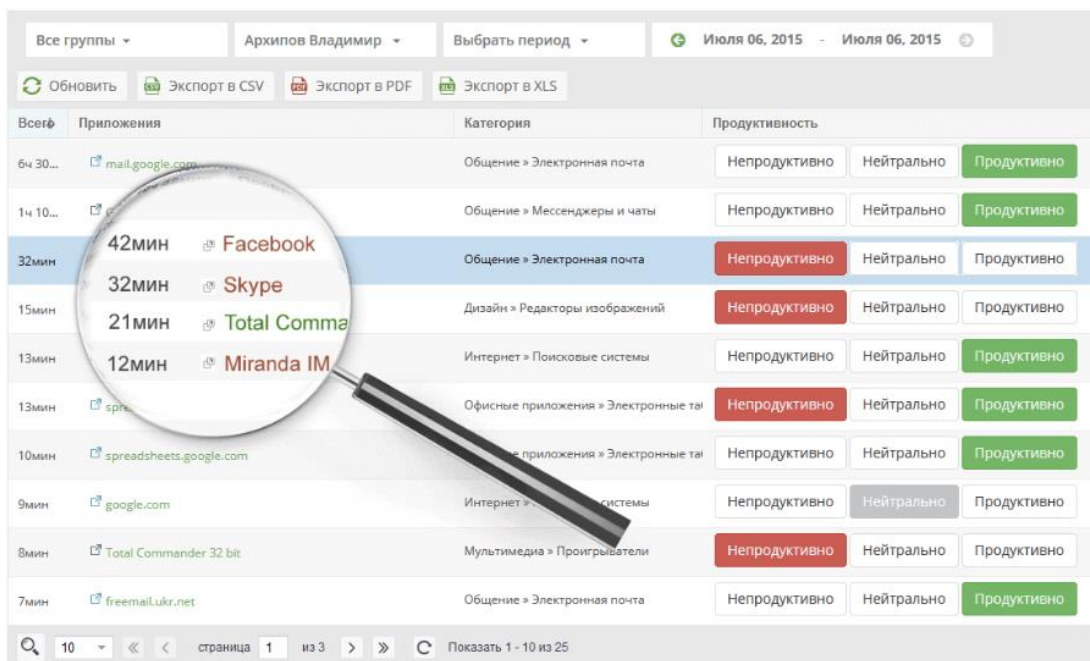


Рисунок 2 – Изображение экрана сервиса Yaware

- Учет рабочего времени не за компьютером – сколько времени сотрудник отсутствовал на рабочем месте, чем он конкретно был занят, когда не работал за компьютером (комментарии с причиной) [4] (рисунок 3).

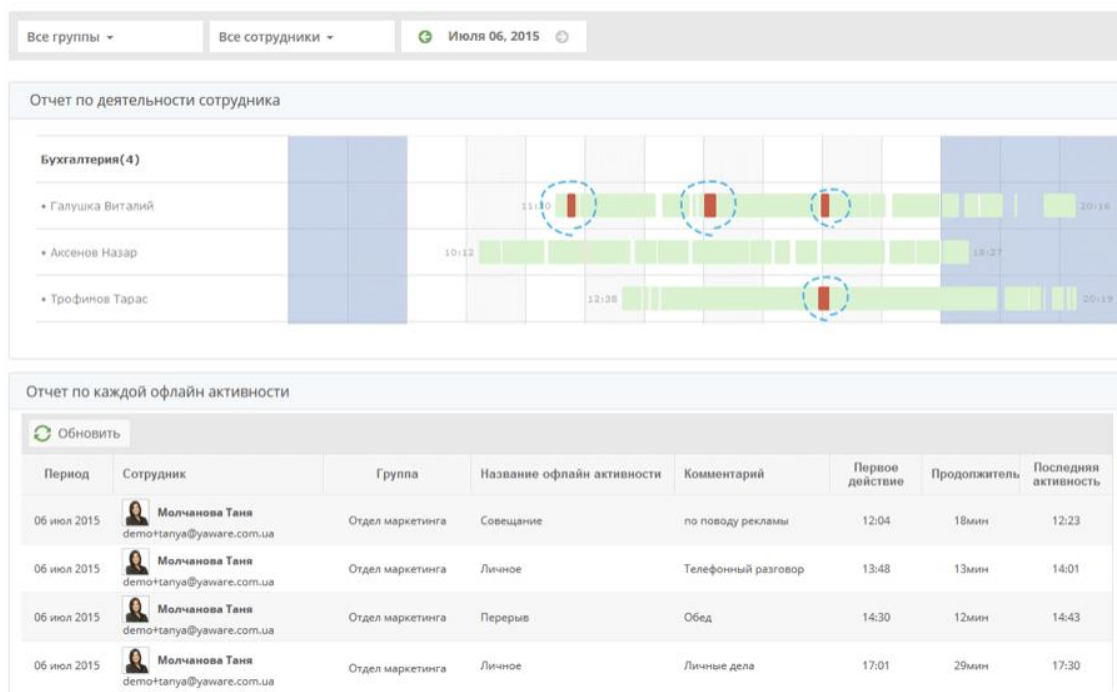


Рисунок 3 – Изображение экрана сервиса Yaware

- Скриншоты и снимки веб-камеры[4] (рисунок 4).

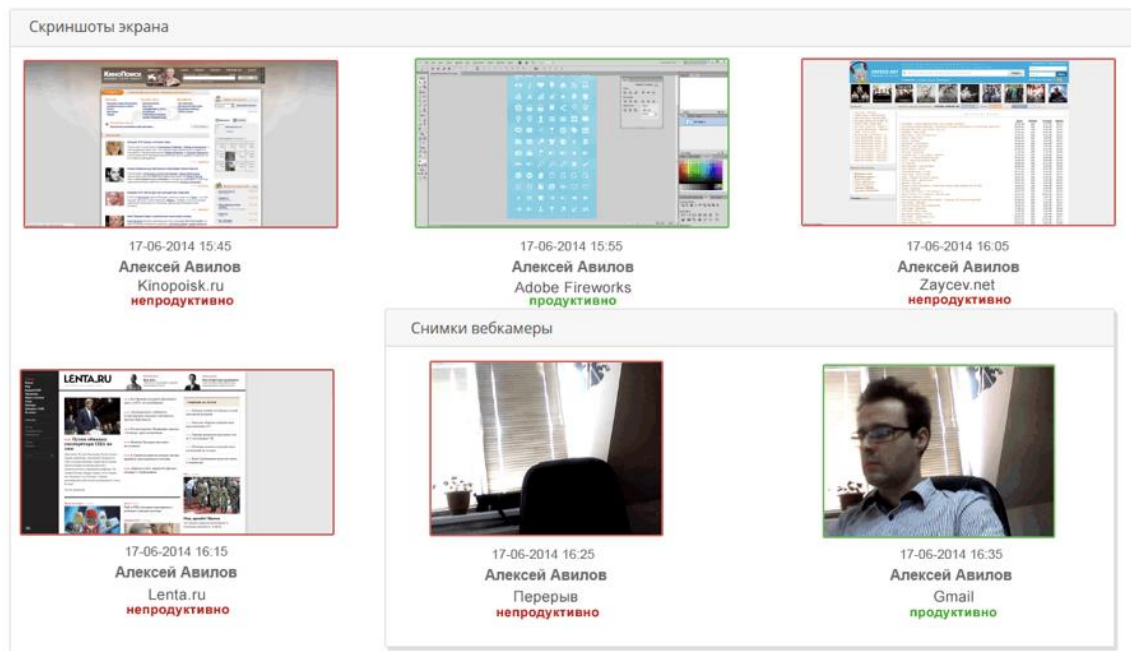


Рисунок 4 – Изображение экрана сервиса Yaware

- Контроль нарушений – различные отчеты, включающие, например, отчет по распорядку дня [4] (рисунок 5).

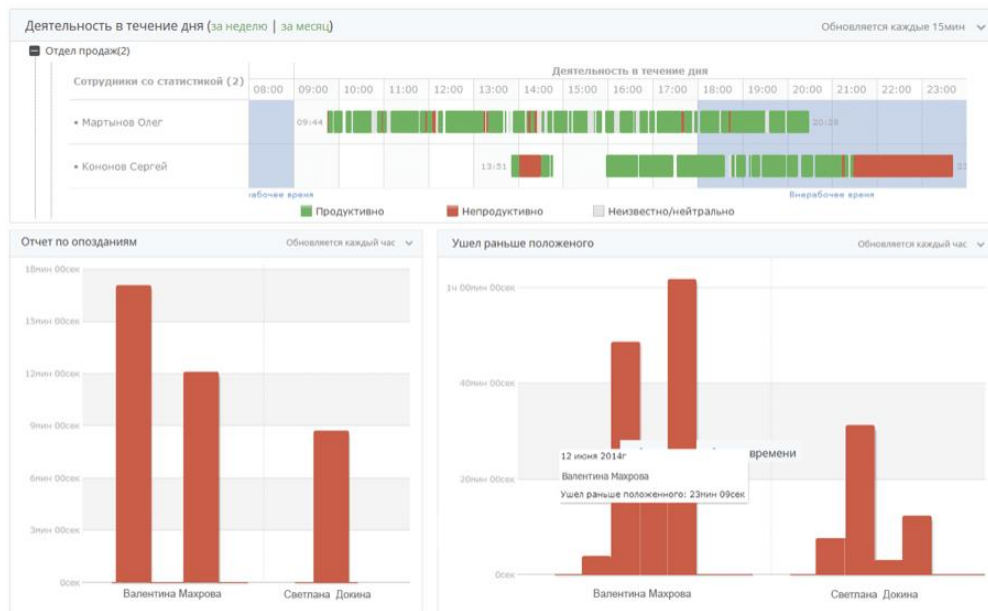


Рисунок 5 – Изображение экрана сервиса Yaware

- Оперативное уведомление – анализ статистики сотрудников с помощью мобильного приложения на базе iOS и Android [4] (рисунок 6).

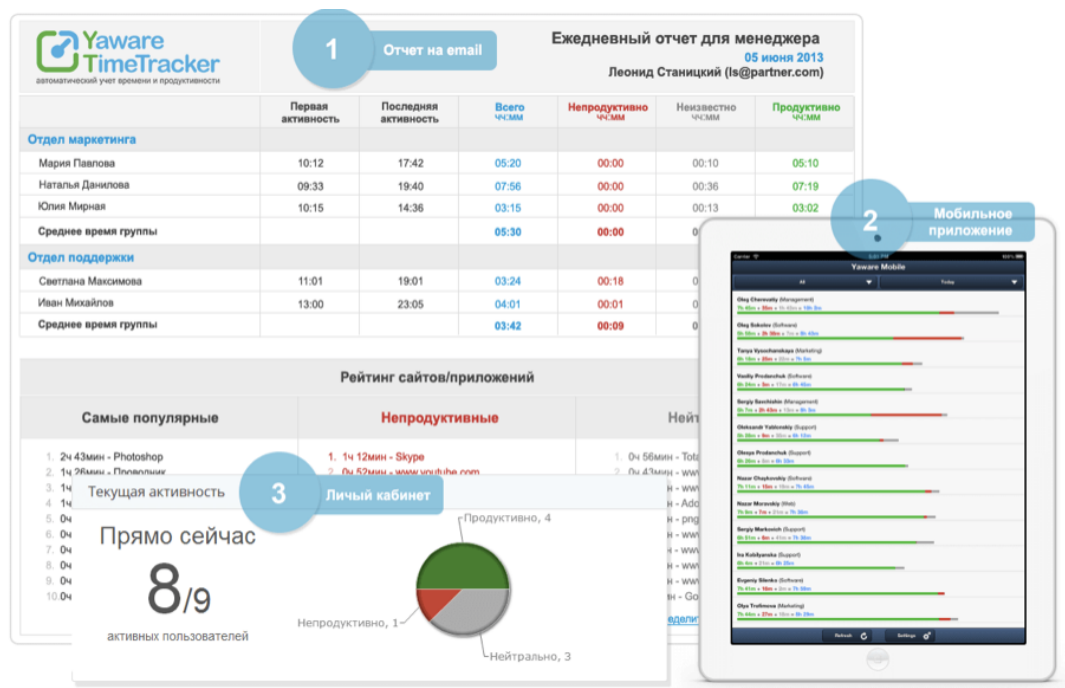


Рисунок 6 – Изображение экрана сервиса Yaware

- Группировка сотрудников по отделам – группировка в группы исходя из отделов, в которых они работают и какими задачами они занимаются [4] (рисунок 7).

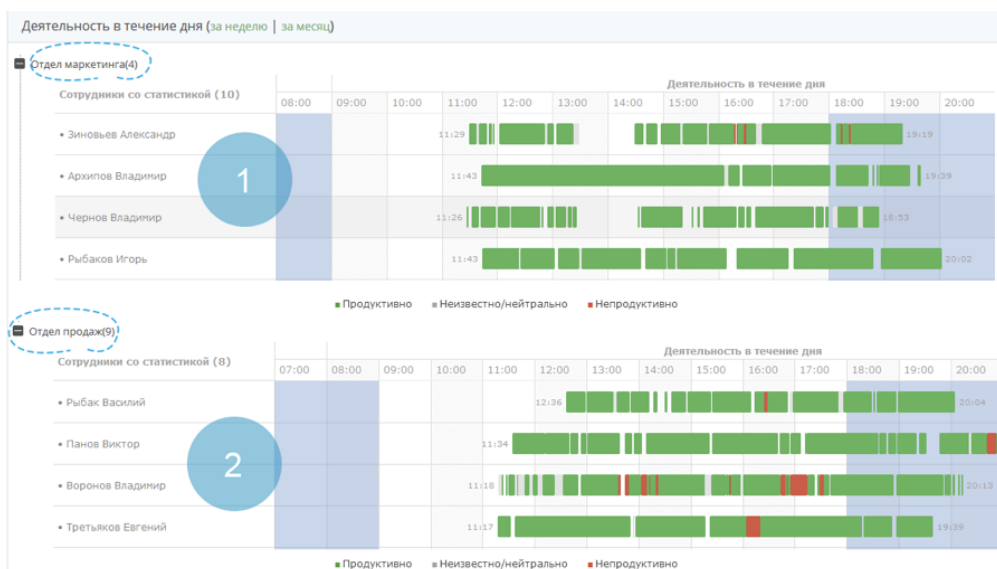


Рисунок 7 – Изображение экрана сервиса Yaware

- Индивидуальные настройки сотрудника и целого отдела – возможность задавать индивидуальный график, и статистику для каждого сотрудника [4] (рисунок 8).

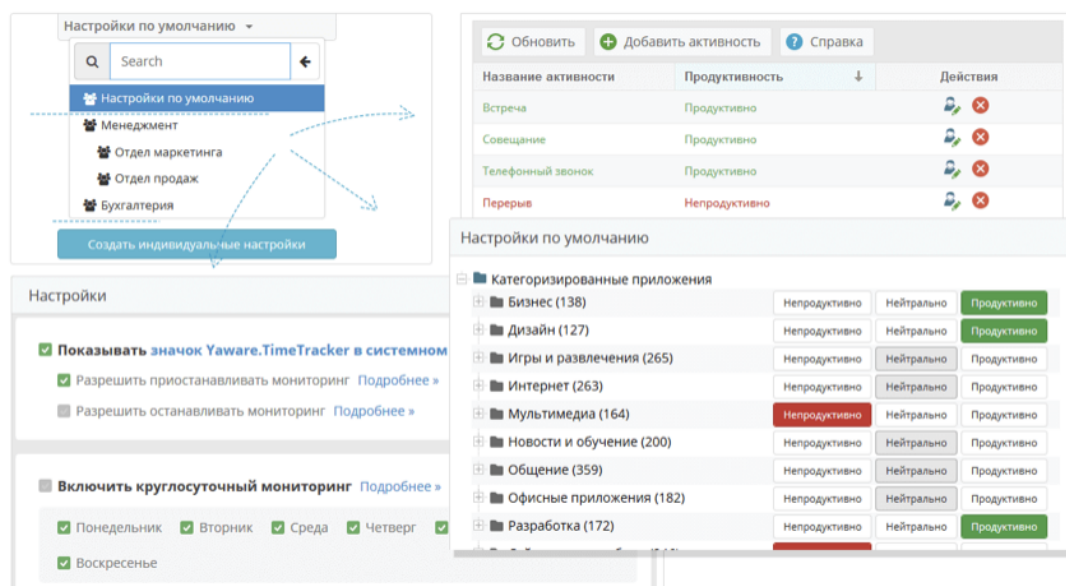


Рисунок 8 – Изображение экрана сервиса Yaware

1.2. Time Doctor

Time Doctor – это учёт рабочего времени, снятие скриншотов экрана, контроль сотрудников по использованию сайтов и приложений, тайм трекер, интеграции [3].

