

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»  
Кафедра IT-инжиниринг

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий кафедрой

PhD, доцент

\_\_\_\_\_ Т.С. Картбаев  
« \_\_\_\_ » \_\_\_\_\_ 2019 г.

**ДИПЛОМНЫЙ ПРОЕКТ**

На тему: Разработка системы слежения и поиска мобильных устройств

Специальность 5В070300 – «Информационные системы»

Выполнил Иембергген А.М. Группа ИС-15-2

Научный руководитель: PhD, доцент Картбаев Т.С.

Консультанты:

по экономической части: к.э.н., доцент \_\_\_\_\_ А.И.Бекишева  
« 23 » \_\_\_\_\_ 05 \_\_\_\_\_ 2019 г.

по безопасности жизнедеятельности: д.т.н., ст.преп \_\_\_\_\_ Ш.Ш.Бекбасаров  
« 22 » \_\_\_\_\_ мая \_\_\_\_\_ 2019 г.

Программное обеспечение: ст. преп. \_\_\_\_\_ М.Н.Майкотов  
« 14 » \_\_\_\_\_ \_\_\_\_\_ 2019 г.

Нормоконтролер: ст. преп. \_\_\_\_\_ Ж.К.Алимсеитова  
« 23 » \_\_\_\_\_ 05 \_\_\_\_\_ 2019 г.

Рецензент: PhD, ассистент профессора кафедры «Компьютерная инженерия и телекоммуникации» МУИТ \_\_\_\_\_ Н. К. Мукажанов  
« \_\_\_\_ » \_\_\_\_\_ 2019 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий  
Кафедра IT-инжиниринг  
Специальность 5В070300 – Информационные системы

**Задание**

на выполнение дипломного проекта

Студенту Иембергелу Айдыну Марленулы

Тема проекта: Разработка системы слежения и поиска мобильных устройств

Утверждена приказом по университету № 124 от «26» октября 2019 г.

Срок сдачи законченного проекта «1» мая 2019 г.

Исходные данные к проекту, требуемые параметры результатов проектирования и исходные данные объекта:

- а) универсальный инструмент для создания прототипов «Justinmind»;
- б) среда разработки Atom для создания программной части;
- в) фреймворк Django;
- г) базы данных MySQL 3.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) описание предметной области;
- б) описание исходной ситуации;
- в) обоснование средств и технологий;
- г) проектирование и разработка приложения;

Перечень графического материала (с точным указанием обязательных чертежей): Представлены 8 таблиц, 81 иллюстрация.

Основная рекомендуемая литература:

1. James Chambers, David Paquette, Simon Timms Python Application Development: Building an application in four sprints (Developer Reference), 2016;
2. Andrew Lock Python in Action, 2018;
3. Michael Fitzgerald Introducing Regular Expressions, 2012;
4. Джозеф А., Бен А. Python 6.0. Карманный справочник, 2015;
5. Mark J. Price MySQL 3 and Django: Modern Cross-Platform Development, 2016.

Консультация по работе (проекту) с указанием относящихся к ним разделов работы (проекта)

Раздел	Консультант	Сроки	Подпись
Экономическая часть	А.И.Бекишева	02.05.2019 23.05.2019	
Безопасность жизнедеятельности	Ш.Ш.Бекбасаров	27.07.2019 22.05.2019	
Программное обеспечение	М.Н.Майкотов	02.05.19 22.05.19	
Нормоконтролер	Ж.К.Алимсеитова	1.03.19-22.03.19	

График  
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Основная часть	14.01.2019 30.01.2019	выполнено
Специальная часть	05.02.2019 12.03.2019	выполнено
Ожидаемые технико-экономические показатели	08.05.2019 13.05.2019	выполнено
Безопасность жизнедеятельности	04.05.2019 14.05.2019	выполнено

Дата выдачи задания «26» октябрь 2019 г.

Заведующий кафедрой \_\_\_\_\_ Т.С. Картбаев  
(подпись)

Научный руководитель работы (проекта) \_\_\_\_\_ Т.С. Картбаев  
(подпись)

Задание принял к исполнению студент \_\_\_\_\_ А.М. Иемберген  
(подпись)

## **Аннотация**

Тема: «Разработка системы слежения и поиска мобильных устройств».

Целью дипломного проекта является разработка приложения для отслеживания геолокации смартфона в режиме реального времени. При этом основная задача, выполняемая в данном дипломном проекте – разработать функционал, который даст возможность управлять смартфоном и получать его данные дистанционно.

Дипломный проект включает в себя 5 разделов. Введение содержит актуальность, цели и задачи разработки. Первый раздел содержит анализ предметной области и существующего ПО. Второй раздел содержит постановку задачи и обзор возможных путей их решения, а также обоснование выбора средств и технологий. Этапы разработки ПО описаны в третьем разделе. Четвертый раздел – ожидаемые технико-экономические показатели. Пятый раздел посвящен рассмотрению мероприятий по охране труда.

## **Аңдатпа**

Тақырыбы: «Жоғалған мобильді құрылғылардың ізін бақылау және іздестіру жүйесін әзірлеу».

Дипломдық жобаның мақсаты - нақты уақыт режимінде смартфонның геолокациясын қадағалауға арналған қолданба. Сонымен қатар, осы дипломдық жобаның басты міндеті - смартфонды бақылауға және деректерді қашықтан алуға мүмкіндік беретін функционалды әзірлеу болып табылады.

Дипломдық жоба 5 бөлімнен тұрады. Кіріспе дамудың өзектілігі, мақсаттары мен міндеттерін қамтиды. Бірінші бөлім тақырыптық аумақты және қолданыстағы бағдарламалық қамтаманы талдауды қамтиды. Екінші бөлікте мәселенің тұжырымдамасы және оны шешудің ықтимал жолдарын шолу, сондай-ақ құралдар мен технологияларды іріктеудің негіздемесі бар. Бағдарламалық жасақтаманы әзірлеу кезеңдері үшінші бөлімде сипатталған. Төртінші бөлім - күтілетін техникалық және экономикалық көрсеткіштер. Бесінші секция еңбекті қорғау жөніндегі шараларды қарауға арналған..

## **Annotation**

Theme: “Development of a tracking and retrieval system for lost mobile devices”.

The aim of the graduation project is to develop application for tracking the geolocation of smartphones in real time. The task is to be able to control smartphones and retrieve their data remotely.

The graduation project includes 5 sections. The introduction contains relevance, goals and development objectives. The first section contains an analysis of the subject area and existing software. The second section contains setting tasks and an overview of ways to solve them, as well as the rationale for the selection of tools and technologies. Stages of software development are available in the third section. The fourth section - the expected technical and economic indicators. The fifth section is devoted to the consideration of measures for labor protection.

## Содержание

Введение.....	9
1 Основная часть .....	10
1.1 Анализ предметной области.....	10
1.2 Цели исследования .....	10
1.3 Обзор существующих решений .....	11
1.3.1 Умные часы .....	11
1.3.2 Мобильные приложения .....	12
1.3.3 Веб-приложения.....	14
1.4 Концептуальная основа.....	14
1.5 Постановка задач .....	15
1.6 Обоснование выбора средств и технологий .....	16
1.6.1 Веб-фреймворк Django .....	16
1.6.2 Фреймворк Flask.....	17
1.6.3 Язык программирования Python.....	18
1.6.4 Atom для Python .....	19
1.6.5 База данных MySQL .....	20
1.6.6 API Яндекс.Карт.....	20
2 Специальная часть.....	21
2.1 Проектирование системы слежения средствами UML.....	21
2.2 Использование Atom в качестве редактора Python .....	23
2.2.1 Установка Atom.....	24
2.2.2 Обновление Atom.....	26
2.2.3 Настройки прокси и брандмауэра .....	26
2.2.4 Пакеты Atom.....	26
2.2.5 Выполнение файла Python в Atom .....	27
2.3 Управление пакетами .....	30
2.4 Запуск программы без привязки к виртуальному окружению .....	30
2.5 Создание виртуальной среды .....	31
2.6 Привязка к базе данных SQL 3.....	33
2.7 Автоматизация геолокации .....	34
2.8 Определение IP и местоположения .....	36
2.9 Отслеживание IP адреса.....	38
2.10 HTML привязка к шаблону.....	39
2.11 Отслеживание.....	41
2.12 Стартовая страница .....	43
2.13 Интерфейс.....	48
3 Ожидаемые технико-экономические показатели .....	50
3.1 Расчет трудоемкости разработки программного продукта.....	51
3.2 Расчет затрат на разработку программного продукта .....	52
3.3 Определение возможной (договорной) цены программного обеспечения.....	56
3.4 Расчет срока окупаемости программного продукта .....	57
3.5 Оценка социально-экономических результатов.....	57

4 Безопасность жизнедеятельности.....	59
4.1 Анализ условий труда .....	59
4.2 Расчет тепловых нагрузок в помещении.....	60
4.3 Расчет внутренних тепловых нагрузок в помещении.....	62
4.4 Расчет теплового баланса помещения.....	63
4.5 Выбор кондиционера. Схема расположения .....	64
4.6 Вывод .....	65
Заключение... ..	66
Список литературы .....	67
Приложение А – Техническое задание .....	<b>Ошибка! Закладка не определена.</b>
Приложение Б – Листинг программы .....	<b>Ошибка! Закладка не определена.</b>
Приложение В – Акт внедрения .....	<b>Ошибка! Закладка не определена.</b>



## **Введение**

Обеспечение безопасности детей, пожилых людей, а также людей с ограниченными возможностями имеет решающее значение для сокращения случаев похищения и потерь как их самих, так и их смартфонов. Любой пропавший человек подвергается риску, который варьируется от наркотиков, плохого состояния здоровья, убийства, изнасилования или даже заражения смертельными заболеваниями. Во всем мире предпринимаются усилия по обеспечению защиты таких категорий населения, особенно в общественных местах, например, в выставках и детских развлекательных зонах. Носимые устройства с GPS-отслеживанием, которые могут точно определить местоположение человека, используя карту на смартфоне или планшете, и являются частью решений, предлагаемых для отслеживания [1].

Несмотря на эти усилия, все еще существуют проблемы с точки зрения количества организаций, которые приняли решения по отслеживанию людей, особенно в развивающихся странах. Это связано с тем, что многие из этих решений довольно дорогие, как программные, так и аппаратные. Кроме того, интеграция решений по отслеживанию людей не легка, так как большинство из них построены как автономные решения.

Поэтому работа направлена на разработку решения для отслеживания человека, которое снижает стоимость инфраструктуры за счет использования существующей беспроводной локальной сети.

Практическая значимость дипломного проекта заключается в предотвращении несчастных случаев, связанных с потерей или похищением детей, пожилых людей и людей с ограниченными возможностями.

## **1 Основная часть**

### **1.1 Анализ предметной области**

Эта глава содержит критический анализ взаимосвязи между различными работами с моим дипломным проектом. Работа, сделанная различными авторами, имеет действительно важное значение для создания основы для исследований и предоставляет информацию о том, что было сделано другими людьми для решения данной проблемы.