Возможности:

- Отслеживание времени – учёт рабочего времени, которое проработал каждый сотрудник, предоставление отчётов [3] (рисунок 9).



Рисунок 9 – Изображение экрана сервиса Time Doctor

- Контроль использования интернета – уведомления, при чрезмерном количестве времени потраченного на нерабочие ресурсы [3] (рисунок 10).

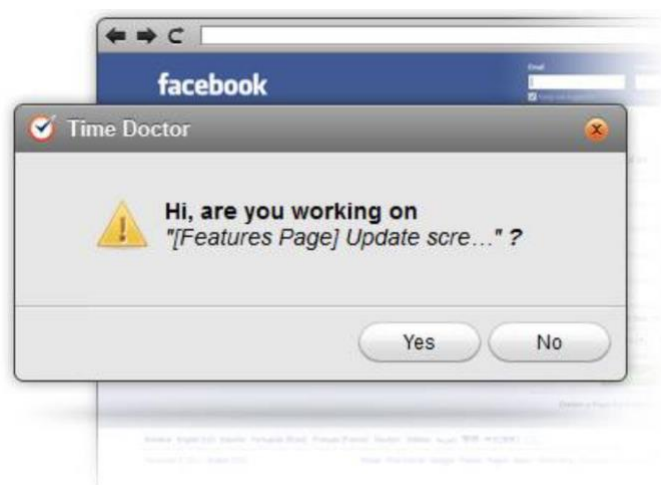


Рисунок 10 – Изображение экрана сервиса Time Doctor

- Мониторинг сайтов и приложений – отчёта по использованию сайтов и приложений, которыми пользуется сотрудник [3] (рисунок 11).



Рисунок 11 – Изображение экрана сервиса Time Doctor

- Интеграции – возможность интеграции с наиболее популярными сервисами управления компаниями [3] (рисунок 12).

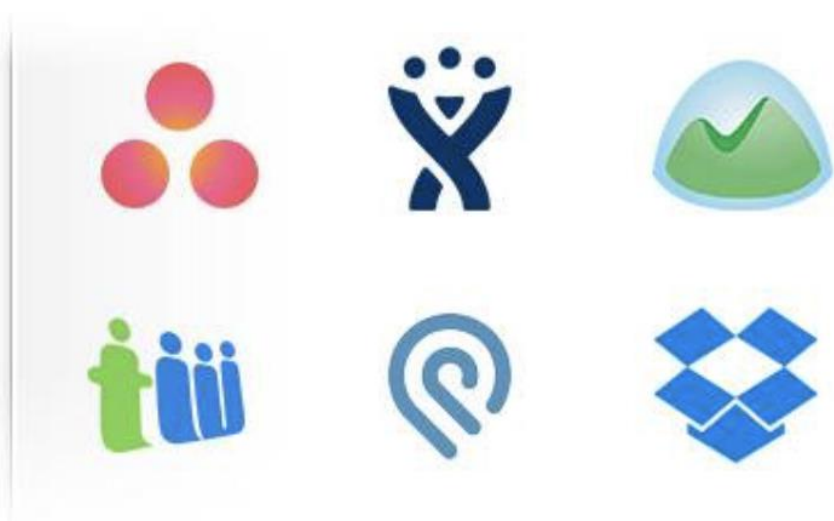


Рисунок 12 – Изображение с логотипами сервисом, с которыми возможна интеграция сервиса Time Doctor

- Скриншоты экрана и движения мыши (опционально) – каждый скриншот экрана предоставляет уровень активности мыши и клавиатуры на данном промежутке времени [3] (рисунок 13).

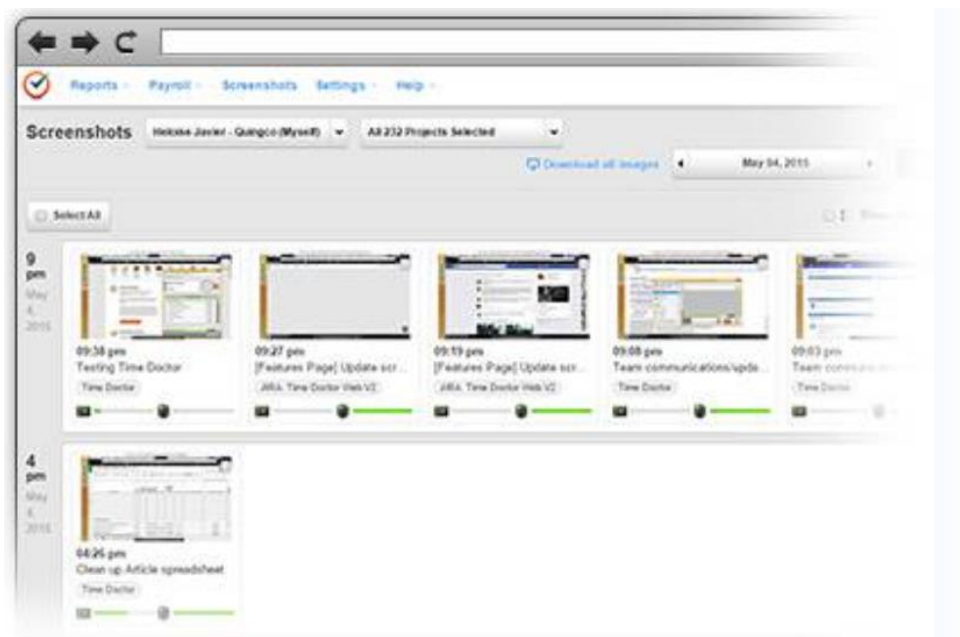


Рисунок 13 – Изображение экрана сервиса Time Doctor

- Вход для клиентов – возможность клиентов просматривать отчёты по задачам, над которыми работают сотрудники [3].
- Платёжные ведомости – настройка ведомостей по выплатам сотрудникам (доступна автоматизация) [3].
- Конфиденциальность и безопасность – вся информация шифруется при отправке на сервера Time Doctor [3].
- Простая установка и поддержка высокого уровня – доступно на MAC, Windows, Linux [3].
- Мобильное приложение – возможность использования сервиса не только на компьютере, но и на мобильном устройстве [3] (рисунок 14).



Рисунок 14 – Изображение экрана мобильного приложения сервиса Time Doctor

1.3. Dreel

Dreel — приложение, которое отслеживает и анализирует рабочее время сотрудников, позволяет оценивать их продуктивность и оптимизировать работу компании. Вся собранная статистика отображается в личном кабинете руководителя в наглядном виде [2].

Возможности:

- Учёт рабочего времени – скриншоты экрана, контроль нажатия клавиш, кликов мышью и запущенных программ происходит каждую минуту [2] (рисунок 15).

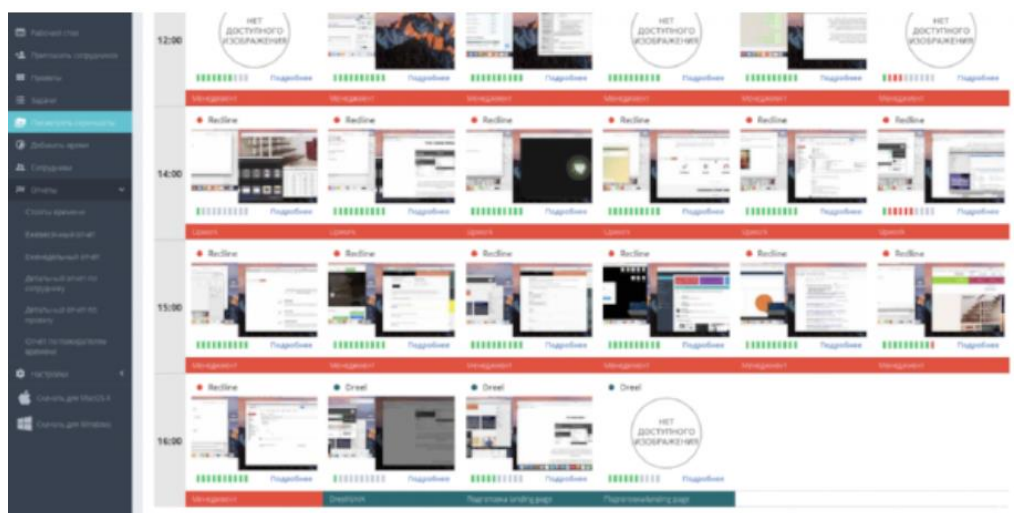


Рисунок 15 – Изображение экрана сервиса Dree!

- Учёт рабочего времени, проведенного сотрудником не за компьютером – комментарии с причиной и пояснением отсутствия на рабочем месте [2].
- Отчеты - сводные отчеты по компании, детальные отчеты по каждому сотруднику, отдельный отчет по пожирателям времени [2] (рисунок 16).

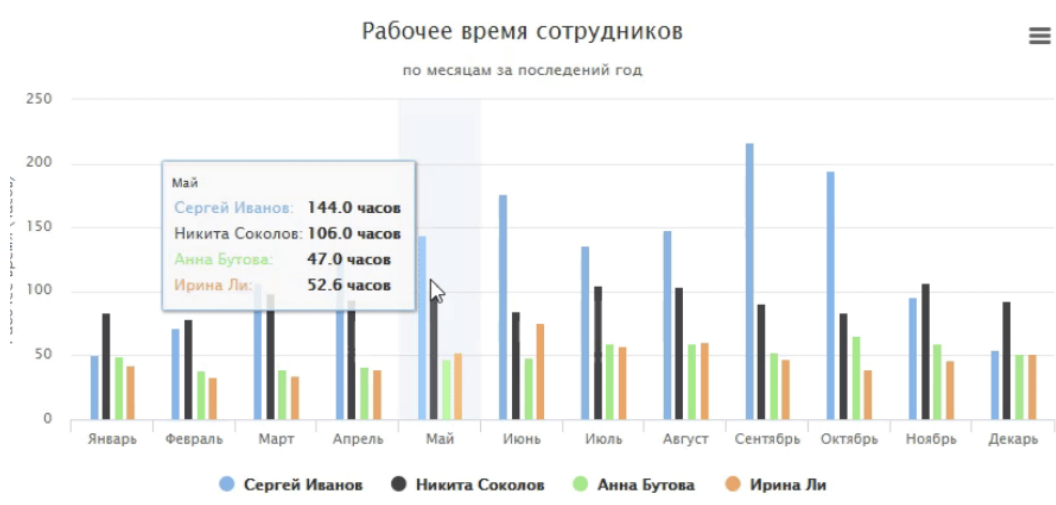


Рисунок 16 – Изображение экрана сервиса Dree!

- Пожиратели времени – фиксация времени, которое сотрудник тратит на социальные сети, youtube-каналы или на подобные сайты, которые отвлекают от работы [2] (рисунок 17).

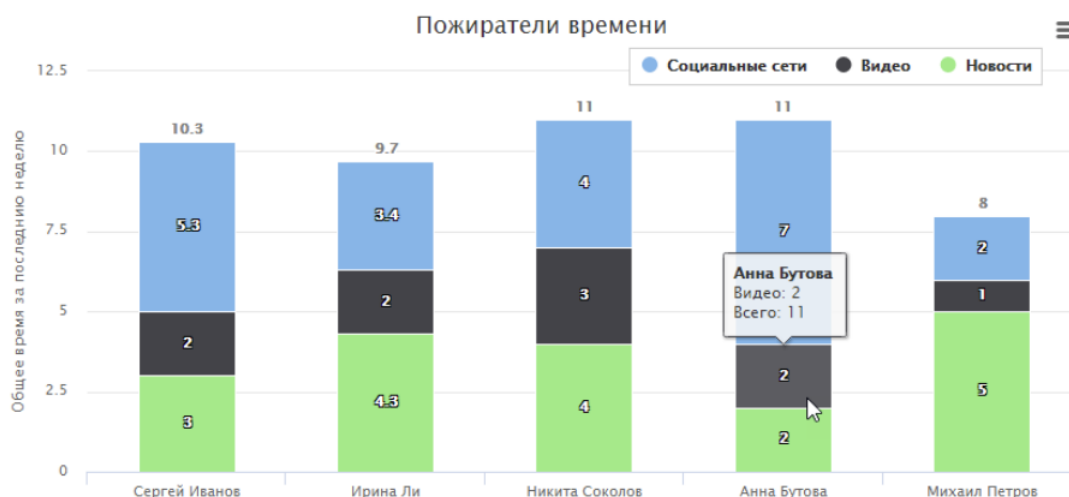


Рисунок 17 – Изображение экрана сервиса Dreel

- Интеграция – возможность интеграции со сторонними приложениями [2] (рисунок 18).



Рисунок 18 – Изображение с логотипами сторонних сервисов, с которыми возможна интеграция сервиса Dreel

- Конфиденциальность и безопасность – вся информация шифруется при отправке на сервера Dreel [2].
- Простая установка и поддержка высокого уровня – доступно на MacOS, Windows, Linux (Ubuntu, Mint), оперативная техническая поддержка [2].
- Автоотключение и выключение приложения – приложение самостоятельно включается и выключается (доступна настройка автоотключения) [2].

1.4. Сравнение существующих решений

После рассмотрения наиболее двух известных конкурентов сервиса Dreel, можно составить таблицу сравнений, представленную в виде сравнения наличия наиболее важных

функций, представленных сервисов. В таблице 1 можно увидеть, что сервисы конкуренты и Dreel имеют почти одинаковые возможности, но у всех есть какие-то свои положительные и отрицательные черты.

Таблица 1 – Сравнение существующих решений и сервиса Dreel

Возможность	Yaware TimeTracker	Time Doctor	Dreel
Учёт рабочего времени и его анализ с отчётами	+	+	+
Мониторинг программ и сайтов	+	+	+
Скриншоты экрана	+	+	+
Учёт времени работы не за компьютером	+	+	+
Отчёты по рабочему времени сотрудников	+	+	+
Автоматический запуск и остановка таймера	+	-	+
Шифрование данных перед отправкой на сервер	-	+	+
Мобильное приложение	+	+	-

Одной из проблем Dreel, по которым он уступает своим конкурентам является отсутствие мобильного приложения. Действительно, данным сервисом пользуются десятки компаний, но у него до сих пор нет мобильного приложения для оперативного мониторинга рабочего времени сотрудников. В ходе телефонных интервью в рамках данной работы было выяснено, что 70% пользователей выразили желание пользоваться мобильным приложением для просмотра отчётов и учета рабочего времени вне офиса.

Таким образом, разработка мобильного приложения для сервиса Dreel является актуальной задачей.

2. Требования к системе

2.1. Нефункциональные требования

Нефункциональные требования:

- Кроссплатформенное мобильное приложение, поддерживающее такие платформы как: iOS и Android
- Ionic Framework – фреймворк, который лучше всего подходит для решения данной задачи

2.2. Функциональные требования

В ходе исследований об актуальности разрабатываемого мобильного приложения в рамках данной выпускной квалификационной работы, также был произведен опрос, на основе которого был составлен список функционала, требуемого для пользователей.

Перечень функционала, выявленный в ходе исследований:

- Подтвердить назначенное время сотрудника
- Посмотреть список сотрудников
- Поменять ставку сотрудника
- Посмотреть список проектов
- Поменять ставку сотрудника в проекте
- Посмотреть отчёты
- Запустить таймер
- Поделиться ссылкой регистрации

«Поменять ставку сотрудника» и «Поменять ставку сотрудника в проекте» - разные желаемые возможности разрабатываемого приложения. Сотрудник имеет стандартную ставку, но в некоторых проектах его ставка может повышаться или уменьшаться, в зависимости от значимости проекта, его сроков, стоимости.

«Запустить таймер» - данная желаемая функция разрабатываемого мобильного приложения дает возможность менеджеру на встречах включать таймер, для мониторинга своей же работы. Сколько времени он тратит на деловые встречи с заказчиками или планерки с сотрудниками.

2.3. Диаграмма вариантов использования

На основе функциональных требований была составлена диаграмма вариантов использования (рисунок 19).

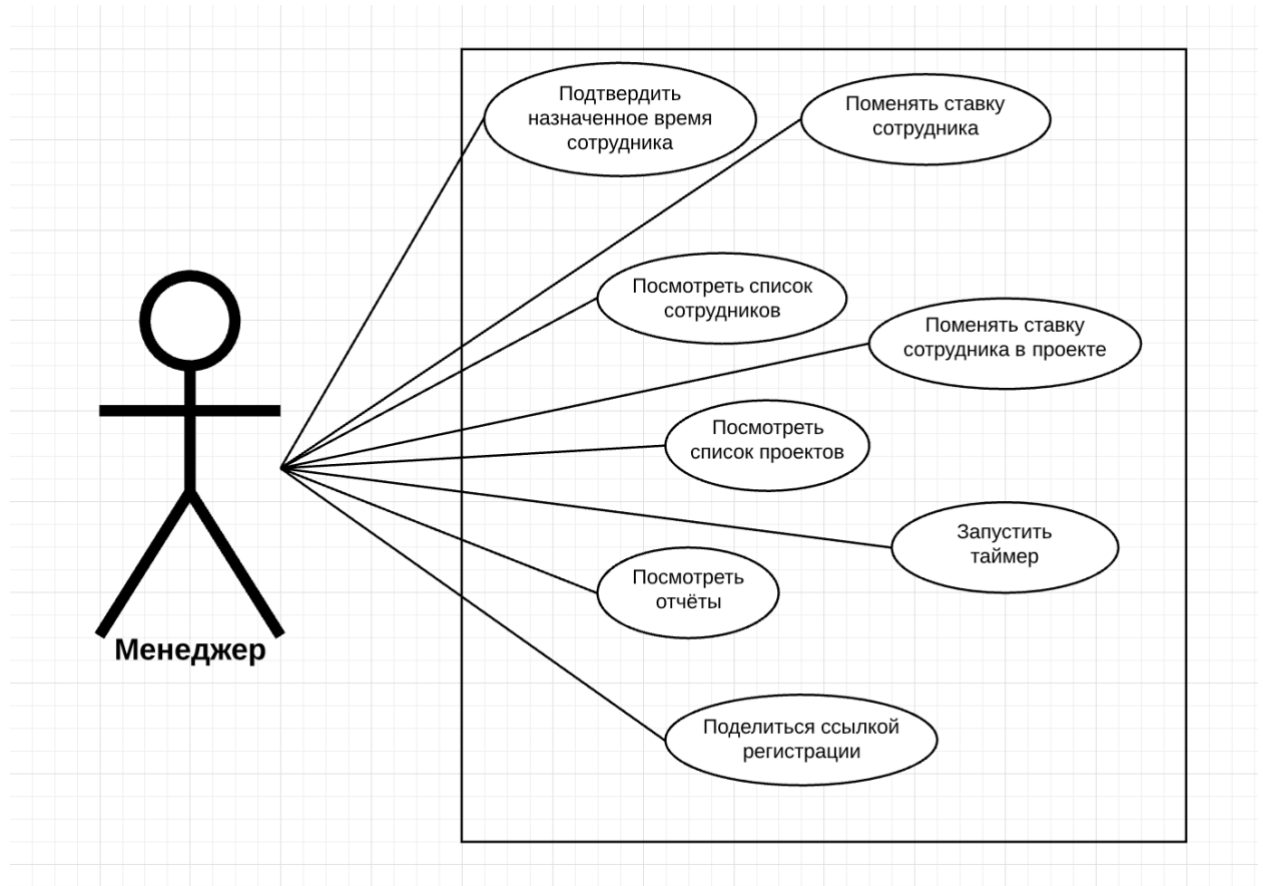


Рисунок 19 – Диаграмма вариантов использования

2.4. Сценарии вариантов использования

Сценарий варианта использования «Подтвердить назначенное время сотрудника»:

1. Менеджер выбирает в основном нижнем меню приложения страницу с подтверждением времени
2. Система открывает экран со списком заявок сотрудников на подтверждение времени
3. Менеджер читает заявку, выбирает подтвердить / не подтвердить время, нажимает на соответствующую кнопку
4. Система отправляет POST запрос на сервер с выбранными параметрами

Сценарий варианта использования «Посмотреть список сотрудников»:

1. Менеджер выбирает в основном нижнем меню приложения страницу с дополнительным меню, состоящим из нескольких пунктов
2. Система открывает экран со списком дополнительного меню
3. Менеджер нажимает на кнопку «список сотрудников»
4. Система открывает экран список всех сотрудников

Сценарий варианта использования «Поменять ставку сотрудника»:

1. Менеджер выбирает в основном нижнем меню приложения страницу с дополнительным меню, состоящим из нескольких пунктов
2. Система открывает экран со списком дополнительного меню
3. Менеджер нажимает на кнопку «список сотрудников»
4. Система открывает экран со списком всех сотрудников
5. Менеджер выбирает нужного сотрудника и нажимает на него
6. Система открывает экран с информацией о соответствующем сотруднике
7. Менеджер вводит в нужное поле ставку, которую он хочет поставить сотруднику и нажимает на кнопку «изменить ставку»
8. Система отправляет POST запрос с нужными данными и изменяет ставку выбранному менеджером сотруднику

Сценарий варианта использования «Посмотреть список проектов»:

1. Менеджер выбирает в основном нижнем меню приложения страницу с проектами
2. Система открывает экран со списком проектов

Сценарий варианта использования «Поменять ставку сотрудника в проект»:

1. Менеджер выбирает в основном нижнем меню приложения страницу с проектами
2. Система открывает экран со списком проектов
3. Менеджер выбирает нужный проект и нажимает на него
4. Система открывает экран со списком сотрудников в соответствующем проекте
5. Менеджер в списке находит нужного сотрудника, вводит в поле ставку, которую требуется поставить в данном проекте для него, нажимает на кнопку «изменить ставку»
6. Система отправляет POST запрос с параметрами на сервер, изменяет ставку соответствующему сотруднику в проекте

Сценарий варианта использования «Посмотреть отчёты»:

1. Менеджер выбирает в основном нижнем меню приложения страницу с дополнительным меню, состоящим из нескольких пунктов
2. Система открывает экран со списком дополнительного меню
3. Менеджер нажимает на кнопку «посмотреть отчёты»
4. Система открывает экран со списком отчётов
5. Менеджер выбирает нужным отчёт
6. Система открывает экран с соответствующим отчётом
7. Менеджер выбирает нужные параметры, нажимает на кнопку «загрузить отчёт»
8. Система отправляет POST запрос с параметрами на сервер, получает данные для отчёта, выводит отчёт с заданными данными

Сценарий варианта использования «Поделиться ссылкой регистрации»:

1. Менеджер выбирает в основном нижнем меню приложения страницу с дополнительным меню, состоящим из нескольких пунктов
2. Система открывает экран со списком дополнительного меню
3. Менеджер нажимает на кнопку «Пригласить пользователя»
4. Система открывает экран с ссылкой на регистрацию
5. Менеджер нажимает на кнопку «копировать ссылку»
6. Система копирует ссылку в буфер устройства

Сценарий варианта использования «Включить таймер»:

1. Менеджер выбирает в основном нижнем меню приложения страницу с таймером
2. Система открывает экран с таймером
3. Менеджер нажимает кнопку «start»
4. Система отправляет POST запрос на сервер, включает таймер времени
5. Менеджер нажимает на кнопку «stop»
6. Система отправляет POST запрос на сервер, отключает таймер времени

3. Используемые технологии

3.1. Ionic Framework

Одним из нефункциональных требований для разработки кроссплатформенного приложения был выбран Ionic. Действительно, проанализировав функционал, который должен быть в разрабатываемом мобильном приложении в рамках данной выпускной квалификационной работы и изучив данный фреймворк, можно уверенно сказать, что для данной задачи - это будет наилучшим решением.

Ionic Framework — это фреймворк для создания гибридных мобильных приложений. В его состав входит набор JavaScript и CSS компонент, созданных на основе Angular 2, SASS и Apache Cordova [6]. История этого SDK начинается в 2013 году, когда компания Drifty Co. решила создать собственную инфраструктуру для написания гибридных приложений, которая будет ориентирована на производительность и будет построена с использованием современных веб-стандартов. Релиз Ionic 1 – состоялся в мае 2015 года. На сегодняшний день уже существует Ionic 3.

По своей сути, Ionic Framework — это дополнение над очень популярным фреймворком Apache Cordova, но со своим мощным CLI (Command Line Interface) и объемной документацией. Следуя принципам Apache Cordova, приложения на Ionic Framework — это гибридные HTML приложения. Такие приложения на телефоне выполняются в специальной оболочке (UIWebView для iOS и WebView для Android), которая позволяет показывать HTML и выполнять JavaScript. Соответственно при работе в приложении пользователь работает как бы в веб-браузере (рисунок 20).

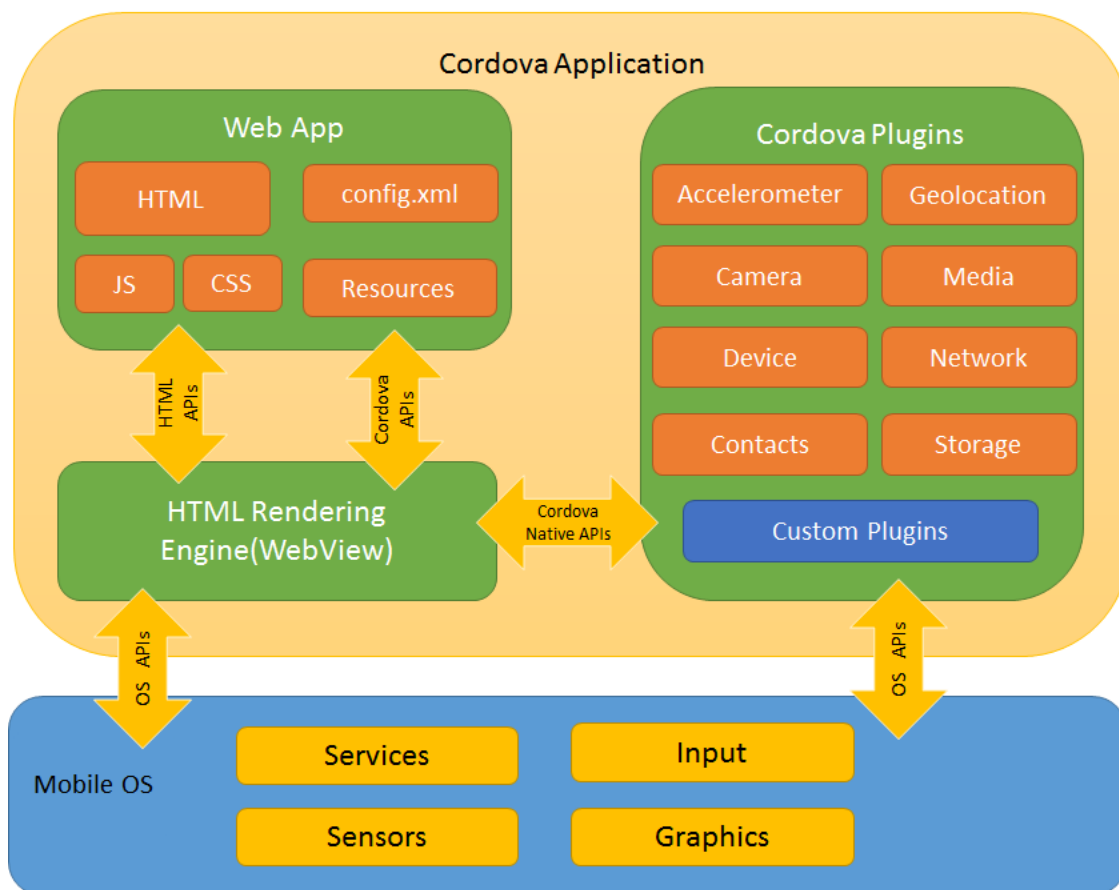


Рисунок 20 – Архитектура Cordova Application

Apache Cordova является своеобразной прослойкой между пользовательским интерфейсом и ресурсами устройства. Какие-либо виджеты пользовательского интерфейса или MV* (Model-View) фреймворков не входят в его состав. Если необходимо использовать UI-виджеты и/или MV* фреймворк, то нужно выбрать и включить их в приложение самостоятельно, как ресурсы третьей стороны. Ionic 3 — это один из фреймворков, который предоставляет UI виджеты. MVC модель для него предоставляет Angular 4.

Нативные функции телефона (например, камера, хранилище ключей, GPS координаты) недоступны из веб-браузера. Поэтому для работы с ними используются плагины Apache Cordova. Кроме официальных плагинов есть ряд сторонних открытых плагинов.

Преимущества [6]:

- Бесплатный и в открытом доступе
- Кроссплатформенность
- Использует более 120 функций устройств, есть возможность использовать плагины других известных фреймворков (Cordova) и расширения TypeScript
- Очень подробная документация с демонстрационными примерами
- Облачный сервис для сборки приложения и его демонстрации

- Быстрая установка и развертывание стартового проекта без особых усилий, с помощью командной строки

Недостатки:

- Не подходит для приложений, в которых графика играет ключевую роль
- Анимация, прокрутка и сетевые операции по качеству уступают нативным приложениям

3.2. WebStorm

WebStorm — интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA [7].

Главное достоинство WebStorm — это удобный и умный редактор для JavaScript, HTML и CSS, который также поддерживает TypeScript, CoffeeScript, Dart, Less, Sass и Stylus и фреймворки, например, Angular, React и Vue.js (рисунок 21).