В двадцать первом веке, с развитием технологий, современное общество, включая даже маленьких детей использует самые различные гаджеты, смартфоны. И в связи с этим, растет и риск кражи мобильных телефонов, или уж тем более кражи детей. Многим родителям становится трудно работать и в то же время заботиться о своих малышах в дневное время. Таким образом, кроме наличия няни на полный рабочий день, нужно обеспечить ребенку полноценный уход и контроль.

Степень кражи смартфонов детей не только в общественных местах, но и в укромных местах просто показывает, насколько важно иметь системы наблюдения за ребенком, чтобы отслеживать каждого ребенка. Это решение предназначено главным образом для больничных учреждений и детских садов, но может быть легко интегрировано и в другие среды.

За прошедшие годы технологии значительно продвинулись. Для отслеживания людей были созданы различные решения. Использование технологии для отслеживания детей дает много преимуществ. За последние несколько лет индустрия детского GPS-отслеживания значительно продвинулась и является довольно новым применением информационных технологий. Некоторые из этих систем GPS-отслеживания могут определять частоту сердечных сокращений, артериальное давление и температуру тела ребенка. Эта информация может помочь родителям узнать, чувствует ли их ребенок себя комфортно или у него есть чувство страха и дискомфорта, указывающее на то, что у него могут быть проблемы. Новые устройства используют GPS, Wi-Fi и другие технологии отслеживания местоположения и могут быть связаны с приложениями на телефоне родителя. С помощью технологий можно значительно упростить отслеживание детей и, следовательно, облегчить бремя родителей и опекунов, которые о них заботятся.

### **1.2 Цели исследования**

Данная дипломная работа основана на следующих целях исследования:

- 1) изучить данные, необходимые для отслеживания детей по их гаджетам;
- 2) рассмотреть недостатки существующих систем слежения за людьми;
- 3) разработать беспроводную систему наблюдения по IP;
- 4) проверить эффективность разработанной системы.

### 1.3 Обзор существующих решений

Различные носимые устройства с GPS-отслеживанием, такие как умные часы, которые могут точно определить местоположение человека, используя карту на смартфоне, планшете или другом гаджете, являются частью решений, предлагаемых для отслеживания людей [2].

#### 1.3.1 Умные часы

Существующие решения по отслеживанию местоположения ребенка продолжают развиваться и с каждым новым этапом учитывают недостатки предыдущих решений. Так, исследования информационных технологий, позволяющих получать локацию ребенка в реальном времени, брали свое начало с так называемых умных часов.



Рисунок 1.1 - Часы Disney Kids

*Умные часы Disney Kids от Huawei.*

На рисунке 1.1 отображены умные часы Disney Kids от Huawei, основанные на GPS-слежении активности детей. Время автономной работы часов от аккумулятора 300 мАч составляет полтора дня, а полная зарядка занимает два часа. Однако у этого решения есть некоторые недостатки: часы стоят дорого (стоят около 100 долларов) и настроены для Disney компанией Huawei.

Безусловным преимуществом этого гаджета является удобство для ребенка. Он психологически предрасположен носить такие часы.

Самый главный недостаток такого устройства – его легко потерять или вовсе целенаправленно избавиться. Ребенок может потерять или обронить такие часы в результате высокой физической активности. А в случае, если происходит похищение ребенка человеком, который хорошо осведомлен возможными средствами защиты ребенка, то он очень легко может избавиться от часов, просто сняв их с руки ребенка.



Рисунок 1.2 - Часы Mi Bunny от Xiaomi

### *Детские умные часы Mi Bunny от Xiaomi.*

На рисунке 1.2 изображены детские умные часы Mi Bunny GPS, выпущенные Xiaomi и предназначенные для отслеживания детей, поддерживают GPS. Он также имеет кнопку SOS (включая положение и семь секунд звука) и отправляет оповещения, когда ребенок выходит за пределы пользовательской зоны безопасности. Эти часы очень подходят для использования на открытом воздухе, а не в помещениях [3].

Часы имеют хороший функционал. С их помощью можно отслеживать ребенка, но лишь в пределах определенного радиуса на карте. Кроме того, отличительной особенностью часов является возможность нажатия на кнопку SOS, что может быть полезным функционалом в чрезвычайных ситуациях.

Однако минусы предыдущего примера не учтены и в этом гаджете. Данные часы так же легко теряются или целенаправленно снимаются.

### **1.3.2 Мобильные приложения**

Дальнейшим, более продвинутым этапом реализации возможности отслеживания местонахождения ребенка стали мобильные телефоны.

Носимые устройства с GPS-отслеживанием, которые могут точно определить местоположение человека, используя карту на смартфоне или планшете, и являются частью решений, предлагаемых для отслеживания.

*Трекер GPS KidControl семейный.* Приложение работает под операционной системой Android. Оно позволяет отследить, что ребенок находится в безопасности вне зависимости от того, на каком расстоянии от родителя он находится. Один из плюсов приложения – сохраняется последнее местоположение в том случае, если соединение с интернетом было прервано. Из недостатков можно выделить, что приложение допускает погрешности в отображении точного адреса и графического отображения местонахождения ребенка на карте.

На рисунке 1.3 изображен экран запуска приложения.



Рисунок 1.3 – Экран запуска приложения

*Локатор «Где мои дети».* Это приложение так же работает под ОС Android. Оно позволяет родителю на дальнем расстоянии иметь информацию о ребенке: местоположение, звук, родительский контроль, а также может интегрироваться с другими устройствами.

Однако есть некоторые минусы, которые могут отталкивать ряд потенциальных пользователей: высокий процент потребления заряда батареи, невозможно одновременное подключение телефона и смарт-часов, необходим отдельный доступ к камере или другим функциям мобильного устройства.



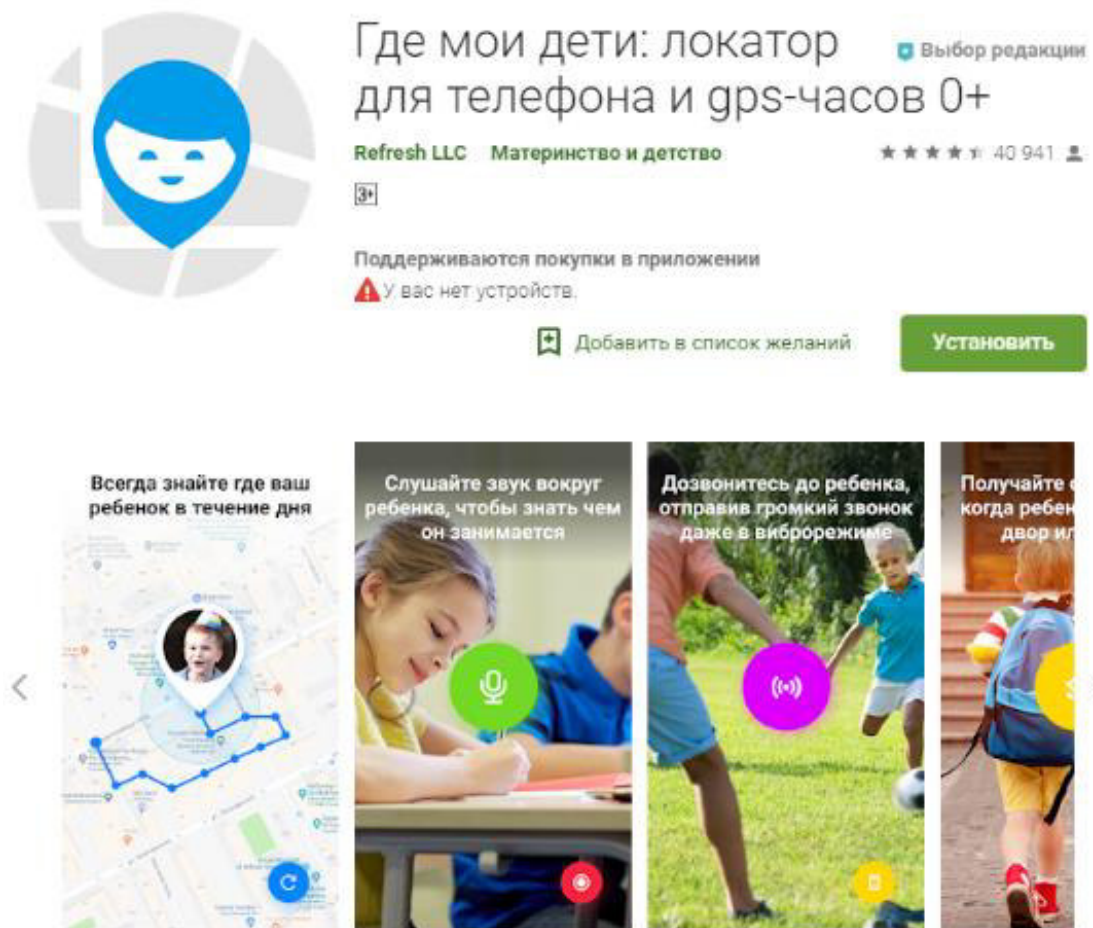


Рисунок 1.4 – Страница приложения «Где мои дети» в PlayMarket

### 1.3.3 Веб-приложения

Веб-приложения ориентированы на широкий охват пользователей. Для доступа к функционалу таких приложений, достаточно иметь устройство с подключением к сети интернет. Обычно таким устройством выступают персональные компьютеры, ноутбуки или планшеты. Среди подобных сервисов можно выделить: 2ip.ru, speed-tester.info, check-host.net, backlinks-checker.dimax.biz и другие.

Все эти сервисы объединяет схожий функционал, а следовательно, и схожие преимущества, и недостатки.

Преимуществом, безусловно считается точность определения. Так как данные приложения работают путем определения местоположения по IP.

Недостатки: отсутствие детализации и синхронизации, отсутствует интеграция с какими-либо сторонними сервисами, узкая функциональность.

### 1.4 Концептуальная основа

На рисунке 1.5 показана концептуальная основа реализации системы.

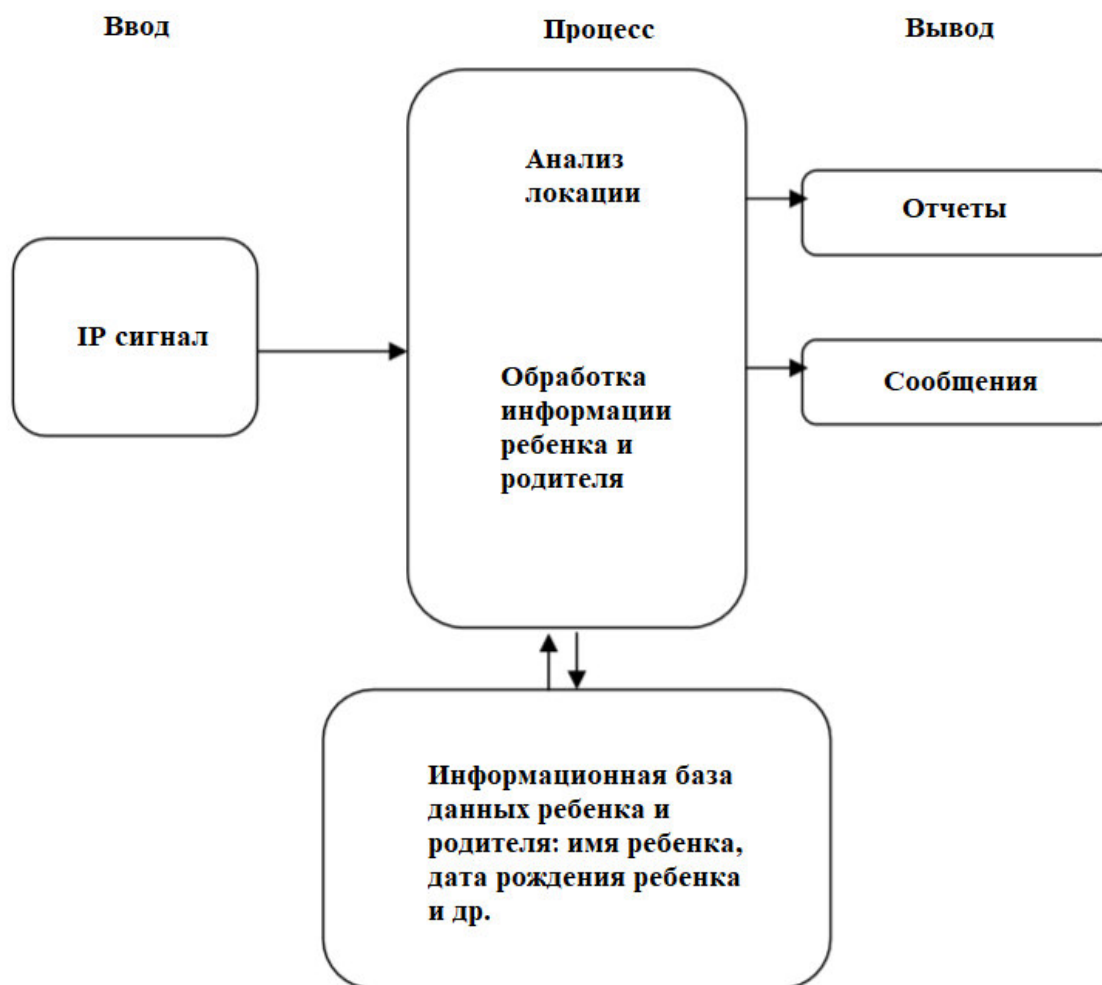


Рисунок 1.5 – Концептуальная основа

Основная информация, которая будет использоваться помимо данных о геолокации, – это местоположение GPS, имя ребенка, имя родителя или опекуна и дата рождения ребенка. Вся эта информация хранится в связанной базе данных, с которой производится регламентная синхронизация по расписанию, либо в режиме реального времени [4].

### 1.5 Постановка задач

На основании анализа существующего программного обеспечения можно выделить несколько недостатков, из которых основные:

- недостаточная интеграция со сторонними сервисами;
- неточность отображения отчетов и другой выходной информации;
- малая защищенность сохраняемой информации;
- отсутствие поддержки работоспособности гаджетов.

С целью обеспечения универсального функционального приложения были поставлены задачи, основные из которых:

- реализовать мониторинг местоположения ребенка по его смартфону;
- реализовать отображение геолокации с Яндекс.Карты
- реализовать скрытность нахождения приложения отслеживания на смартфоне ребенка;
- расширить возможные варианты использования разработанного программного обеспечения;
- протестировать процессы внедрения и эксплуатации приложения.

## **1.6 Обоснование выбора средств и технологий**

### **1.6.1 Веб-фреймворк Django**

Django — это веб-фреймворк с открытым исходным кодом на Python, используемый для быстрой разработки, прагматичного, удобного в обслуживании, чистого дизайна и защиты веб-сайтов и веб-приложений. Каркас — это набор всех компонентов, необходимых для разработки приложений. Основная цель инфраструктуры Django – позволить разработчикам сосредоточиться на компонентах приложения.

Django имеет ряд преимуществ, которые положительно влияют на выбор его как одного из основных средств разработки:

- Django проверен временем. Django – это первая платформа для реагирования на новые проблемы и уязвимости, которая уже 13 лет не теряет своей актуальности [5].
- Разработка приложений. Django был разработан командой с целью создавать веб-приложения с использованием языка программирования Python. Фреймворк имеет шаблоны, библиотеки и APIS, которые работают вместе. В целом, приложения, разработанные с использованием Django, могут быть обновлены с минимальными затратами, изменениями и дополнениями, что значительно упрощает их внедрение, эксплуатацию, а также сопровождение.
- Прост в использовании. Django использует язык программирования Python, который в настоящее время является популярным языком и используется многими. Это означает, что можно легко найти ответы на проблемы с помощью Google.
- Зависимость от операционной системы. Фреймворк Django работает на любой платформе, такой как ПК, Windows, Mac, Linux и т.д. Он обеспечивает слой между разработчиком и базой данных, называемый ORM (объектно-реляционный маппер), который позволяет перемещать или мигрировать в другие основные базы данных с помощью нескольких строк изменения кода.
- Масштабируемость и надежность. Django – это хорошо поддерживаемая платформа веб-приложений, которая широко используется в



различных отраслях промышленности. Это означает, что, приложения Django могут управляться авторизованным разработчиком с помощью одной команды в облачной среде. Поскольку разработчики Django долгое время работают в одной и той же среде разработки, они становятся все лучше и эффективнее.

– Функциональность. Фреймворк Django имеет все для создания надежной фреймворк с основными функциями:

- а) Шаблонные слои;
- б) Формы, процесс разработки;
- в) Локализация, производительность и оптимизация;
- г) Географическая структура, общие инструменты для разработки веб-приложений;
- д) Другие основные функции, необходимые для веб-сайтов.

### 1.6.2 Фреймворк Flask

Flask – это фреймворк Python, основанный на Werkzeug, Jinja2 и вдохновленная платформой Sinatra Ruby, доступной по лицензии BSD. Он был разработан в россо Армином Ронахером. Хотя Flask довольно молодой по сравнению с большинством фреймворков Python, он имеет большие перспективы и уже завоевал популярность среди веб-разработчиков Python. Flask, так называемый «микро» фреймворк для Python имеет ряд уникальных преимуществ и удобного набора функций и возможностей.

Flask был разработан чтобы быть простым в использовании и сопровождении программного обеспечения. Идея Flask заключается в создании прочной основы для веб-приложений различной сложности. С этого момента открывается возможность подключать любые расширения, которые необходимы для решения поставленных задач. Есть возможность создавать свои собственные модули. Flask отлично подходит для любых видов проектов. Его отличительное преимущество – он чрезвычайно хорош при прототипировании. Flask зависит от двух внешних библиотек: шаблонизатора Jinja2 и инструментария Werkzeug WSGI.

Flask впечатляет своей функциональностью:

- встроенный сервер разработки и быстрый отладчик;
- интегрированная поддержка модульного тестирования;
- отправка RESTful-запроса;
- Jinja2 шаблонизатор;
- поддержка безопасных файлов cookie (сеансы на стороне клиента);
- WSGI 1.0 соответствует;
- Построен на основе юникода.

Подводя итог, можно сказать, что Flask – это одна из самых полных и многофункциональных доступных микросистем. Еще Flask имеет первоклассные расширения и элегантный API. Flask обладает всеми преимуществами быстрых шаблонов, мощных функций WSGI, тщательной

тестируемости модулей на уровне веб-приложений и библиотек, обширной документации.

### 1.6.3 Язык программирования Python

Python является одним из 10 самых популярных языков программирования. Python – это язык программирования общего назначения и высокого уровня. Его можно использовать для разработки настольных GUI-приложений, веб-сайтов и веб-приложений. Преимущества Python:

- Читабельный и обслуживаемый код. При написании программного приложения обязательно надо сосредоточиться на качестве его исходного кода, чтобы упростить обслуживание и обновления. Синтаксические правила Python позволяют выразить понятия без написания дополнительного кода. В то же время Python, в отличие от других языков программирования, делает упор на читабельность кода и позволяет использовать английские ключевые слова вместо знаков препинания.

- Несколько парадигм программирования. Python поддерживает несколько парадигм программирования. Он поддерживает объектно-ориентированное и структурированное программирование. Таким образом, его языковые возможности поддерживают различные концепции в функциональном и аспектно-ориентированном программировании. В то же время, Python имеет динамическую систему типов и автоматическое управление памятью. Парадигмы программирования и языковые функции помогают использовать Python для разработки больших и сложных программных приложений.

- Совместим с основными платформами и системами. В настоящее время Python поддерживает множество операционных систем. Даже можно использовать интерпретаторы Python для запуска кода на определенных платформах и инструментах, что позволяет запускать один и тот же код на нескольких платформах без перекомпиляции. Полностью отсутствует необходимость перекомпилировать код после внесения каких-либо изменений. Можно запустить измененный код приложения без повторной компиляции и немедленной проверки влияния изменений, внесенных в код. Эта функция действительно облегчает внесение изменений в код без увеличения времени разработки [6].

- Надежная стандартная библиотека. Его большая и надежная стандартная библиотека позволяет Python превосходить другие языки программирования. Стандартная библиотека позволяет выбирать из широкого спектра модулей в соответствии с потребностями.

- Тест-ориентированная разработка. Есть возможность использовать Python для быстрого создания прототипа программного приложения. Таким образом, представляется возможным собрать программное обеспечение непосредственно из прототипа просто путем рефакторинга кода Python.

## 1.6.4 Atom для Python

Основным способом создания и запуска программы на Python является создание пустого файла с расширением .py и указание на этот файл из командной строки с помощью `python filename.py`. Кроме того, можно использовать Python IDLE (среда разработки) по умолчанию, который устанавливается вместе с Python. Писать и выполнять код можно сразу из среды разработки IDLE. Однако эти варианты можно считать устаревшими, так как они не отвечают современным требованиям продуктивности.

В таком случае, лучшим выходом из ситуации является редактор Atom.

Атом был построен GitHub со слоганом «Взломанный текстовый редактор для 21-го века». И он действительно гибкий и имеет отличную поддержку внешних пакетов, что делает его мощной интерактивной средой разработки (IDE). IDE-python требует Atom 1.21+, языковой сервер Python 0.19+ и пакет `atom-ide-ui` для предоставления функциональности в Atom.

После установки atom можно запустить, введя `atom` в командной строке. Для этого необходимо убедиться, что атом был добавлен в переменные пути / среды поиска [7].

Хотя более удобный способ открыть Atom может состоять в том, чтобы щелкнуть правой кнопкой мыши папку, в которой находятся исполняющие файлы, а затем перейти к «Открыть с помощью Atom». Это добавит все файлы этой папки в древовидную структуру, как видно на рисунке 1.6. Это замечательно, когда разрабатываемая программа или веб-приложение состоит из нескольких файлов. Можно просто переходить от файла к файлу и редактировать их изнутри Atom.