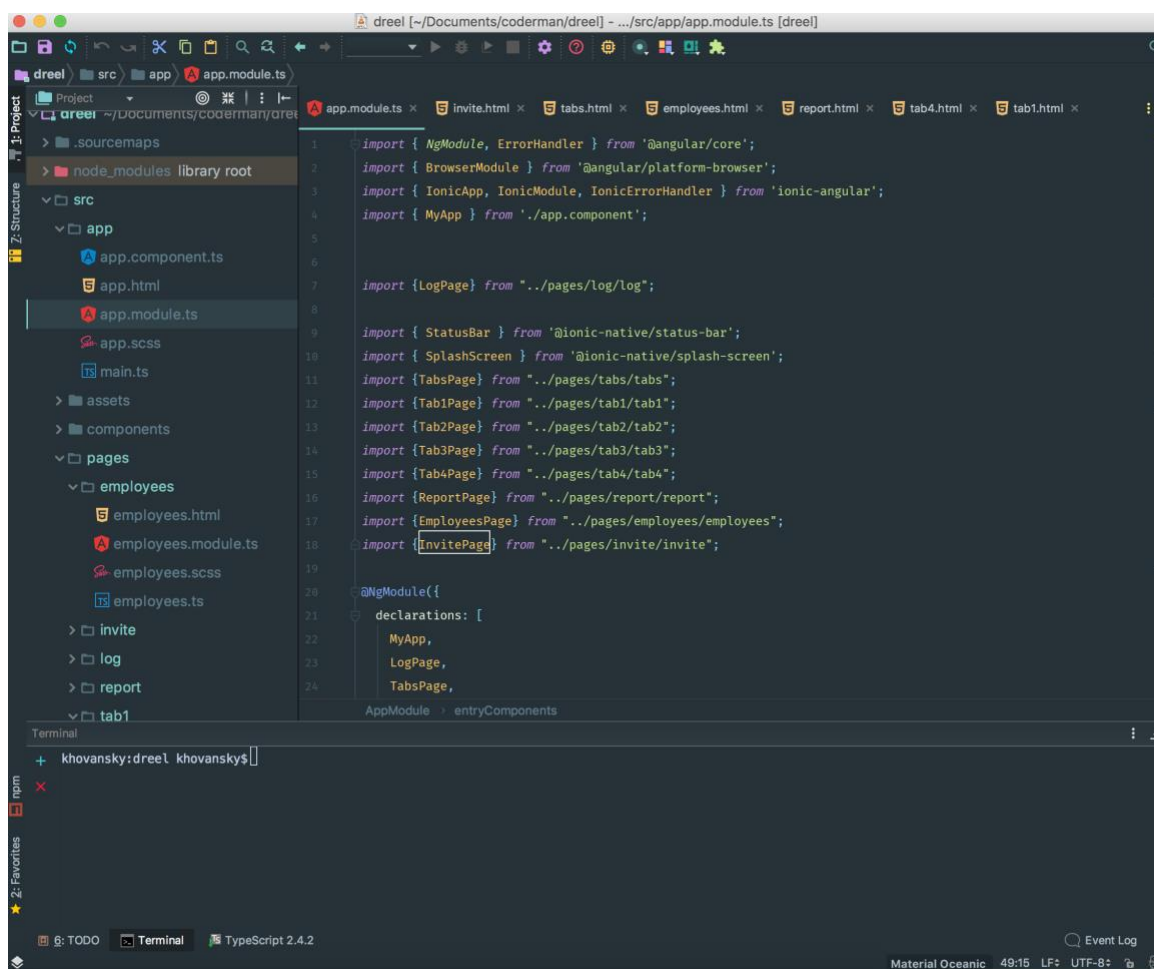


Рисунок 21 – Изображение экрана компьютера с IDE WebStorm

WebStorm, как и другие IDE на платформе IntelliJ IDEA, делает разработку проще и удобней. WebStorm обеспечивая подсветку и автодополнение кода, проверяет его на ошибки, помогает быстро навигироваться по проекту и безопасно вносить изменения с помощью рефакторингов. В WebStorm есть инструменты для отладки кода и интеграция с системами управления версиями [7].

WebStorm понимает структуру проекта и код, обнаруживает возможные проблемы еще до того, как проект откроется в браузере, и предлагает их решение.

Встроенные в IDE инструменты для тестирования помогают в разработке и делают ее удобней и продуктивней.

Ключевые возможности:

- отладка client-side-, Node.js- и React Native-приложений прямо в WebStorm
- помощь при работе с приложениями на Angular, React, Vue.js, Electron и Meteor
- интеграция с системами управления версиями Git, GitHub, Subversion, Perforce и Mercurial
- интеграция с системами отслеживания ошибок (ESLint, JSHint, JSLint, TSLint и Stylelint)

Live Edit позволяет мгновенно видеть результаты ваших изменений в CSS, HTML и JavaScript файлах проекта в браузере без перезагрузки страницы. Live Edit доступен в режиме отладки JavaScript и работает в браузере Google Chrome с предустановленным плагином JetBrains IDE Support [7].

WebStorm поддерживает современные языки, компилируемые в JavaScript, такие как TypeScript, CoffeeScript и новую версию JavaScript – ECMAScript 2017, предлагая автодополнение кода, проверку ошибок, рефакторинг и отладку [7].

WebStorm позволяет эффективно разрабатывать приложения на Node.js и поддерживает полноценную отладку Node.js приложений. Новое приложение можно создать, используя шаблон Node.js Express, а необходимые модули легко установить через npm, не используя командную строку [7].

Кроме того, JetBrains предоставляют бесплатную годовую лицензию для студентов на свои продукты при отправке подтверждающего документа.

Таким образом, данная IDE была выбрана для разработки мобильного приложения.

3.3. Chart.js

Chart.js является JavaScript-библиотекой, с помощью которой можно создавать на странице различные типы диаграмм, при этом используются возможности HTML5-элемента [5].

Библиотека Chart.js не имеет каких-либо зависимостей; размер ее так мал, что составляет примерно 11Kb в минифицированном, конкатенированном и архивированном виде [5]. Однако, есть возможность еще уменьшить размер библиотеки путем включения в нее только тех модулей, которые необходимы в конкретном случае; то есть, если нет необходимости в поддержке всех шести типов диаграмм, включенных по умолчанию в ядро Chart.js [5].

К примеру, если на конкретной странице нужно создать только диаграмму линейного типа (bar chart), то можно подключить лишь ядро библиотеки Chart.js и модуль, с помощью которого создаются диаграммы подобного типа; таким образом, будет уменьшен общий размер библиотеки Chart.js и увеличена скорость загрузки в целом.

Другой замечательной особенностью библиотеки Chart.js является тот факт, что создаваемые ею диаграммы адаптивные; то есть, они могут изменять свой размер, при изменении размеров окна браузера таким образом, чтобы эффективно занимать все доступное для этого пространство страницы.

И наконец, в отличие от многих других библиотек подобного рода, которые можно найти на просторах Интернет, Chart.js имеет подробную и хорошо изложенную документацию, с помощью которой можно легко создавать как простые, так и сложные образцы диаграмм.

На рисунке 22 показан примерный результат того, какую круговую диаграмму можно получить с помощью библиотеки Chart.js:

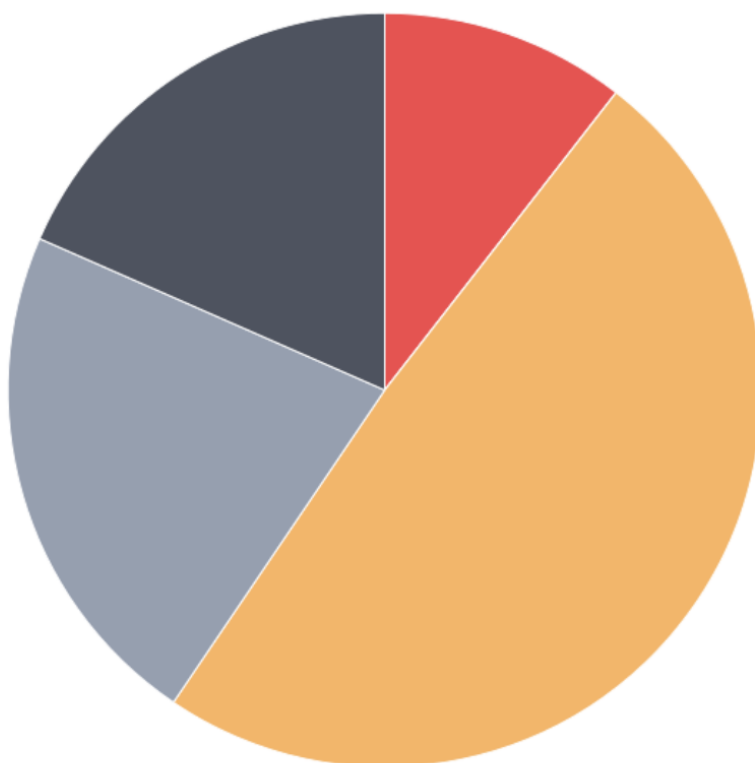


Рисунок 22 – Изображение с примером круговой диаграммы построенной при помощи библиотеки Chart.js

На рисунке 23 показан примерный результат того, какую круговую диаграмму можно получить с помощью библиотеки Chart.js:

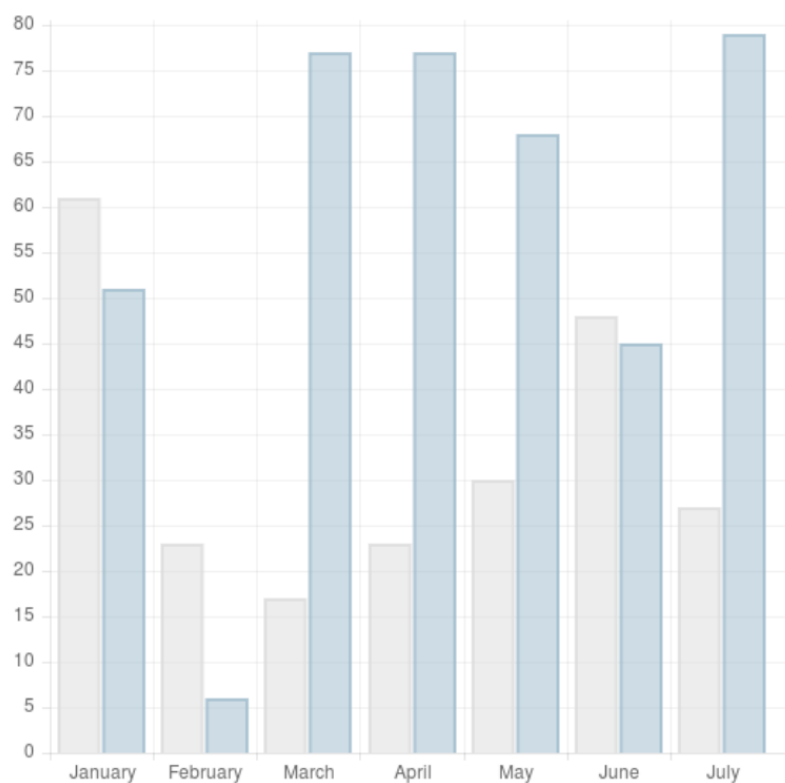


Рисунок 23 – Изображение с примером столбцовой диаграммы построенной при помощи библиотеки Chart.js

Таким образом, Chart.js является JavaScript-библиотекой для быстрого и удобного создания красивых диаграмм, которая отлично подходит для решения задач, поставленных в данной выпускной квалификационной работе. Также, использование библиотеки не представляет каких-либо трудностей.

4. Разработка приложения

4.1. Прототип интерфейса

Одной из первых задач, поставленных при разработке мобильного приложения стала разработка прототипа интерфейса удовлетворяющим функциональным требованиям. Ведь при создании прототипа интерфейса надо задуматься о многих моментах, например, чтобы потом можно было без каких-либо проблем его реализовать.

Для прототипа был выбран Marvelapp. Marvelapp - бесплатный веб-сервис для связки макетов прототипа интерфейса приложений с имитацией на различных устройствах, представляющий возможность в онлайн режиме делиться своим прототипом с другими людьми.

Макеты интерфейса разрабатываемого мобильного приложения нарисованы в векторном редакторе CoralDraw.

В итоге, изначально был сделан начальный вариант прототипа интерфейса разрабатываемого мобильного приложения. После чего, для дальнейших исправлений он был отправлен на тестирование потенциальным пользователям. В ходе исследований было выяснено, что нравится и не нравится. Второй вариант прототипа интерфейса разрабатываемого мобильного приложения был также представлен для тестирования потенциальным пользователям, после чего и был сделан финальный прототип интерфейса разрабатываемого мобильного приложения (рисунок 24).



Рисунок 24 – Изображения экранов прототипа интерфейса разрабатываемого мобильного приложения

4.1. Архитектура системы

На рисунке 25 представлена диаграмма развертывания, представляющая архитектуру системы, состоящую из 3 частей:

- Узел Computer - это компьютеры сотрудников, с которых посылаются данные об их работе на узел Server.
- Узел Server – место где хранятся данные о работе сотрудников, а также скриншоты. В данной дипломной работе узел Server не был рассмотрен, потому что он был реализован другим разработчиком.
- Узел Mobile phone – мобильные устройства, на которых установлено разрабатываемое мобильное приложение. Взаимодействие узла Mobile phone и узла Server происходит по протоколу HTTP. Приложение отправляет post-запросы. Обмен информацией между узлами Server и Mobile phone производится в текстовом формате JSON, а для изображений используется формат JPEG.

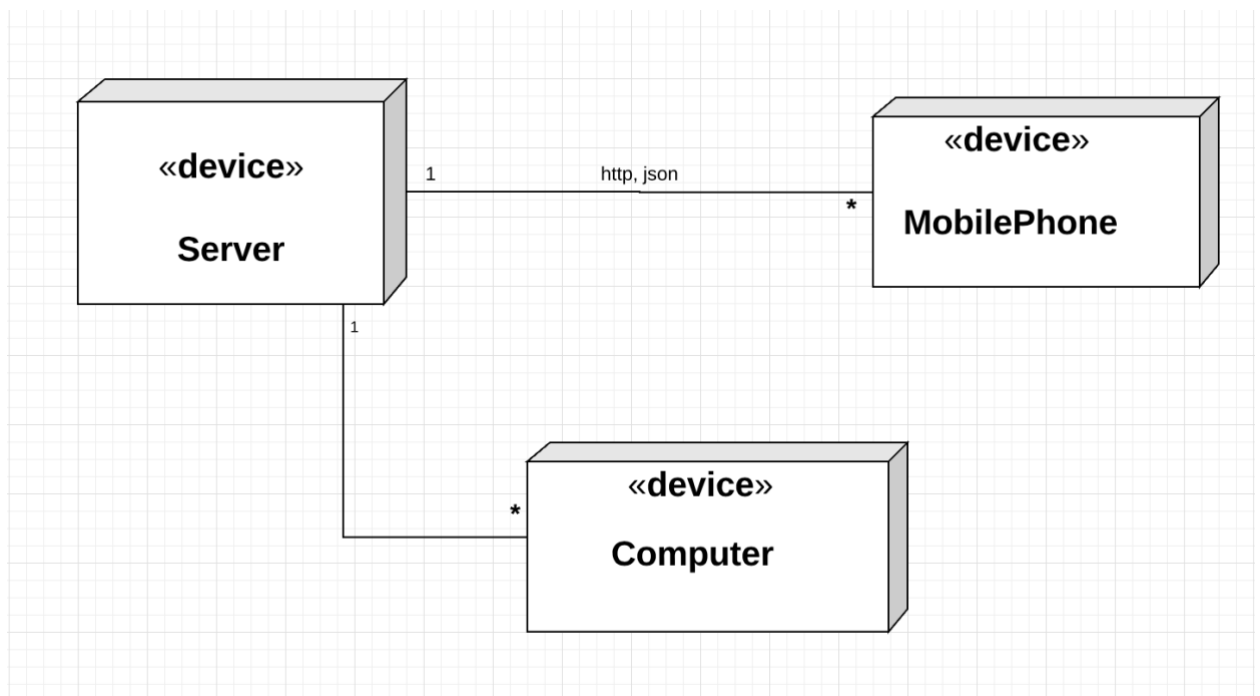


Рисунок 25 – Диаграмма развёртывания

4.3. Взаимодействие с сервером

Для взаимодействия разрабатываемого мобильного приложения с сервером Dreel используется Dreel API.

Для обмена данными используется формат JSON.

JSON основан на двух структурах данных:

- Коллекция пар ключ/значение. В разных языках, эта концепция реализована как объект, запись, структура, словарь, хэш, именованный список или ассоциативный массив.
- Упорядоченный список значений. В большинстве языков это реализовано как массив, вектор, список или последовательность.