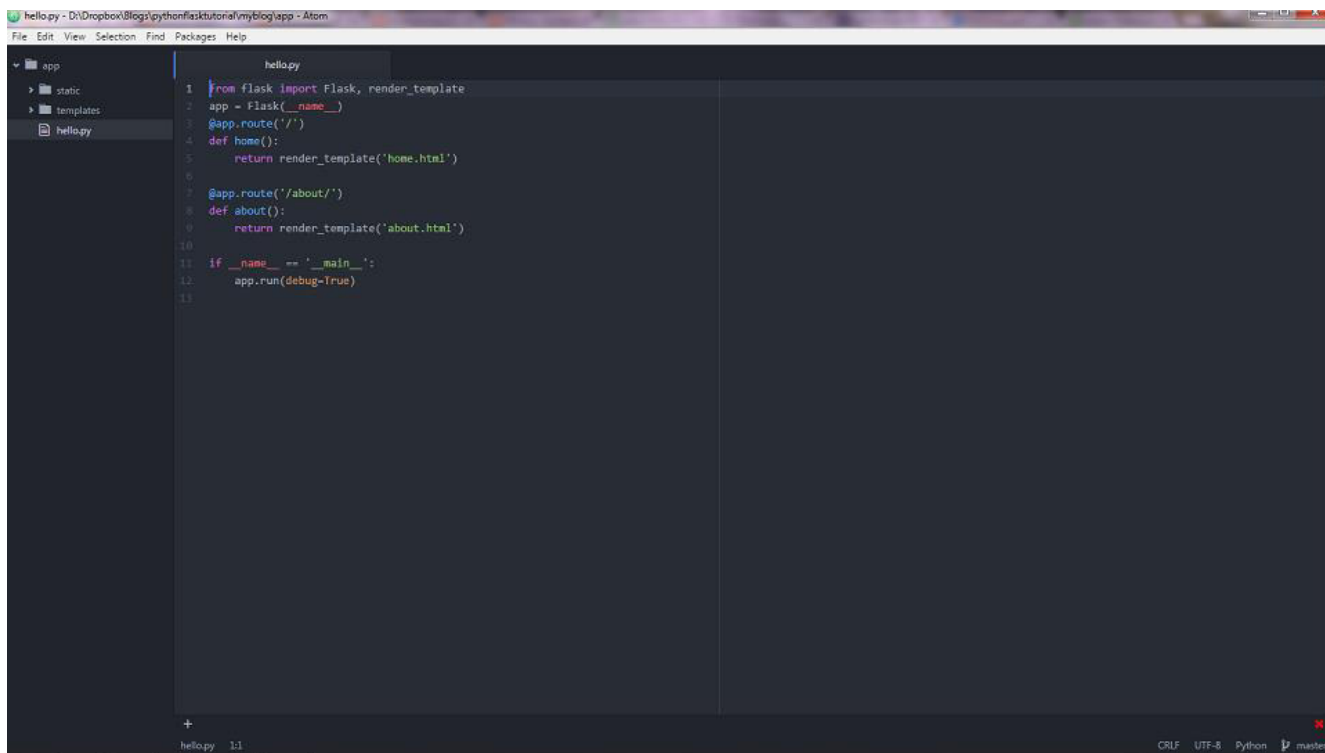


Рисунок 1.6 – Интерфейс Atom

### **1.6.5 База данных MySQL**

Наиболее полный набор расширенных функций, инструментов управления и технической поддержки для достижения высочайшего уровня масштабируемости, безопасности, надежности и бесперебойной работы MySQL.

Django поддерживает MySQL 5.6 и выше. Функция `inspectdb` в Django использует базу данных `information_schema`, которая содержит подробные данные обо всех схемах базы данных. Django ожидает, что база данных будет поддерживать Unicode (кодировка UTF-8) и делегирует ей задачу обеспечения транзакций и ссылочной целостности [8].

MySQL очень прост в установке, и благодаря множеству сторонних инструментов, которые могут быть добавлены в базу данных, настройка реализации является относительно простой задачей. Кроме того, это также простая база данных для работы. При условии, что разработчик знаком с языком, возникновение различного рода проблем минимизировано.

Природа MySQL, которая начиналась как платформа с открытым исходным кодом, означает, что существует большое и процветающее сообщество разработчиков и энтузиастов, к которому можно обратиться за помощью. Во многом это связано с популярностью решения, конечным результатом которого является нехватка специалистов.

В зависимости от того, для чего планируется его использовать, реализация MySQL может варьироваться в цене от бесплатной до 10000 долларов и более. В любом случае, это значительно дешевле, чем большинство других вариантов баз данных на рынке (за исключением конкурентов MySQL с открытым исходным кодом).

Хотя популярность MySQL в последние годы несколько снизилась, он остается одной из наиболее часто используемых систем баз данных в мире. Он совместим практически с любой операционной системой и является более или менее отраслевым стандартом. Именно эти причины делают возможным легко вносить доработки в проект, а также сопровождать его продолжительное время.

### **1.6.6 API Яндекс.Карт**

API Яндекс.Карт – это объединение функциональных возможностей, которые можно удобно и эффективно внедрить в проектах, где необходимы картографические данные и технологии Яндекса [9].

API Яндекс.Карт можно разделить на 3 функциональные составляющие:

- адреса и организации;
- карты;
- сервисы для решения логистических задач.

## 2 Специальная часть

### 2.1 Проектирование приложения средствами UML

UML (от сокр. Unified Modeling Language), является стандартизированным языком моделирования, состоящим из интегрированного набора диаграмм, разработанных, чтобы помочь разработчикам систем и программного обеспечения в определении, визуализации, конструировании и документировании аспектов программных систем. UML является очень важной частью разработки программного обеспечения и процесса разработки программного обеспечения. UML использует в основном графические обозначения для выражения дизайна программных проектов.

На рисунке 2.1 изображена диаграмма UML – Передача информации.

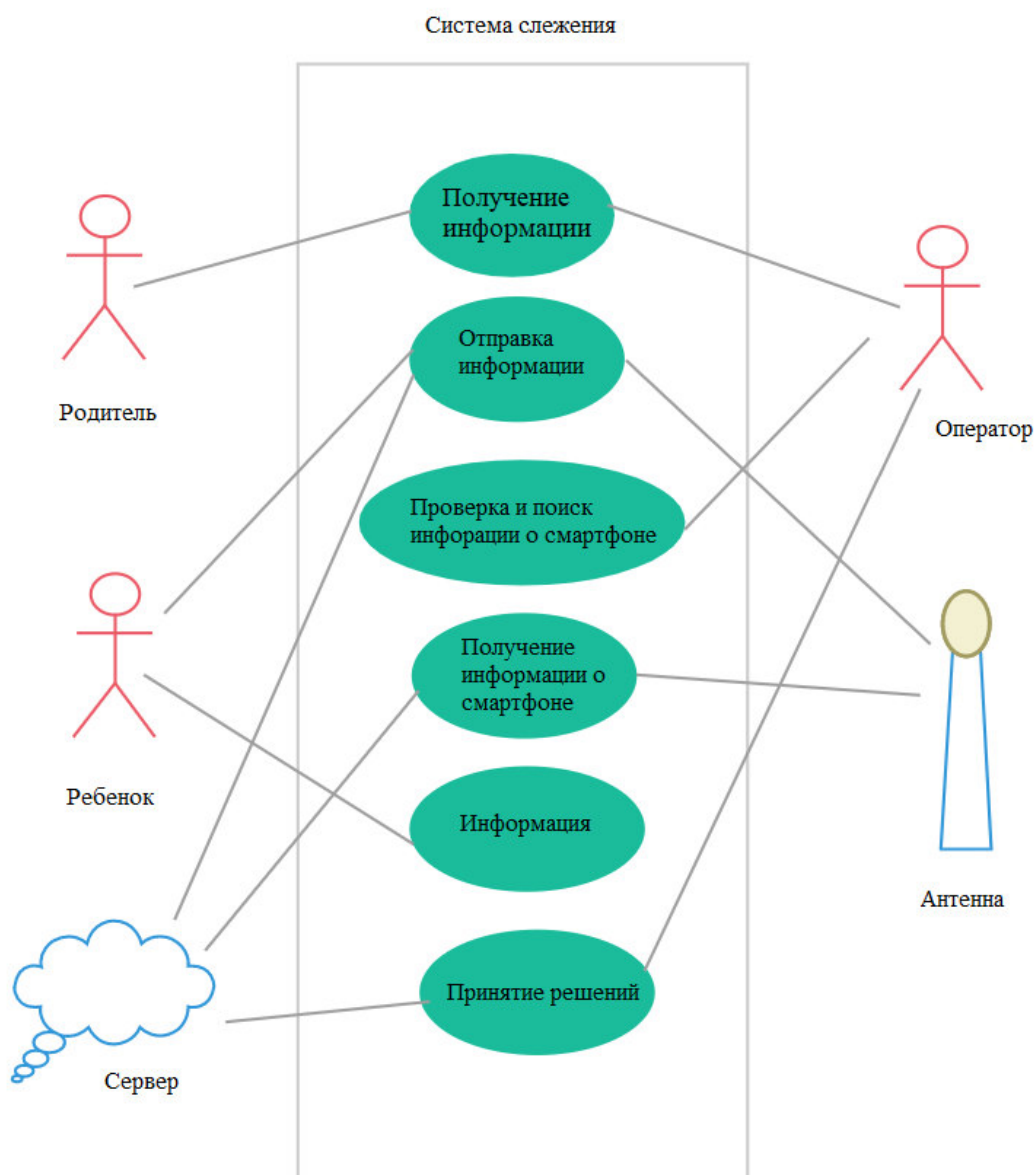


Рисунок 2.1 - Диаграмма UML – Передача информации

На данной диаграмме представлен процесс передачи информации между внутренними компонентами системы. На диаграмме видно, что в системе взаимодействуют три пользователя: администратор, пользователь – ребенок и пользователь – родитель.

Со смартфона ребенка на сервер поступает определенная информация. Передача происходит при наличии доступа в интернет, что проверяется наличием соединения с антенной. При успешной отправке данных, они попадают на сервер, после чего через фильтрацию данных, которой занимается администратор (в основном на основании автоматизированных процессов) попадает на выход к пользователю – родителю, который в свою очередь сохраняет эту информацию у себя.

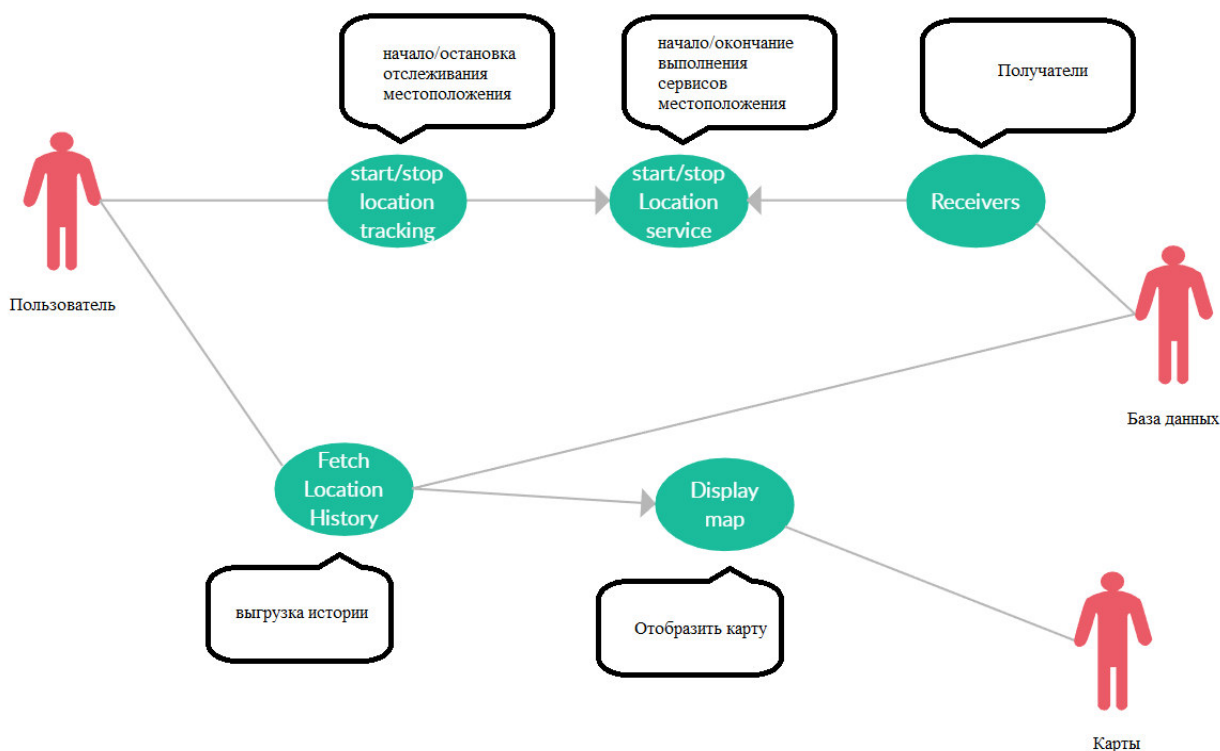


Рисунок 2.2 - Диаграмма UML – Местоположение

На рисунке 2.2 представлена диаграмма UML – Местоположение.

Данная диаграмма отображает логику получения местонахождения устройства и отображения на карте. Пользователь своей регистрацией на сайте запускает процесс отслеживания местоположения. С этим процессом взаимодействуют другие процессы, такие как запуск сервисов местонахождения и функции получения информации.

Результатом данного процесса является карта, которая показывает точное месторасположение устройства.

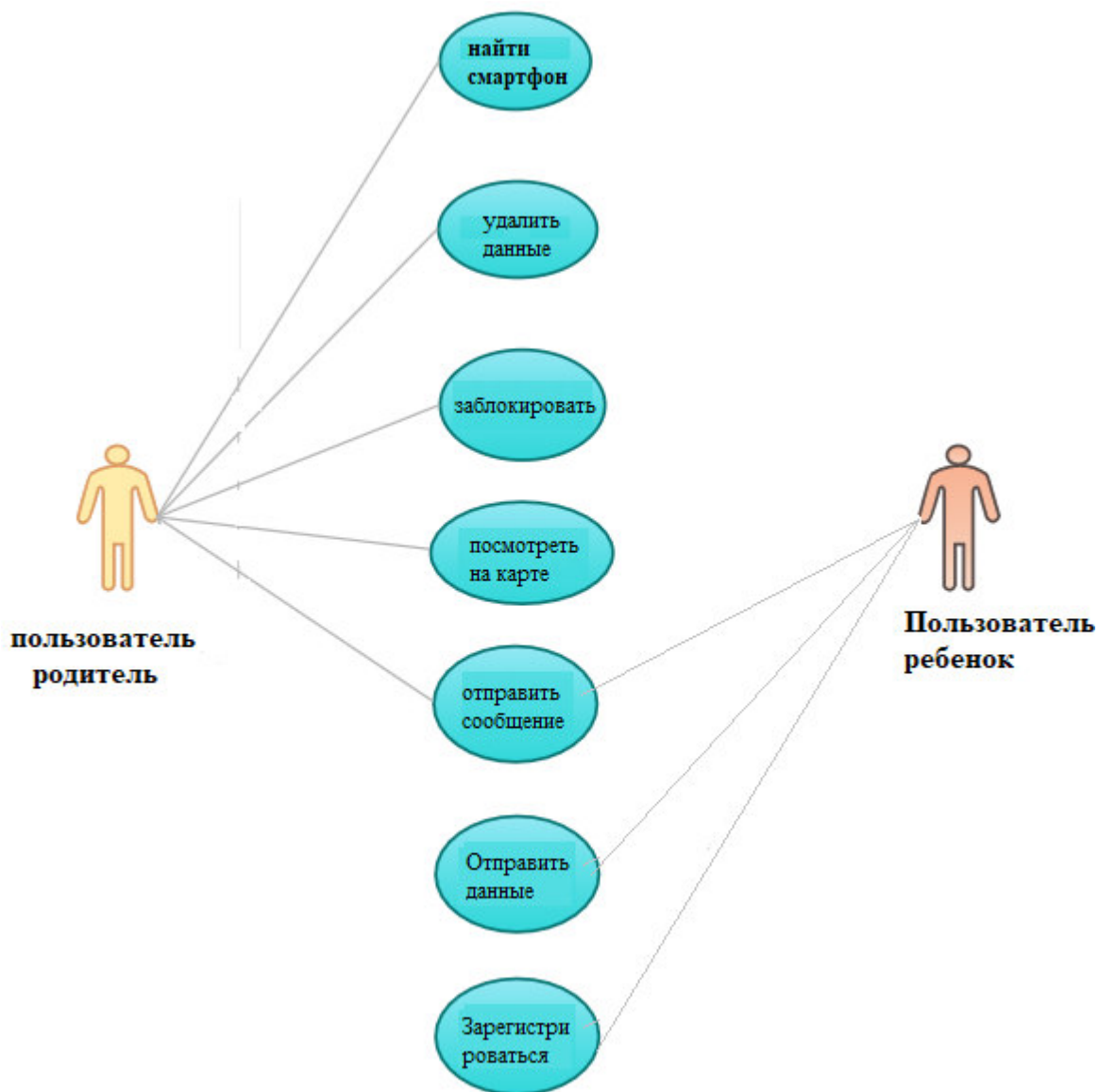


Рисунок 2.3 - Диаграмма UML – Пользователи

На рисунке 2.3 показана диаграмма UML – Пользователи.

На данной диаграмме показано разделение функциональных возможностей двух пользователей. Видно, что пользователь – родитель может: находить устройства, удалять с него данные, блокировать его, смотреть местонахождение на карте, отправлять сообщение. А пользователь – ребенок может: также отправлять сообщения, отправлять данные, проходить регистрацию.

## 2.2 Использование Atom в качестве редактора Python

Атом был построен GitHub со слоганом «Взломанный текстовый редактор для 21-го века». И он действительно гибкий и имеет отличную поддержку внешних пакетов, что делает его мощной интерактивной средой разработки (IDE).

Для того чтобы начать работу с Atom (см. рисунок 2.4) необходимо проделать несколько не сложных шагов.

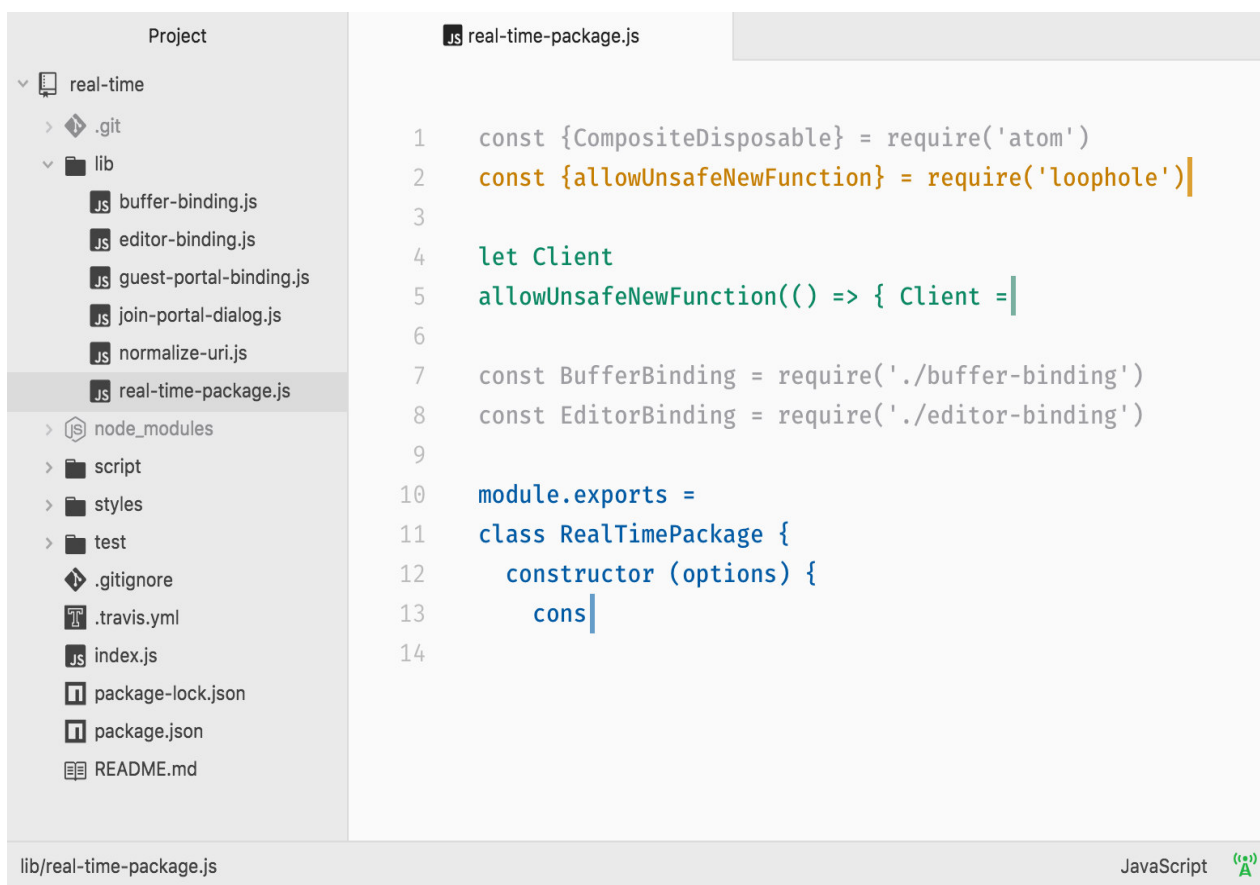


Рисунок 2.4 – Интерфейс редактора Atom

### 2.2.1 Установка Atom

Установка не является слишком трудозатратной.