Основные API методы, которые используются для взаимодействия:

- desktop_auth – post-запрос. Используется для авторизации пользователя разрабатываемого мобильного приложения.
- manualtime – post-запрос. Используется для получения списка назначенного времени сотруднику, для подтверждения и отклонения заявки назначенного времени сотруднику.
- users_overall – post-запрос. Используется для получения списка всех сотрудников компании.
- users_rate – post-запрос. Используется для изменения ставки сотрудника.
- users_time – post-запрос. Используется для изменения автоотключения таймера сотрудника.
- timer – post – запрос. Используется для включения и выключения таймера.
- company_inviteinfo – post-запрос. Используется для получения ссылки для приглашения сотрудника в сервис.
- projects – post-запрос. Используется для получения списка проектов и сотрудников в соответствующем проекте.
- projects_rate - post – запрос. Используется для получения ставки сотрудника в соответствующем проекте, для изменения ставки сотрудника в соответствующем проекте.
- timesheet – post-запрос. Используется для получения данных для составления отчётов о работе сотрудников.

- `reports_overall` – post-запрос. Используется для получения данных для составления отчётов о работе сотрудников.
- `rerorts_apps` – post-запрос. Используется для получения данных для составления отчётов о работе сотрудников.

4.4. Подготовка окружения

Чтобы начать разработку мобильного приложения, необходимо провести подготовку окружения, для чего требуется выполнить ряд действий в указанном порядке [6]:

1. Установить Node.js
2. Установить Ionic Framework, для чего нужно в командной строке ввести команду:
`npm install -g cordove ionic`
3. Создать стартовый проект, для этого нужно ввести в командную строку: `ionic start myApp tabs`, где `3` слово – название проекта, а `4` – вариант шаблона (Ionic предлагает различные шаблоны, от самых простых до более продвинутых).
4. Чтобы запустить проект для теста в браузере, нужно ввести `ionic serve`, при данной команде приложение откроется в браузере по умолчанию (если воспользоваться командой `ionic serve --lab` в браузере откроется приложение для тестирования, с возможностью просмотра на различных мобильных операционных систем, ниже на рисунке 26 показан пример)

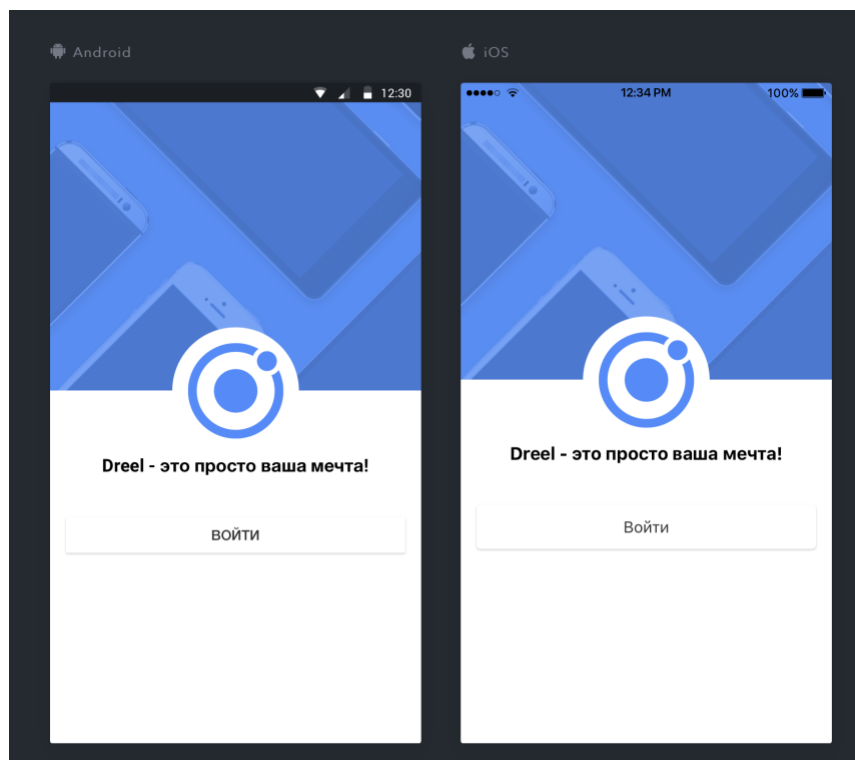


Рисунок 26 – Изображение с примером просмотра на различных мобильных операционных системах

5. Чтобы запустить мобильное приложение на устройстве, необходимо загрузить приложение в облако, которое соберёт приложение и даст к нему доступ командой `ionic upload`. Для непосредственного доступа к приложению на устройстве необходимо установить на телефон приложение Ionic View, данное приложение используется для быстрого тестирования приложения на мобильных устройствах.

После прохождения данных действий, будет готова стартовая сборка проекта к дальнейшей работе с ним.

Для постройки структуры приложения и манипуляциями данным, Ionic использует Javascript и фреймворк AngularJS, который позволяет быстро и удобно взаимодействовать с API. Для визуализации данных используются внутренние HTML5 шаблоны фреймворка Ionic.

4.5. Визуальные компоненты

Одним из преимуществ Ionic Framework является уже существующая огромная библиотека UI компонентов, которые будут выглядеть нативно, как на iOS, так и на Android и соответствующих последним требованиям мобильного дизайна. Также к ним

имеется подробная документация, при помощи которой можно создавать, к примеру, динамические списки, формы, таблицы и другие лаконичные интересные объекты пользовательского интерфейса. Примеры собранных компонентов, реализованных в данной выпускной квалификационной работе:

- Форма для вывода информации о сотруднике и смене ставки и автоотключения таймера (рисунок 27)

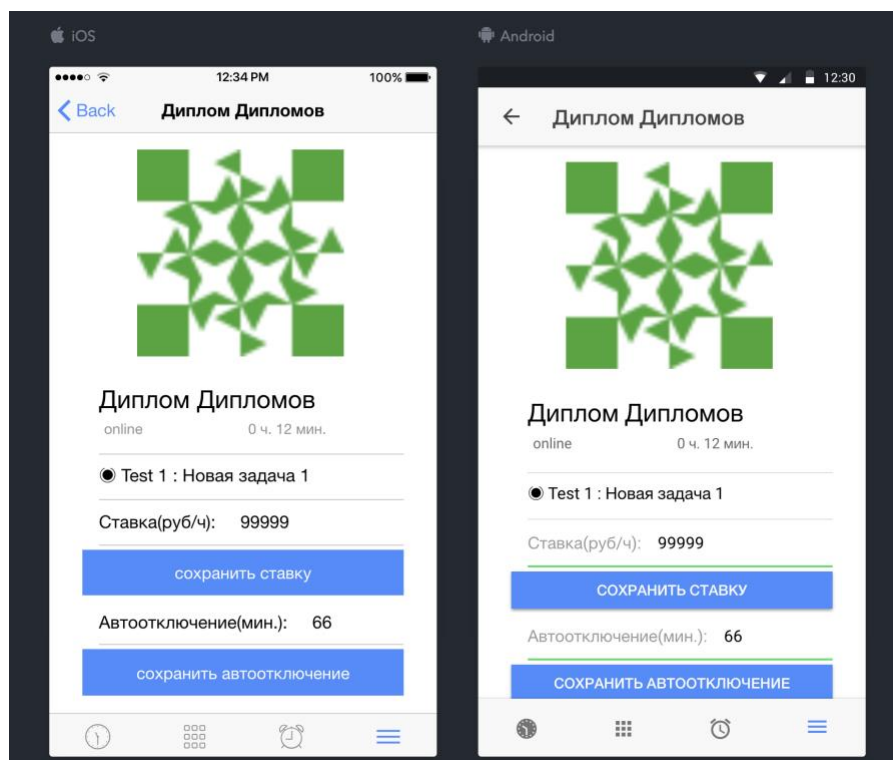


Рисунок 27 – Пример изображения формы для вывода информации о сотруднике, смене ставки сотрудника, автоотключения таймера сотрудника на различных мобильных операционных системах

- Список для вывода сотрудников (рисунок 28)

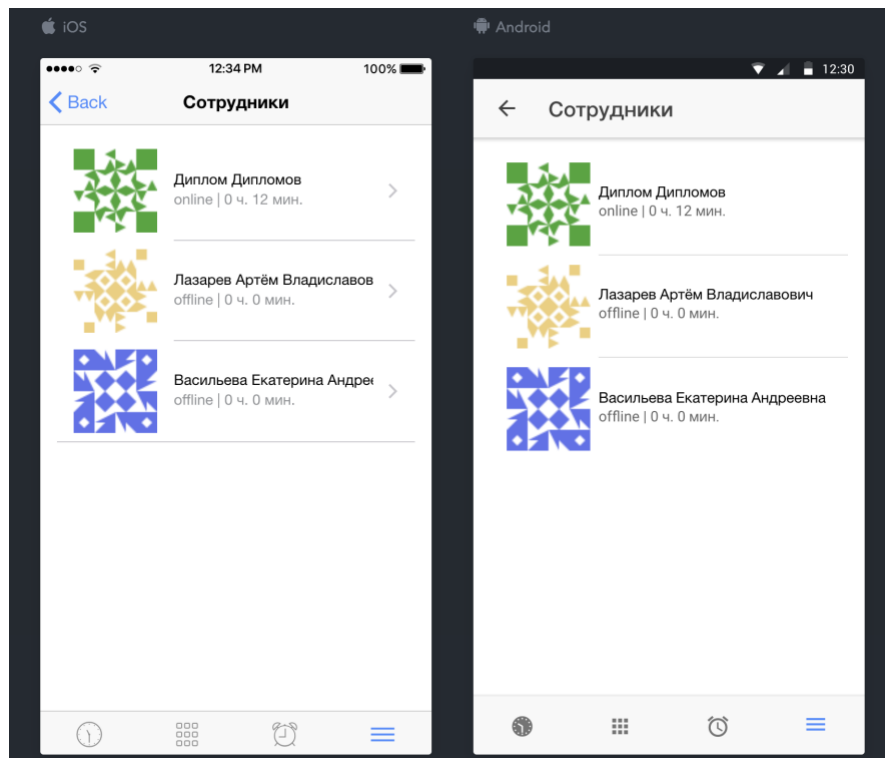


Рисунок 28 – Пример изображения со списком сотрудников на различных мобильных операционных система

- список заявок на подтверждении времени сотрудников (рисунок 29)

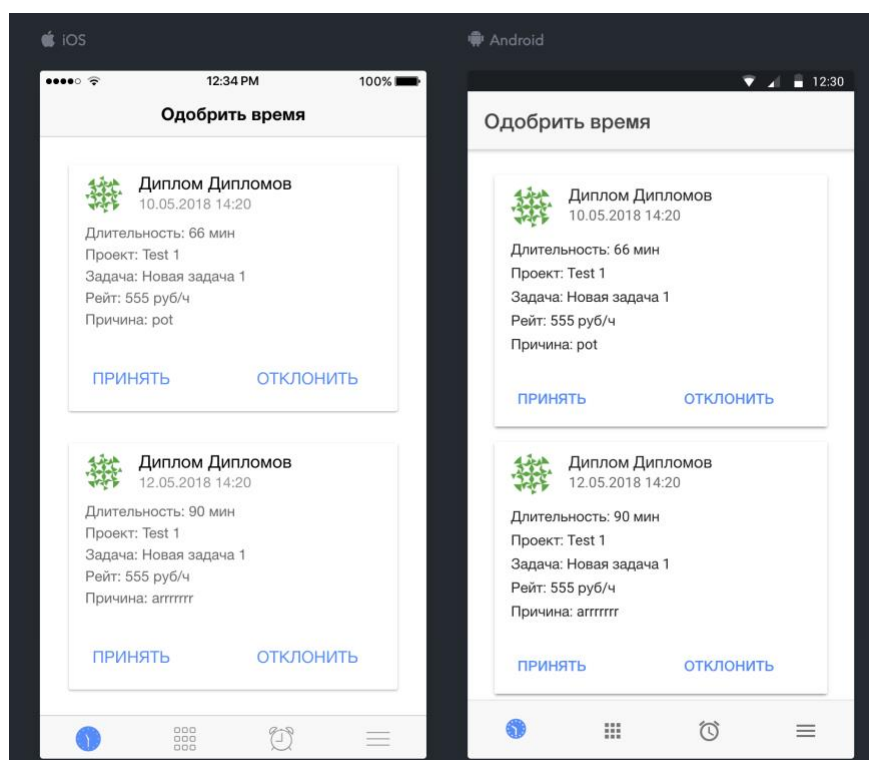


Рисунок 29 – Пример изображения списка заявок на подтверждении времени сотрудников на различных мобильных операционных системах

- Выпадающие списки, для выбора интервала времени, отчёт о сотрудниках, который появляется после выбора интервала (рисунок 30).

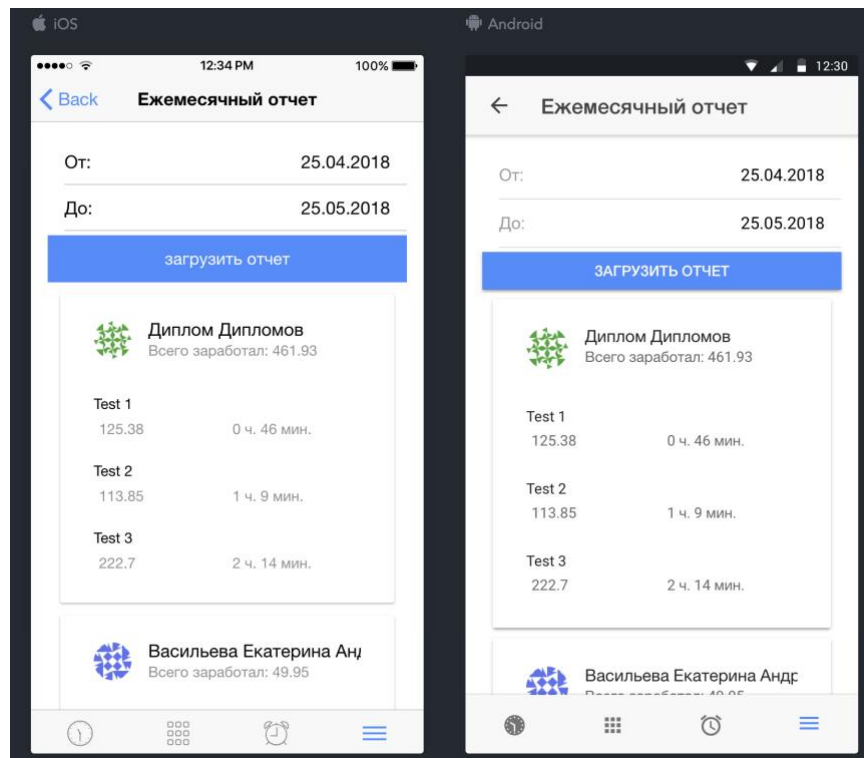


Рисунок 30 – Пример изображения выпадающих списков, для выбора интервала времени и отчёта о сотрудниках, который появляется после выбора интервала на различных мобильных операционных системах

- Выпадающие списки, для выбора сотрудника и интервала времени, отчёт о приложениях и веб сайтах, с потраченным временем на них (рисунок 31).