На официальном сайте сразу появляется кнопка для скачивания соответствующих архивов. (см. рисунок 2.5)



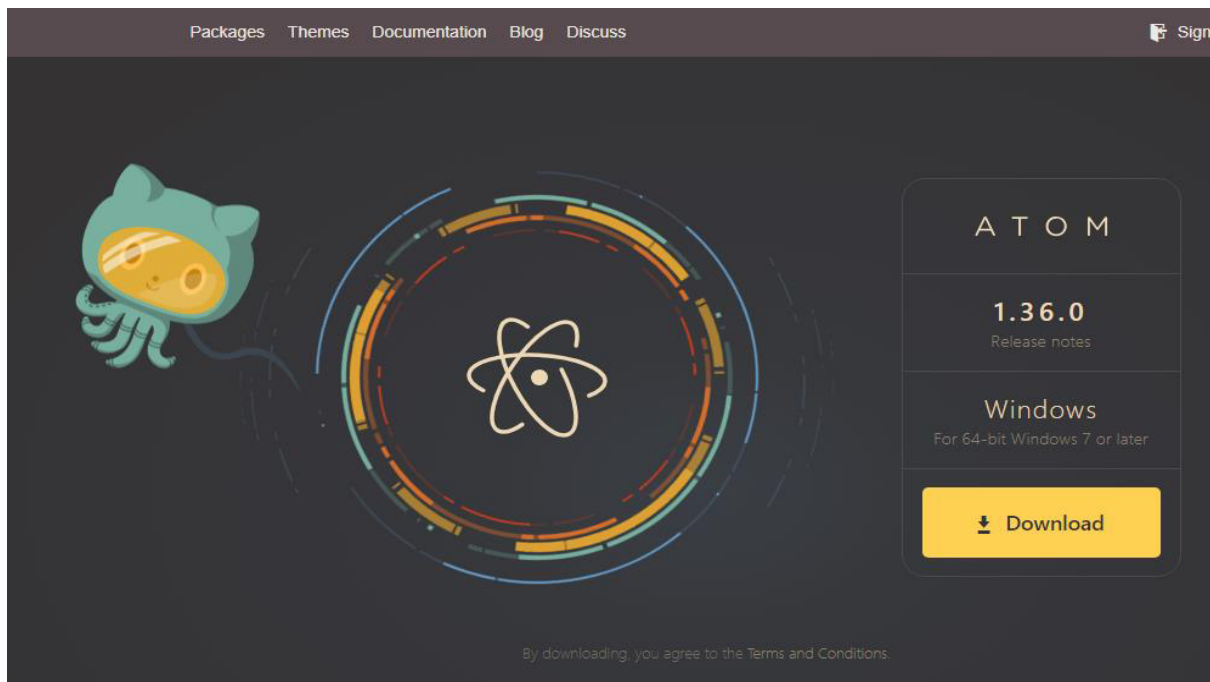


Рисунок 2.5 – Скачивание Atom

Кнопка обычно корректно соответствует платформе, а загружаемый пакет легко устанавливается.

Atom доступен с установщиками Windows, которые можно загрузить с сайта [atom.io](https://atom.io) или со страницы релизов Atom. В нашем случае система является 64-разрядной. Эта программа установки установит Atom, добавит команды `atom` и `arm` в PATH и создаст ярлыки на рабочем столе и в меню «Пуск». Эти системные действия выполняются автоматически после запуска файла `AtomSetup-x64.exe` [10].

Для повышения продуктивности необходимо изменить системные настройки. Контекстное меню «Открыть с помощью Atom» в проводнике и возможность сделать Atom доступным для сопоставления файлов с помощью «Открыть с помощью...» управляется панелью «Системные настройки», как показано на рисунке 2.6.

Открыв Atom, выбираем Файл - Настройки, а затем вкладку Система слева. Устанавливаем флажки рядом с «Показать в контекстных меню файлов», а также «Показать в контекстных меню папок». На этом установка основных средств Atom считается завершенной.

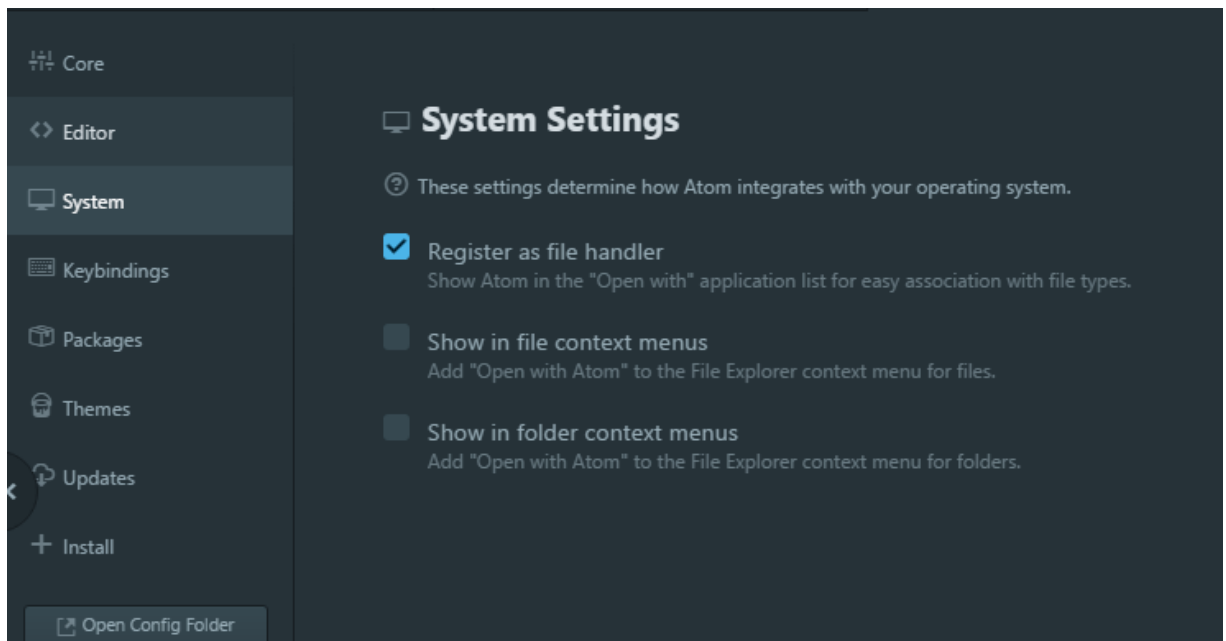


Рисунок 2.6 – Системные настройки

### 2.2.2 Обновление Atom

Для корректного сопровождения программного продукта есть необходимость обновлять Atom для последних улучшений программного обеспечения. «Автоматическое обновление» включено по умолчанию в основных настройках представления настроек, что позволит Atom автоматически проверять наличие обновлений.

### 2.2.3 Настройки прокси и брандмауэра

Брандмауэр выдает ошибки SSL при установке пакетов, так как пытается оградить пользователя от нелегальных и неизвестных архивов и компонентов (см. рисунок 2.7 и 2.8).

Необходимо отключить строгий SSL, выполнив:

```
C:\> apm config set strict-ssl false
```

Рисунок 2.7– Отключение жестких сертификатов

```
C:\> apm config set https-proxy YOUR_PROXY_ADDRESS
```

Рисунок 2.8 – Настройка apm для его использования прокси

### 2.2.4 Пакеты Atom

Чтобы установить новый пакет, используем вкладку «Установка» в привычном представлении «Настройки». Открываем окно настроек с помощью Ctrl +, нажимаем на вкладку «Установить» и вводим свой поисковый запрос в поле «Установить пакеты» (см. рисунок 2.9) [11].

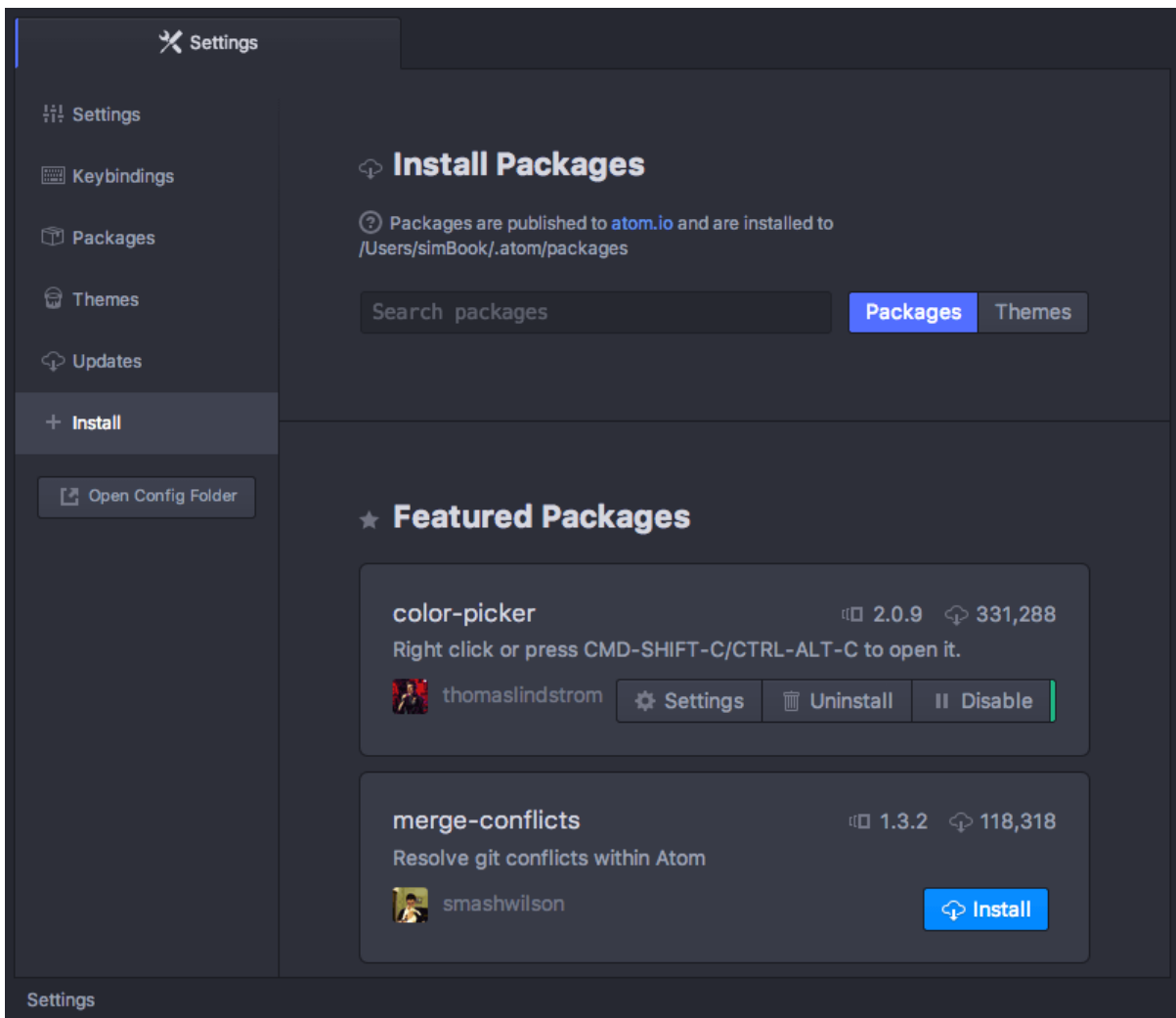


Рисунок 2.9 – Установка пакетов

## 2.2.5 Выполнение файла Python в Atom

Открываем командную строку и указываем пути к файлам или используем отличный пакет Atom, который называется `platform-ide-Terminal`. который интегрируется с Atom, так что можно выполнять файлы изнутри Atom, что очень удобно и положительно влияет на производительность разработки.

Чтобы установить пакет, нужно перейти в «Файл» -> «Настройки», а затем «Установить» и найти имя пакета (в нашем случае `platformio-ide-Terminal`). После установки пакета `Terminal-Plus` можно найти инструмент и открыть экземпляр терминала, перейдя в Пакеты. Более быстрый способ открыть терминал - щелкнуть на значке плюса, который был добавлен в нижней части окна Atom. Это должно открыть терминал.

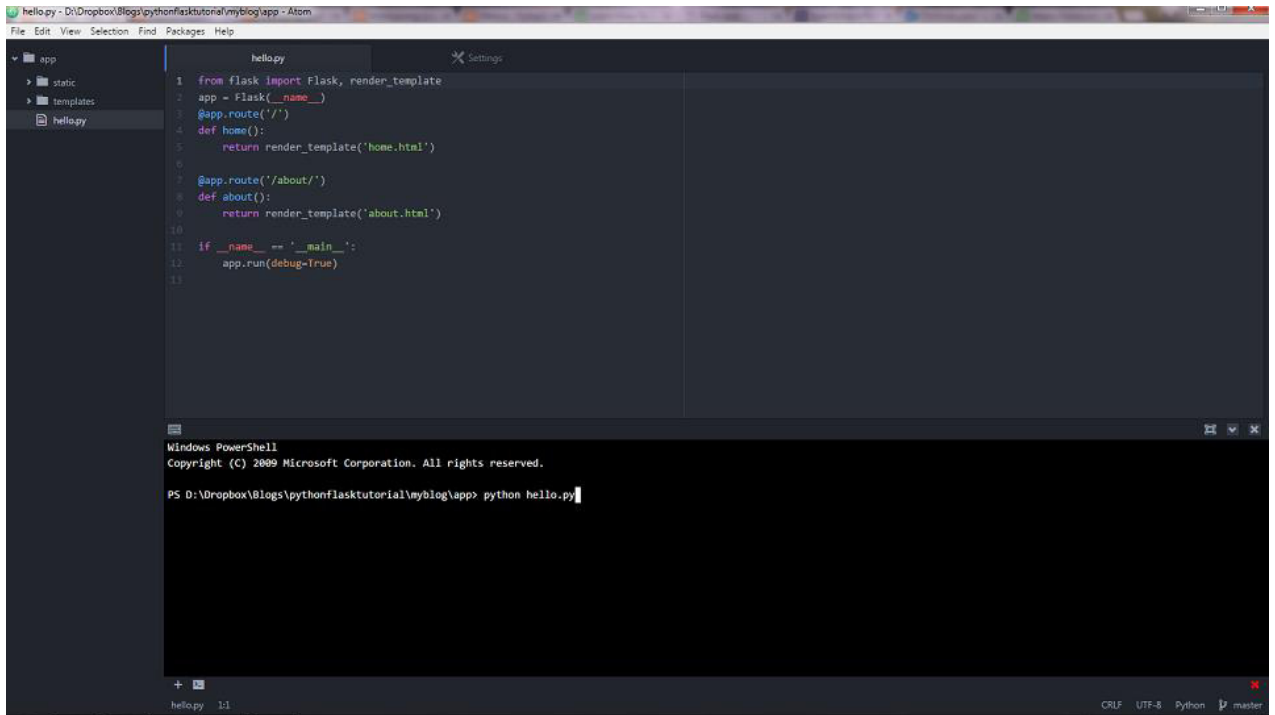


Рисунок 2.10 – Терминал окна Atom

Видно, что терминал указывает на главный каталог файлов. Теперь можно выполнять скрипты Python оттуда, как показано на рисунке 2.10.

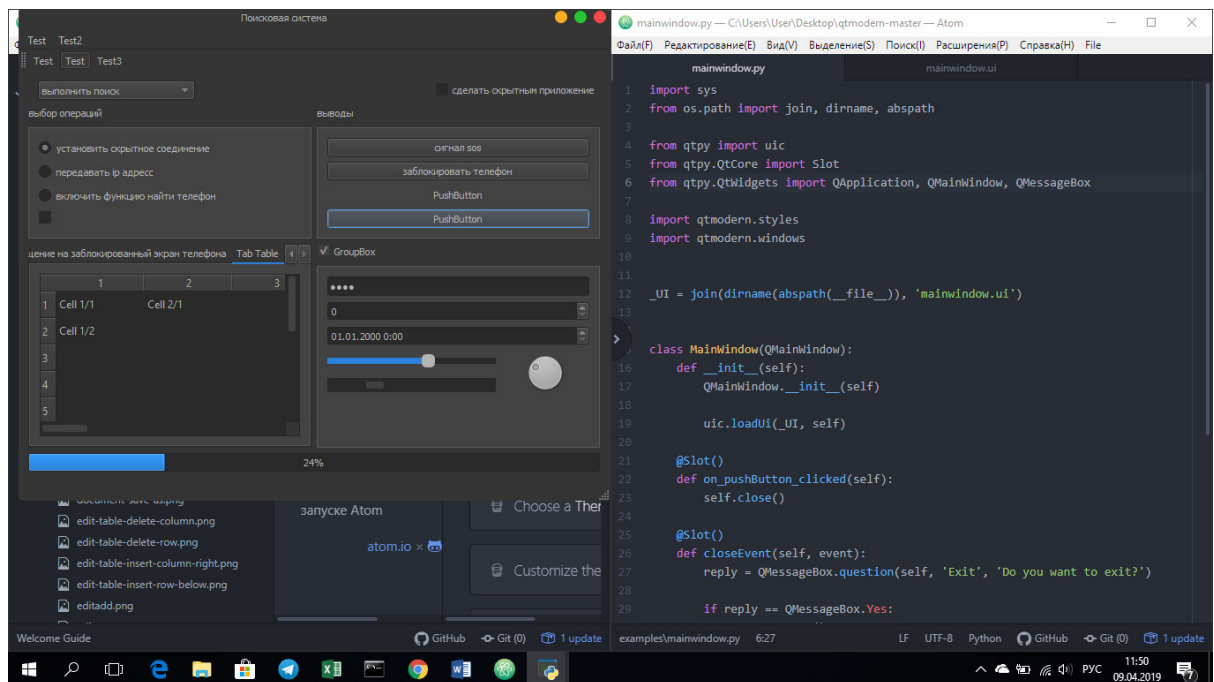


Рисунок 2.11 – Создание интерфейса и подключение к классам

Библиотеки Python. PyQt5 Qdesigner. Создаем интерфейс и подключаем к классам.

В ходе проведения работ по связке моделей возникла ошибка, которая указана на рисунке 2.12

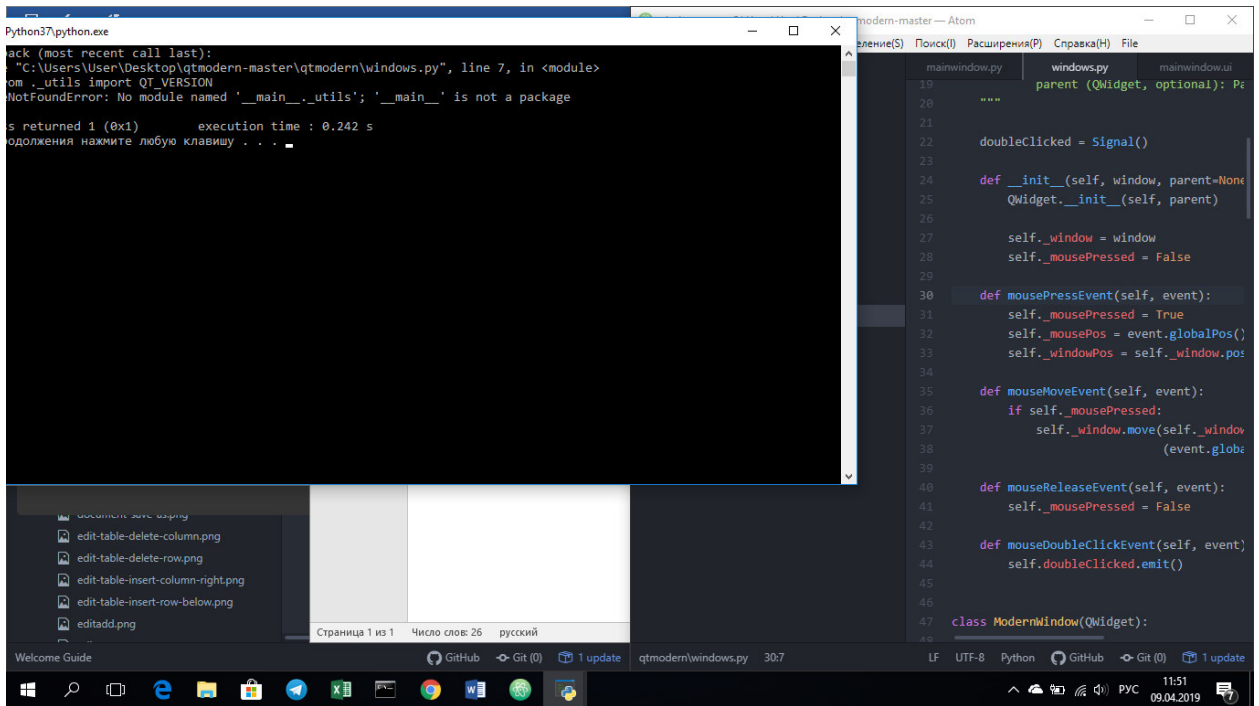


Рисунок 2.12 – Ошибка привязки

Причина возникновения подобной ошибки обуславливается тем, что путь к script не был прописан.

Ситуация налаживается после того, как путь к script с которого он запускается прописан (см. рисунок 2.13).