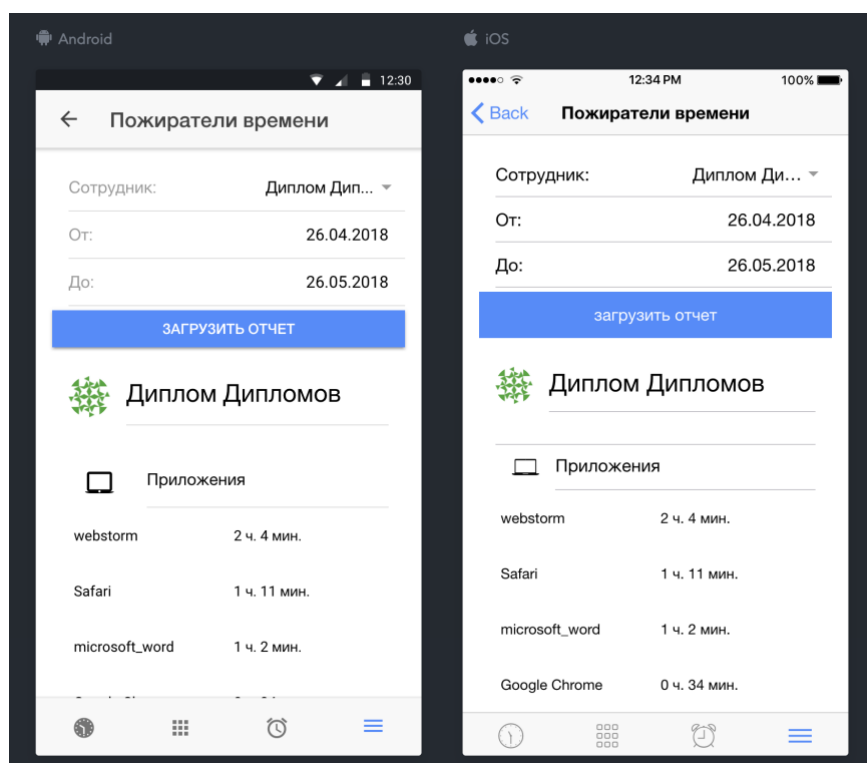


Рисунок 31 – Пример изображения выпадающих списков для выбора сотрудника и интервала времени, отчёт о приложениях, на которые сотрудник потратил время

4.6. Сервисы

В Ionic Framework существует такое понятие, как «providers». В сообществе принято называть их сервисами, которые повторно используются в любой части проекта. Сервисы регистрируются в главном модуле приложения, тогда компоненты, которые будут использовать данный сервис будут использовать один и тот же экземпляр сервиса, если этого не сделать и зарегистрировать сервис в другом модуле, то компоненты, которые будут использовать данный сервис, будут использовать разные экземпляры сервиса.

В данной выпускной квалификационной работе были реализованы такие сервисы, как

- Api – сервис для взаимодействия с сервером, который содержит в себе методы:
 - post– метод принимает 3 параметра: адрес запроса, тело запроса, параметры запроса и отправляет их на сервер, при помощи стандартной функции в Angular.
 - postJson – метод принимает 3 параметра: адрес запроса, тело запроса, параметры запроса и отправляет их на сервер, при помощи стандартной функции в Angular в виде строки.
- Employees – сервис для взаимодействия с сервером, который содержит в себе методы:

- `getItems` – метод принимает на вход токен и отправляет на сервер post-запрос для получения списка сотрудников.
- `refreshRate` – метод принимает на вход 3 параметра: токен, `id`-сотрудника, ставку сотрудника и отправляет на сервер post-запрос для изменения ставки сотрудника.
- `refreshTimeRate` – метод принимает на вход 3 параметра: токен, `id`-сотрудника, время автоотключения таймера сотрудника и отправляет на сервер post-запрос для автоотключения таймера сотрудника.
- **Invite** – сервис для взаимодействия с сервером, который содержит в себе метод:
 - `getLinks` – метод принимает на вход токен и отправляет на сервер post-запрос для получения ссылки на регистрацию.
- **Projects** – сервис для взаимодействия с сервером, который содержит в себе методы:
 - `getItems` – метод принимает на вход токен и отправляет на сервер post-запрос для получения списка проектов.
 - `gerProjectDetail` – метод принимает на вход 2 параметра: токен, `id`-проекта и отправляет на сервер post-запрос для получения информации о соответствующем проекте.
 - `refreshRateProject` – метод принимает на вход 4 параметров: токен, `id`-проекта, `id`-сотрудника, ставку сотрудника в соответствующем проекте и отправляет на сервер post-запрос для изменения ставки соответствующему сотруднику.
- **Reports** – сервис для взаимодействия с сервером, который содержит в себе методы:
 - `getUsers` – метод принимает на вход токен и отправляет post-запрос на сервер для получения списка сотрудников.
 - `loadReport` – метод принимает на вход 3 параметра: токен, начальную границу интервала времени для получения отчёта, конечную границу интервала времени для получения отчёта и отправляет post-запрос на сервер для получения информации для составления отчёта и сотрудников.
 - `loadReportTime` – метод принимает на вход 4 параметров: токен, `id`-пользователя, начальную границу интервала времени для получения отчёта, конечную границу интервала времени для получения отчёта и отправляет post-запрос на сервер для получения информации для составления отчёта о сотруднике.

- `loadReportPillarTime` – метод принимает на вход 3 параметров: токен, id-пользователя, высчитываемое значение промежутка времени и отправляет post-запрос на сервер для получения информации для составления отчёта о сотруднике.
- `Time` – сервис для взаимодействия с сервером, который содержит в себе методы:
 - `getItems` – метод принимает на вход токен и отправляет на сервер post-запрос для получения списка заявок на подтверждение времени.
 - `requestTime` – метод принимает на вход 4 параметров: токен, id-пользователя, id-заявки, значение переменной, отвечающее за подтверждение заявки и отправляет на сервер post-запрос для подтверждения или отклонения заявки на подтверждения времени сотрудника.
- `User` – сервис для взаимодействия с сервером, который содержит в себе методы:
 - `login` – метод принимающий на вход данные для авторизации, содержащие в себе почту и пароль и отправляющие post-запрос на сервер для получения токена.
 - `checkAccessToken` – проверяющий наличия токена в `localStorage` устройства.
 - `saveAccessToken` – метод принимающий на вход токен и сохраняющий его в `localStorage` устройства.
- `ClickBoard` – сервис, реализующий копирование строки в буфер обмена устройства, который содержит в себе метод:
 - `copy` – метод принимающий на вход строку и возвращающий `promise` (`Promise` (обещание) представляет собой обертку для значения, неизвестного на момент создания обещания).

5. Руководство пользователя

Далее будут представлены изображения экранов на различных мобильных операционных системах разработанного мобильного приложения в рамках данной выпускной квалификационной работы.

Основное меню сделано в нижней части экрана, с «табов», которых 4:

1. Список подтверждения времени
2. Список проектов
3. Таймер
4. Дополнительное меню

На рисунке 32 представлен начальный экран, на котором присутствует анимация с логотипом сервиса.

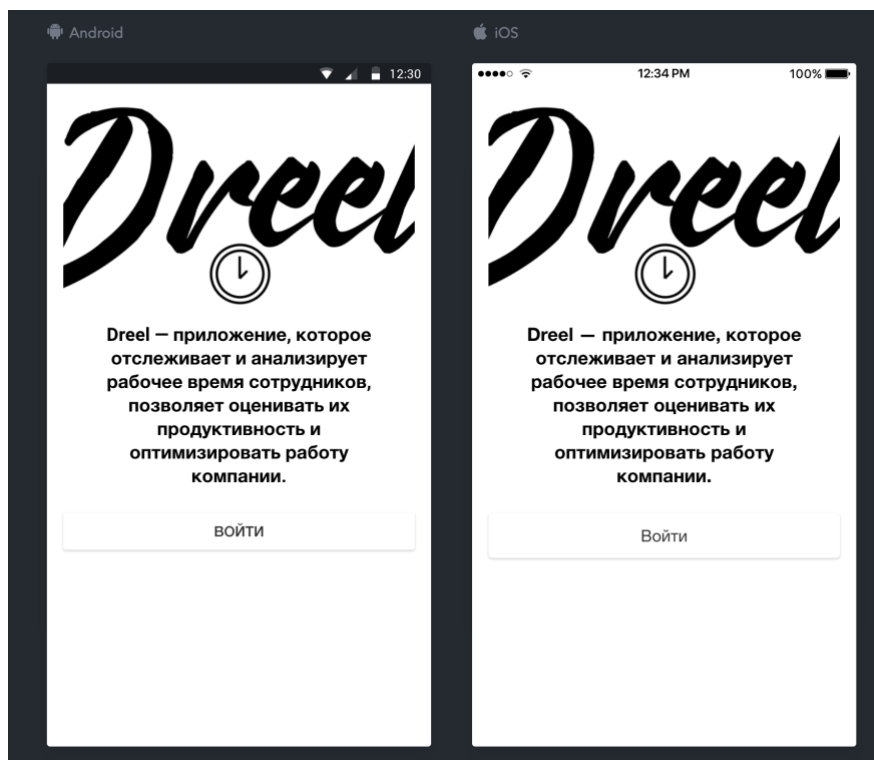


Рисунок 32 – Изображения с экраном начальной страницы

На рисунке 33 представлена экран с формой для авторизации.

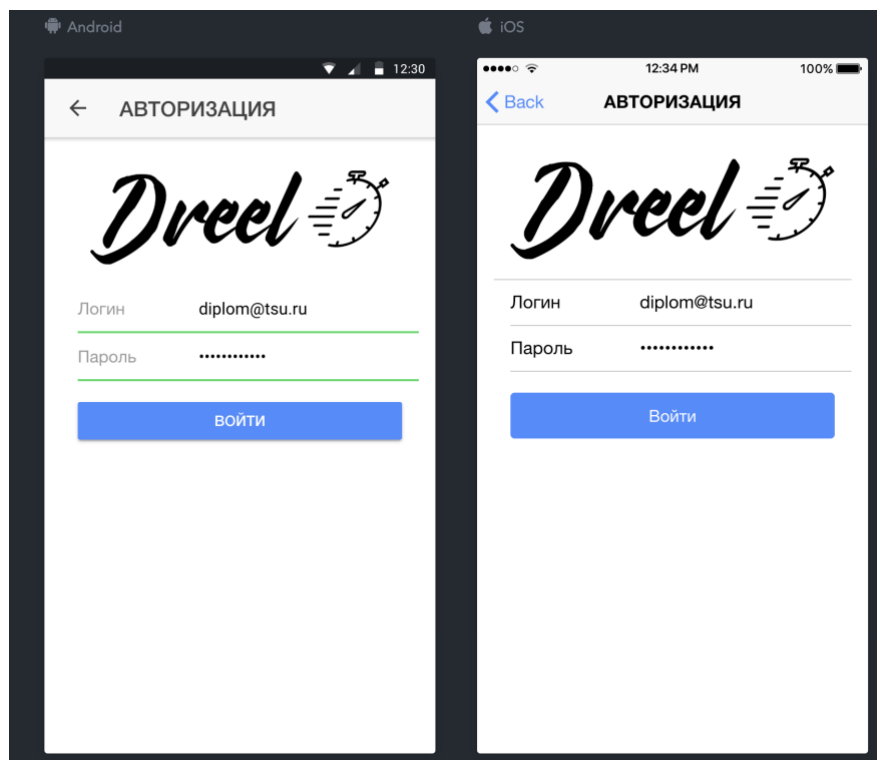


Рисунок 33 – Изображения с экраном авторизации

На рисунке 34 представлен список, состоящий из заявок на подтверждение времени, с возможностью подтверждения и отклонения заявки, при нажатии на соответствующую кнопку.

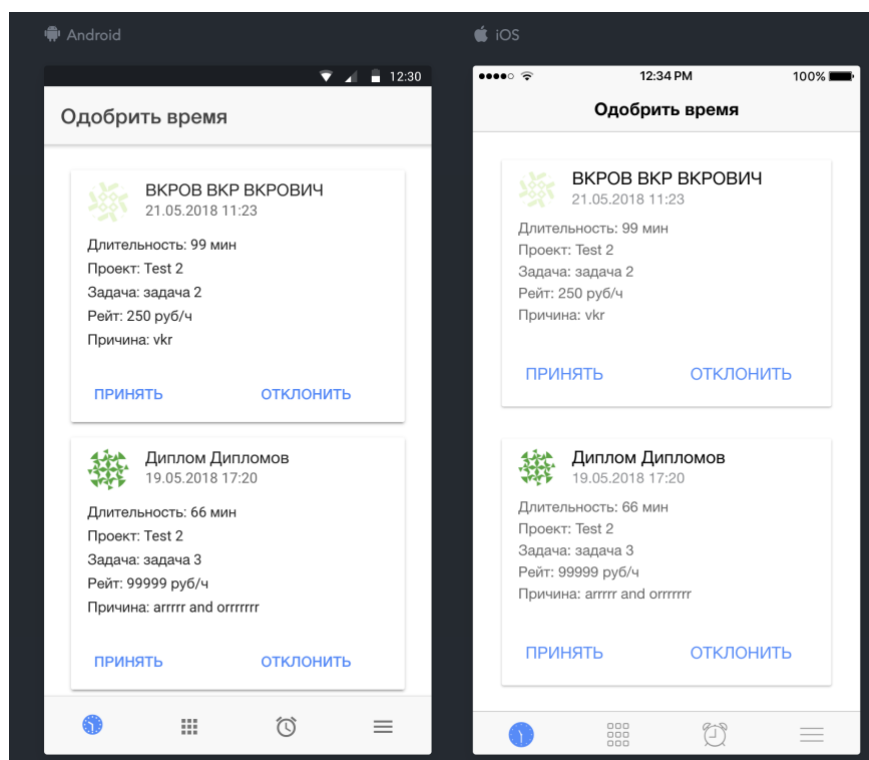


Рисунок 34 – Изображения с экраном списком подтверждения заявок на время

На рисунке 35 представлен список проектов с возможностью перехода к списку сотрудников проекта при нажатии кнопки «подробнее» у соответствующего проекта.

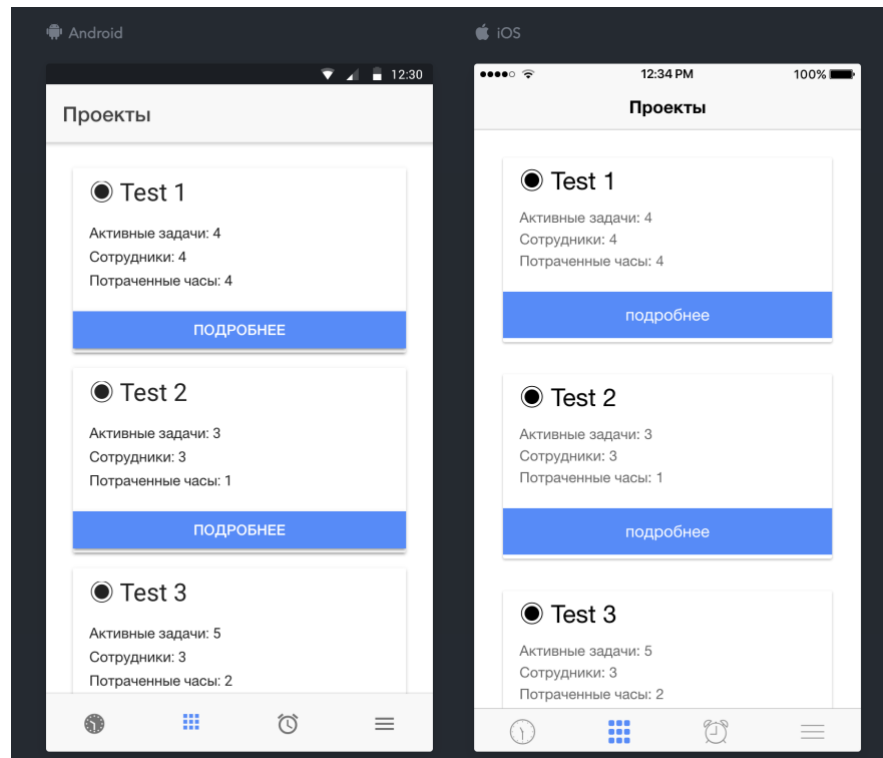


Рисунок 35 – Изображения с экраном списка проектов

На рисунке 36 представлен список сотрудников выбранного проекта, с возможностью изменения ставки в проекте у сотрудника, при введении в поле ставки нужного значения и нажатия на кнопку «сохранить ставку в проекте»

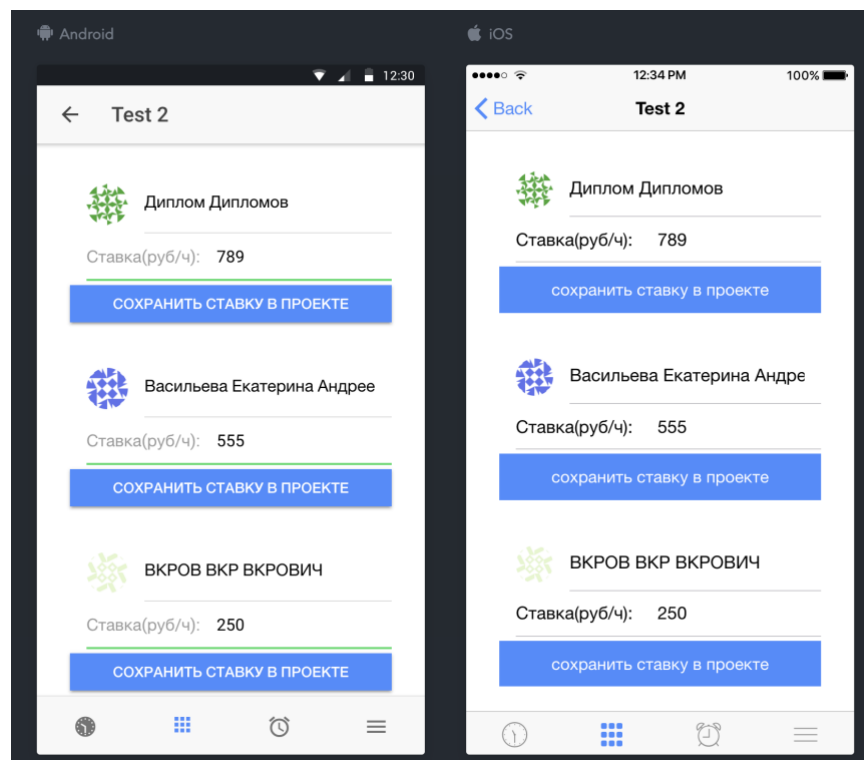


Рисунок 36 – Изображения с экраном списка сотрудников в соответствующем проекте

На рисунке 37 представлен таймер, при нажатии на кнопку «start» таймер запускается, при нажатии на кнопку «stop» останавливается

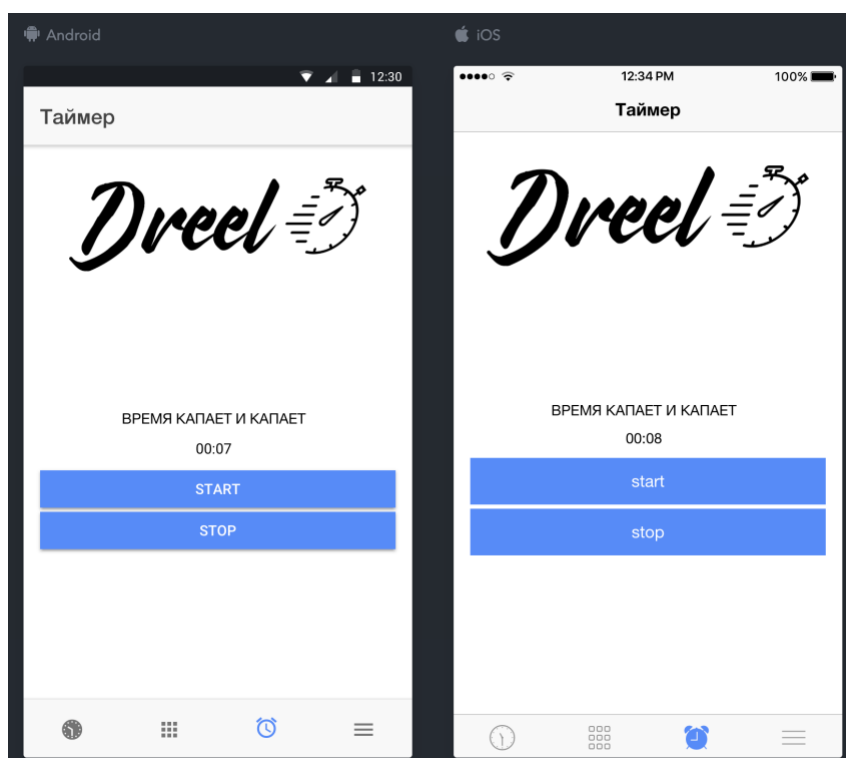


Рисунок 37 – Изображения с экраном таймера

На рисунке 38 представлен список дополнительного, с возможностью перехода в другие дополнительные разделы и выхода из своего аккаунта, при нажатии на соответствующую кнопку.

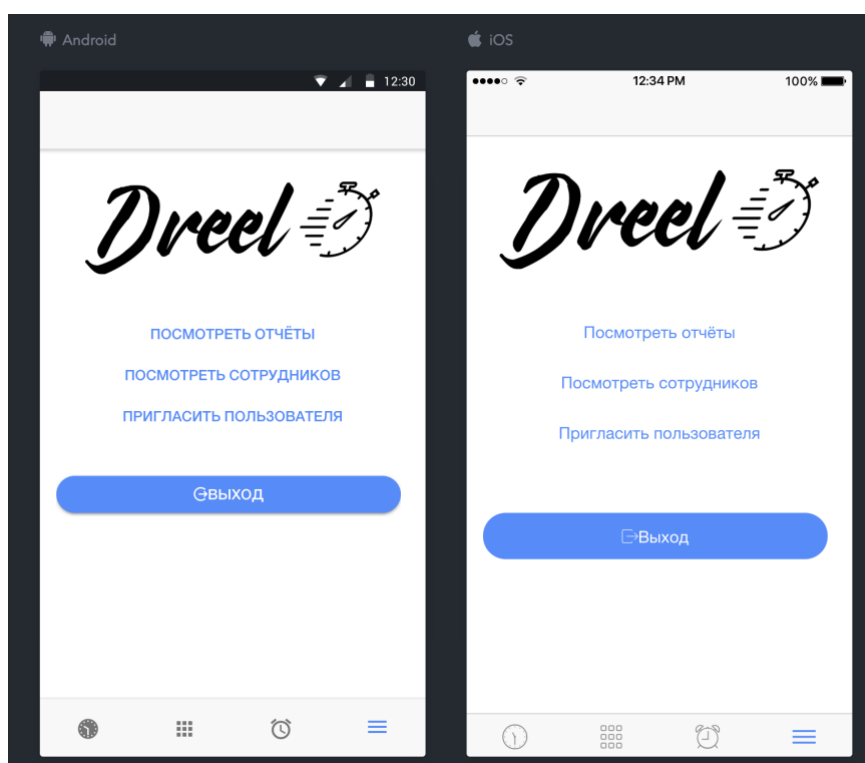


Рисунок 38 – Изображения с экраном списка дополнительного меню, с возможностью перехода в другие разделы и выходом из своего аккаунта

На рисунке 39 представлен список возможных отчётов, с возможностью просмотра нужного отчёта, при нажатии на нужную кнопку

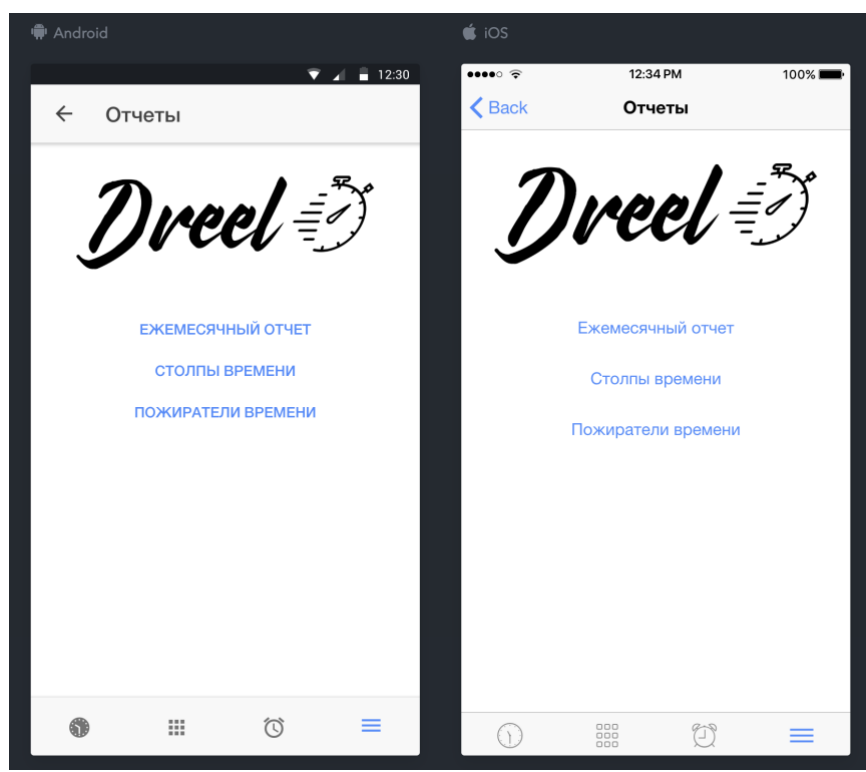


Рисунок 39 – Изображения с экраном списка возможных отчётов

На рисунке 40 представлен ежемесячный отчёт по всем сотрудникам, с возможностью задать нужный временной интервал для отчёта. При нажатии на кнопку «загрузить отчёт», данные на странице будут изменяться в зависимости от поставленного временного интервала для отчёта.

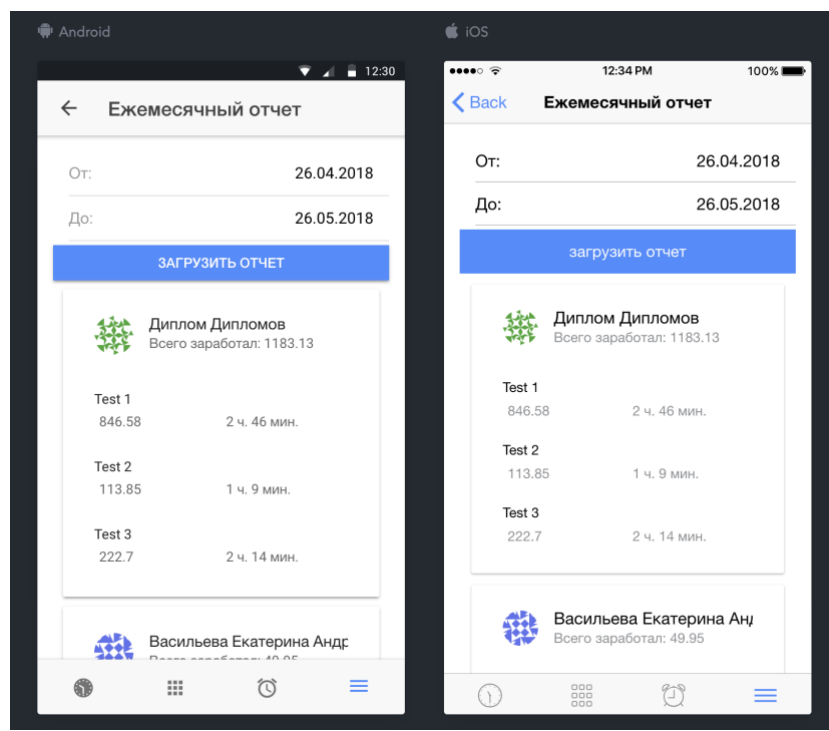


Рисунок 40 – Изображения с экраном ежемесячного отчёта

На рисунке 41 представлен отчёт по времени за последнюю неделю, с возможностью выбора любого сотрудника. На графике видно в какой день сотрудник провёл больше всего времени на работе. При нажатии на кнопку «загрузить отчёт», данные на странице и график будут изменяться в зависимости от поставленного сотрудника для отчёта.

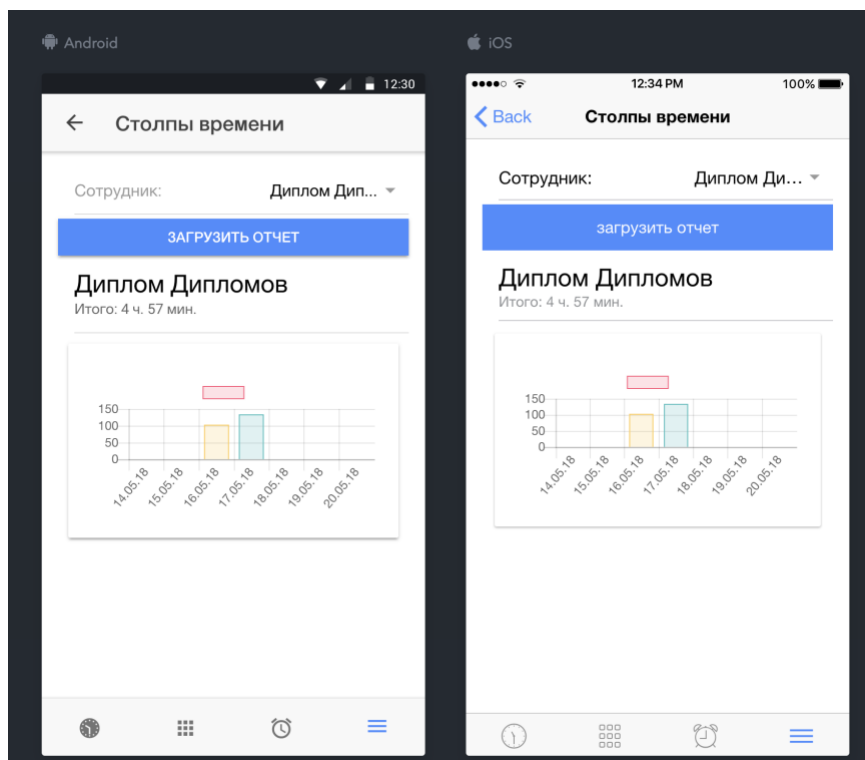


Рисунок 41 – Изображения с экраном отчёта по времени

На рисунке 42-43 представлен отчёт по использованию приложений сотрудником и сайтов, на которые он заходил. С возможностью поставить временной интервал в полях и выбором нужного сотрудника. При нажатии на кнопку «загрузить отчёт», данные на странице и график будут изменяться в зависимости от поставленного сотрудника и временного интервала для отчёта.

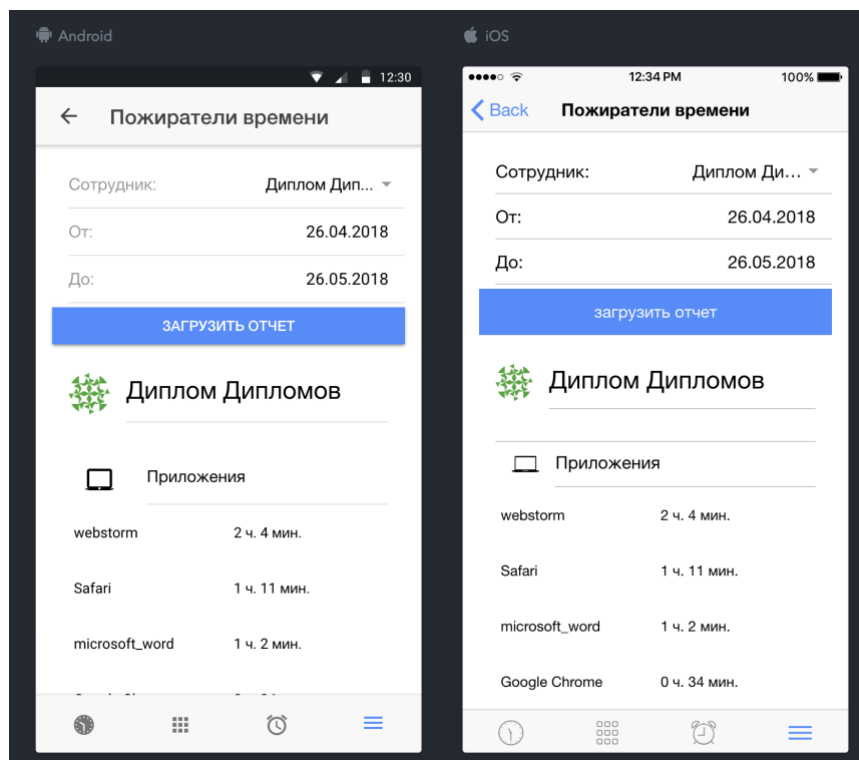


Рисунок 42 – Изображения с экраном отчёта по использованию приложений сотрудником и сайтов, на которые заходил сотрудник

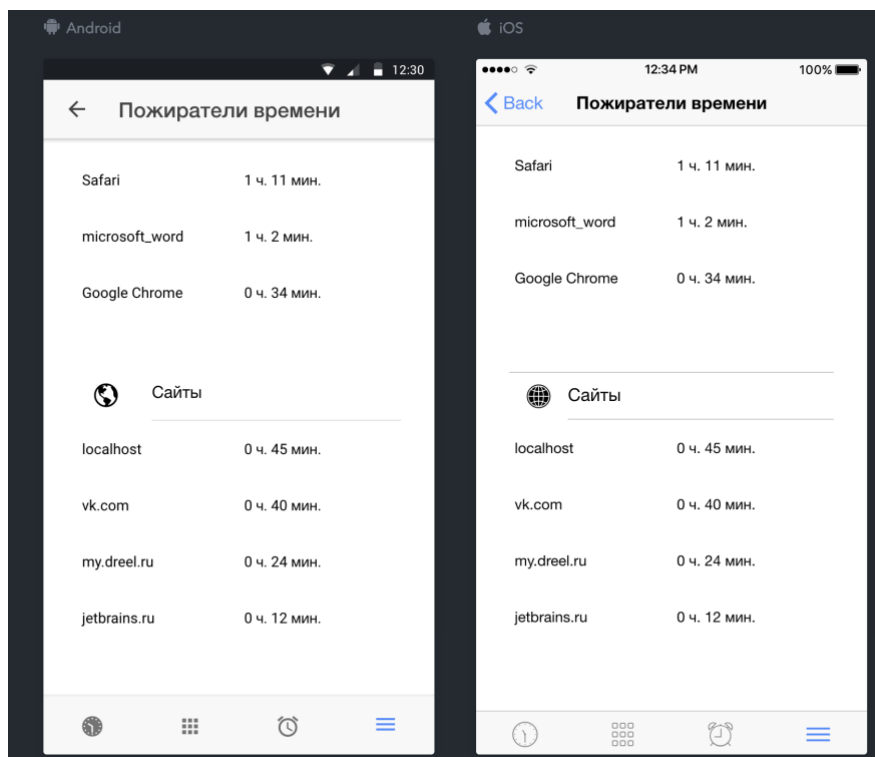


Рисунок 43 – Изображения с экраном отчёта по использованию приложений сотрудником и сайтов, на которые заходил сотрудник

На рисунке 44 представлен список сотрудников, с возможностью перехода в подробную информацию о сотруднике, при нажатии на соответствующего сотрудника.

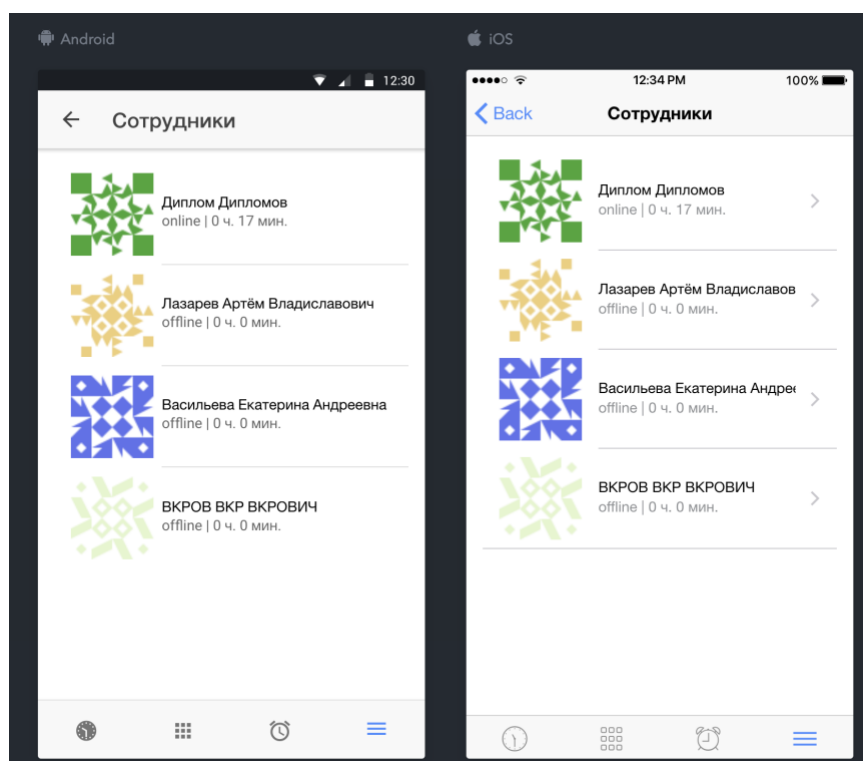


Рисунок 44 – Изображения с экраном списка сотрудников

На рисунке 45 представлена форма с детальной информацией о сотруднике, с возможностью изменения ставки сотрудника и автоотключение его таймера, при введении нужного значения и нажатия соответствующей кнопки.

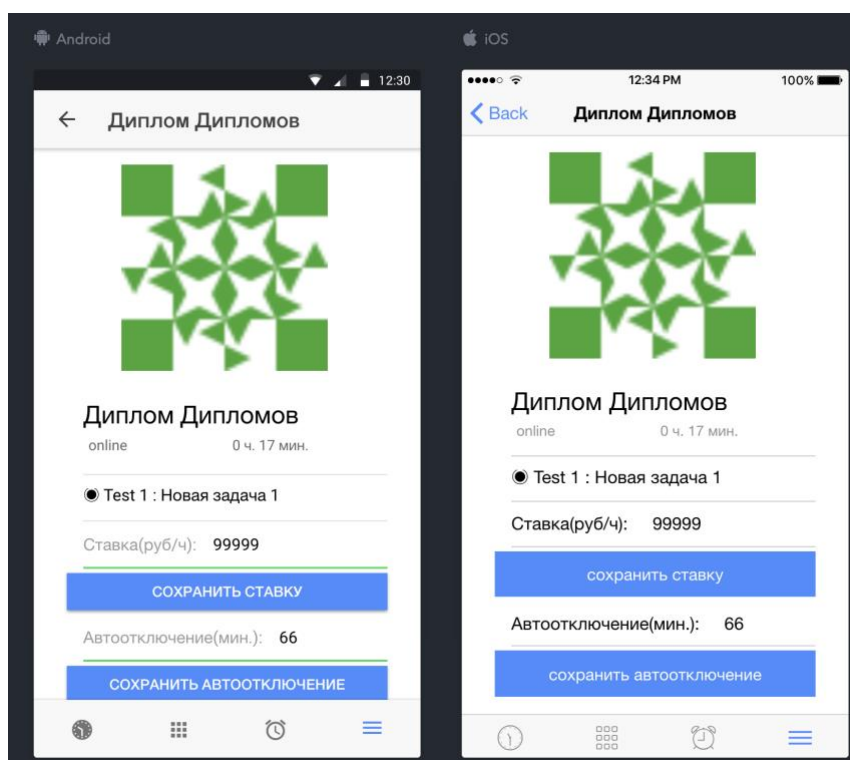


Рисунок 45 – Изображения с экраном детальной информации о соответствующем сотруднике

На рисунке 46 представлена ссылка для приглашения на регистрацию в сервисе для соответствующей компании. При нажатии на кнопку «копировать ссылку» ссылка скопируется в буфер обмена мобильного устройства.

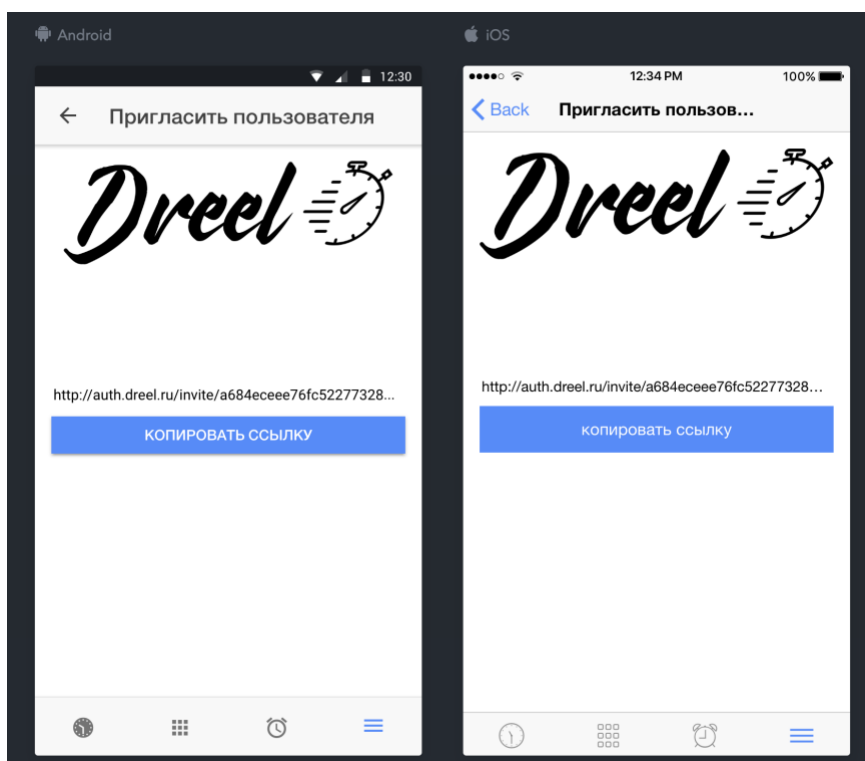


Рисунок 46 – Изображения с экраном ссылки для регистрации

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе была представлена разработка кроссплатформенного мобильного приложения для мониторинга работы сотрудников.

В ходе проделанной работы были решены следующие задачи:

- Анализ существующих решений.
- Проектирование вариантов использования.
- Разработка архитектуры приложения.
- Проектирование интерфейса приложения.
- Разработка приложения.

Мобильное приложение, разработанное в рамках данной квалификационной работы, используется ООО «Редлайн».

На сегодняшний день приложение проходит внутреннее тестирование для ограниченного круга лиц. Приложение не будет выложено в открытый магазин, оно предназначено только для пользователей сервиса.

В дальнейшем планируется расширение приложения. Оно будет предназначено не только для руководителей, но и для сотрудников с функционалом, востребованным для них.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Википедия – свободная энциклопедия [Электронный ресурс] – URL: <https://ru.wikipedia.org> (дата обращения: 25.05.2018)
2. Система учёта рабочего времени Dreel [Электронный ресурс] – URL: <https://dreel.ru> (дата обращения: 17.04.2018)
3. Система учёта рабочего времени TimeDoctor [Электронный ресурс] – URL: <https://www.timedoctor.com/ru/> (дата обращения: 23.04.2018)
4. Система учёта рабочего времени Yaware TimeTracker [Электронный ресурс] – URL: <https://yaware.ru> (дата обращения: 17.04.2018)
5. Chart.js [Электронный ресурс] – URL: <http://www.chartjs.org> (дата обращения: 15.05.2018)
6. Ionic Framework [Электронный ресурс]. – URL: <https://ionicframework.com> (дата обращения: 20.05.2018)
7. WebStorm [Электронный ресурс] – URL: <https://www.jetbrains.com/webstorm/> (дата обращения: 02.05.2018)

ПРИЛОЖЕНИЕ А



помогаем бизнесу
продавать онлайн

ООО «Редлайн»
ул. Карла Маркса 17а, 4 этаж
Тел.: 8 (3822) 99-41-30
info@redlg.ru
www.redlg.ru

Актуальность дипломной работы

Для Хованского Якова Александровича

Dreel – программа для контроля и анализа рабочего времени сотрудников. Изначально сервис разрабатывался для решения внутренних задач компании, но разработкой заинтересовались и другие компании.

Сервисом пользуются десятки компаний и сотни сотрудников ежедневно на компьютерах под управлением операционных систем: Windows, MacOS и Ubuntu.

Но у компании до сих пор нет приложения для мобильных устройств.

Пользователи сервиса просят разработать мобильное приложение для оперативного мониторинга сотрудников и учёта рабочего времени вне офиса.

В ходе телефонных интервью мы выяснили, что 70% пользователей выразили желание пользоваться мобильным приложением для просмотра отчетов и учета времени вне офиса.

В случае успешного выполнения работы мы планируем трудоустроить Хованского Якова Александровича на работу для дальнейшего развития мобильного приложения.

Директор ООО «Редлайн»



Н.А. Воеводин

ПРИЛОЖЕНИЕ Б



помогаем бизнесу
продавать онлайн

ООО «Редлайн»
ул. Карла Маркса 17а, 4 этаж
Тел.: 8 (3822) 99-41-30
info@redlg.ru
www.redlg.ru

от 29 «мая» 2018

АКТ

о внедрении результатов дипломной работы

Кроссплатформенное мобильное приложение для мониторинга работы сотрудников сервиса Dreel, разработанное студентом Томского Государственного Университета Хованским Яковом Александровичем в период прохождения им преддипломной практики и подготовки к защите дипломной работы: «Разработка кроссплатформенного мобильного приложения для мониторинга работы сотрудников», внедрено и используется в деятельности ООО «Редлайн».

Использование мобильного приложения повысило удобство использования программного продукта Dreel.

Директор ООО «Редлайн»



Н.А. Воеводин

Отчет о проверке на заимствования №1

Автор: Хованский Яков yakovkhovansky@gmail.com / ID: 5659840

Проверяющий: Хованский Яков (yakovkhovansky@gmail.com) / ID: 5659840)

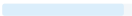
Отчет предоставлен сервисом «Антиплагиат»- <http://www.antiplagiat.ru>

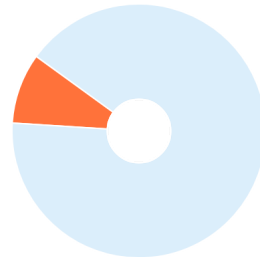
ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 2
Начало загрузки: 07.06.2018 11:26:39
Длительность загрузки: 00:00:02
Имя исходного файла: ВКР отчет_т
Хованский_
Размер текста: 4377 кБ
Символов в тексте: 48234
Слов в тексте: 5451
Число предложений: 229

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 07.06.2018 11:26:41
Длительность проверки: 00:00:01
Комментарии: не указано
Модули поиска:

ЗАИМСТВОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
9,05% 	0% 	90,95% 



Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.

Заимствования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Ссылка	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	2,3%	2,3%	JetBrains WebStorm 2016.1.3 ...	http://filegood.ru	17 Июл 2017	Модуль поиска Интернет	7	7
[02]	2,09%	2,09%	Разработка мобильного при...	http://swsys.ru	19 Ноя 2016	Модуль поиска Интернет	6	6
[03]	1,84%	1,84%	«FREE AND OPEN SOURCE SO...	http://docplayer.net	11 Июл 2017	Модуль поиска Интернет	9	9

Еще источников: 7

Еще заимствований: 2,82%