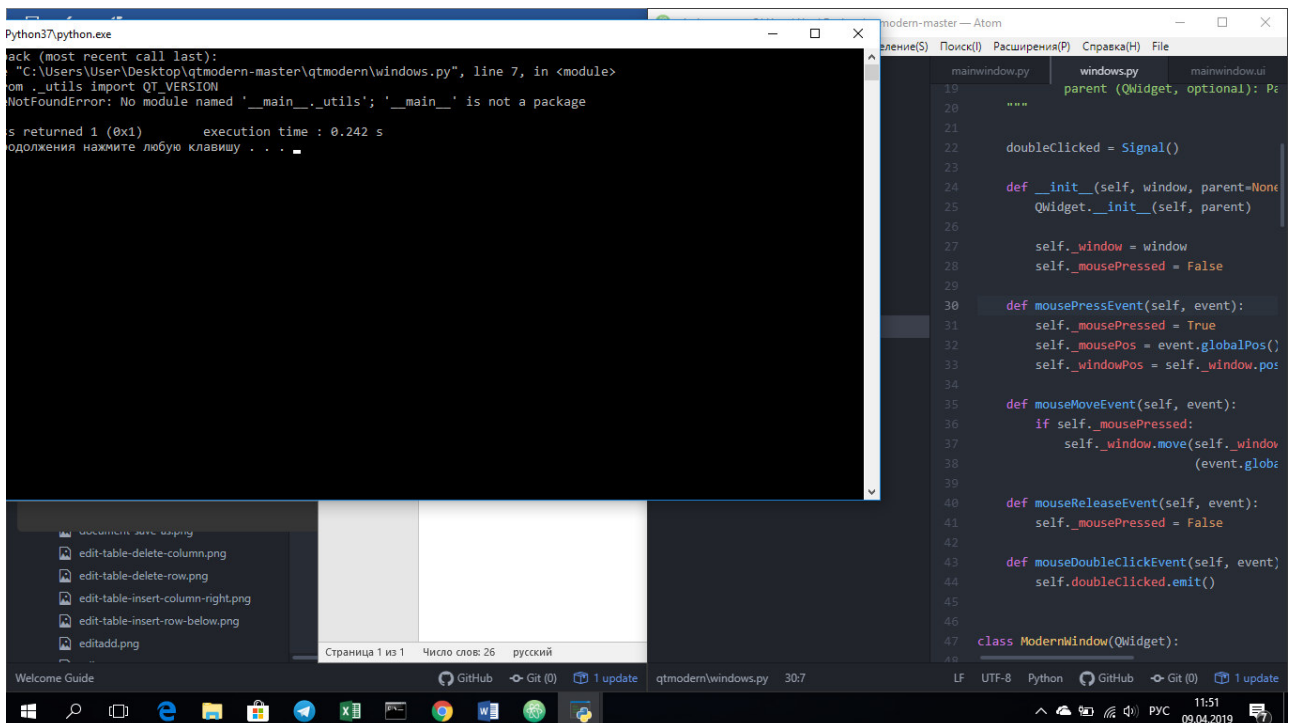


Рисунок 2.13 – Запуск приложения без ошибок

## 2.3 Управление пакетами

С помощью программы `pip` можно устанавливать, обновлять и удалять пакеты. По умолчанию `pip` [13] установит пакеты из индекса пакетов Python, <<https://pypi.org>>. Также можно просмотреть индекс пакетов Python, перейдя к нему в веб-браузере, или использовать ограниченную функцию поиска `pip`:

```
(tutorial-env) $ pip install -r requirements.txt
Collecting novas==3.1.1.3 (from -r requirements.txt (line 1))
| ...
Collecting numpy==1.9.2 (from -r requirements.txt (line 2))
| ...
Collecting requests==2.7.0 (from -r requirements.txt (line 3))
| ...
Installing collected packages: novas, numpy, requests
| Running setup.py install for novas
Successfully installed novas-3.1.1.3 numpy-1.9.2 requests-2.7.0
```

Рисунок 2.14 – Установка необходимых пакетов

Можно установить все необходимые пакеты с помощью команды `install -r`, как показано на рисунке 2.14.

## 2.4 Запуск программы без привязки к виртуальному окружению

Приложения Python часто используют пакеты и модули, которые не входят в стандартную библиотеку. Приложениям иногда требуется определенная версия библиотеки, потому что приложение может требовать исправления определенной ошибки, или приложение может быть написано с использованием устаревшей версии интерфейса библиотеки.

Решением этой проблемы я нашел создание виртуальной среды, отдельного дерева каталогов, которое содержит установку Python для конкретной версии Python, а также ряд дополнительных пакетов [12].

Различные приложения могут использовать разные виртуальные среды. Чтобы разрешить предыдущий пример противоречивых требований, первое приложение может иметь собственную виртуальную среду с установленной версией 1.0, а второе приложение имеет другую виртуальную среду с версией 2.0. Если второе приложение требует обновления библиотеки до версии 3.0, это не повлияет на среду первого приложения. Таким образом решается проблема привязки к виртуальному окружению (см. рисунок 2.15).



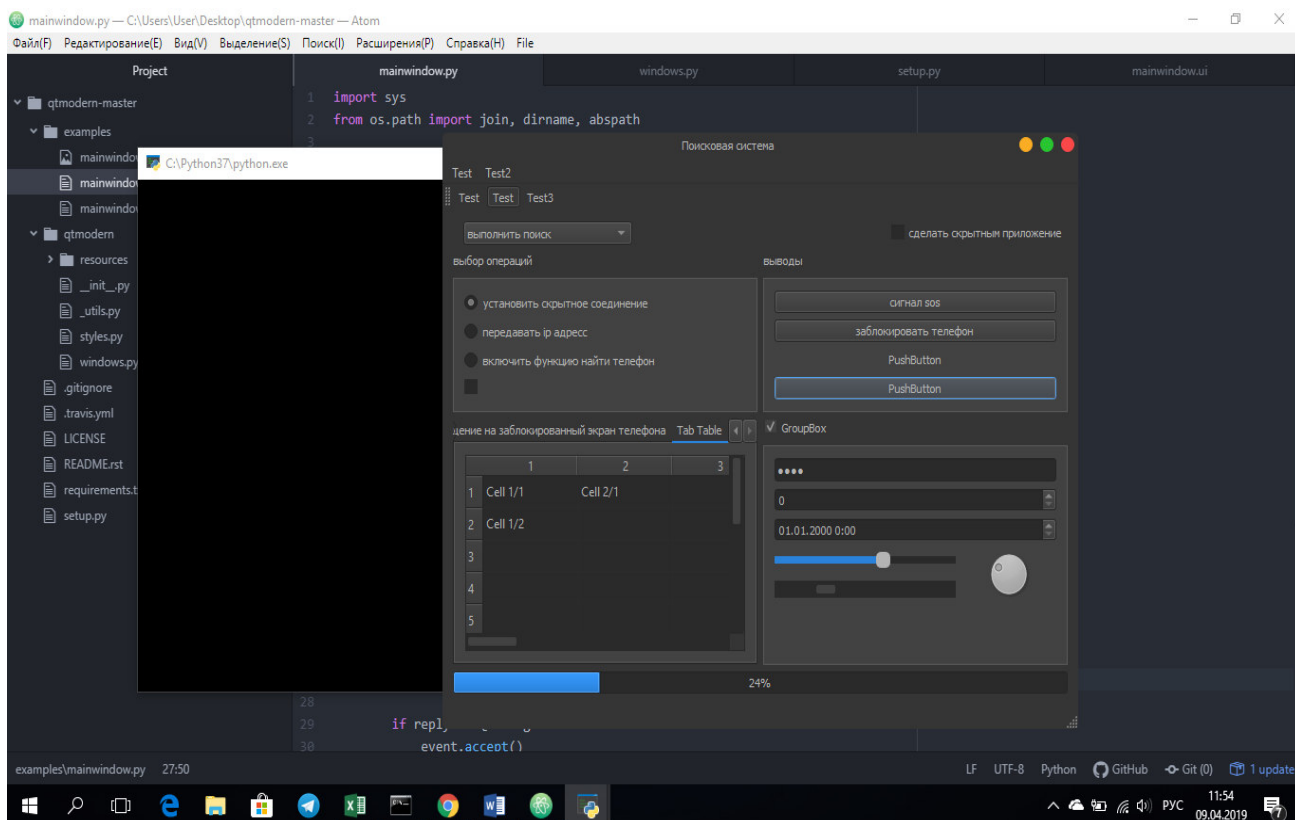


Рисунок 2.15 – Запуск программы без привязки к виртуальному окружению

## 2.5 Создание виртуальной среды

Модуль, используемый для создания и управления виртуальными средами, называется `venv`. Обычно `venv` устанавливает самую последнюю версию Python, которая у есть в системе. Если в системе имеется несколько версий Python, то предоставляется возможным выбрать конкретную версию Python, запустив `python3` или любую другую версию, какую необходимо.

Чтобы создать виртуальную среду, нужно выбрать каталог, в который он в дальнейшем будет помещен, и запустить модуль `venv` в виде сценария с путем к каталогу как на рисунке 2.16.

```
python3 -m venv tutorial-env
```

Рисунок 2.16 – Создание виртуальной среды

Это создаст каталог `tutorial-env`, если он не существует, а также создаст внутри него каталоги, содержащие копию интерпретатора Python, стандартную библиотеку и различные вспомогательные файлы.

После создания среды, необходимо ее активировать как на рисунке 2.17.

```
tutorial-env\Scripts\activate.bat
```

Рисунок 2.17 – Активация виртуальной среды

Код активации виртуальной среды на рисунке 3.13 соответствует ситуации в данном дипломном проекте, так как используется операционная система Windows.

Активация виртуальной среды изменит оболочку так, чтобы показать, какая виртуальная среда используется в данный момент, и изменит эту среду так, чтобы при запуске python выводилась именно эта версия и установка Python (см. рисунки 2.18 и 2.19).

```
$ source ~/envs/tutorial-env/bin/activate
(tutorial-env) $ python
Python 3.5.1 (default, May  6 2016, 10:59:36)
| ...
>>> import sys
>>> sys.path
['', '/usr/local/lib/python35.zip', ...,
 '~/.envs/tutorial-env/lib/python3.5/site-packages']
>>>
```

Рисунок 2.18 – Выбор запуска среды разработки по умолчанию



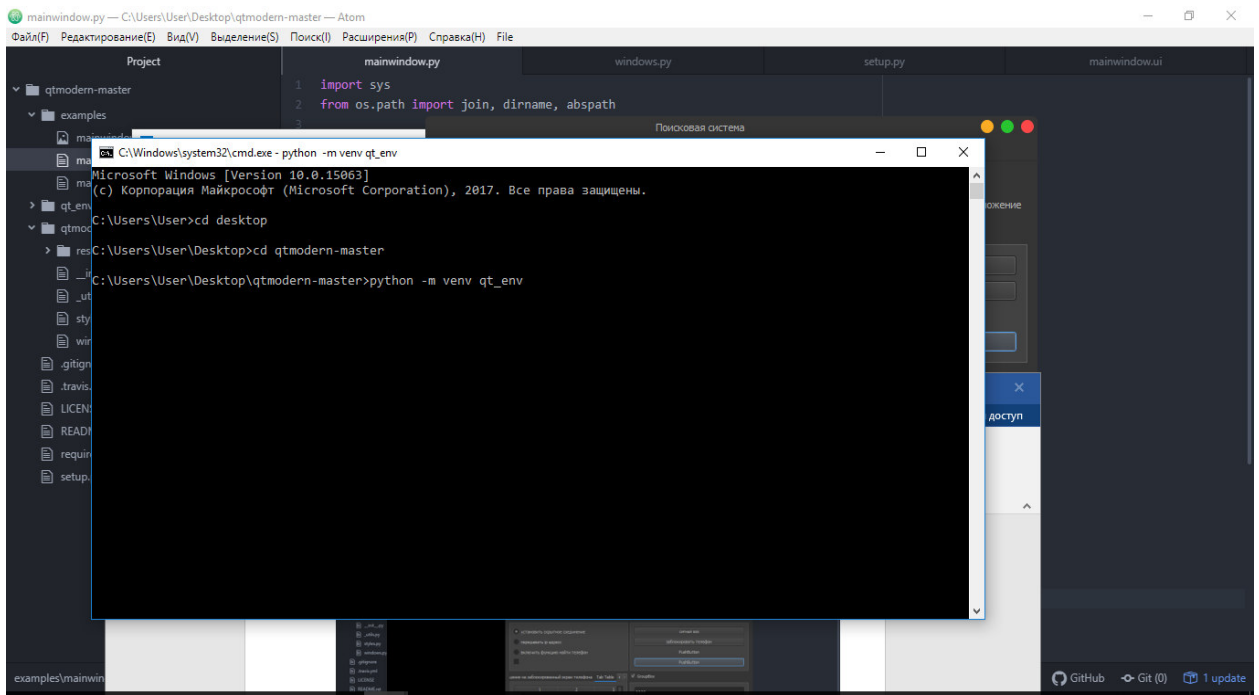


Рисунок 2.19 - Создание виртуальной среды

## 2.6 Привязка к базе данных SQL 3

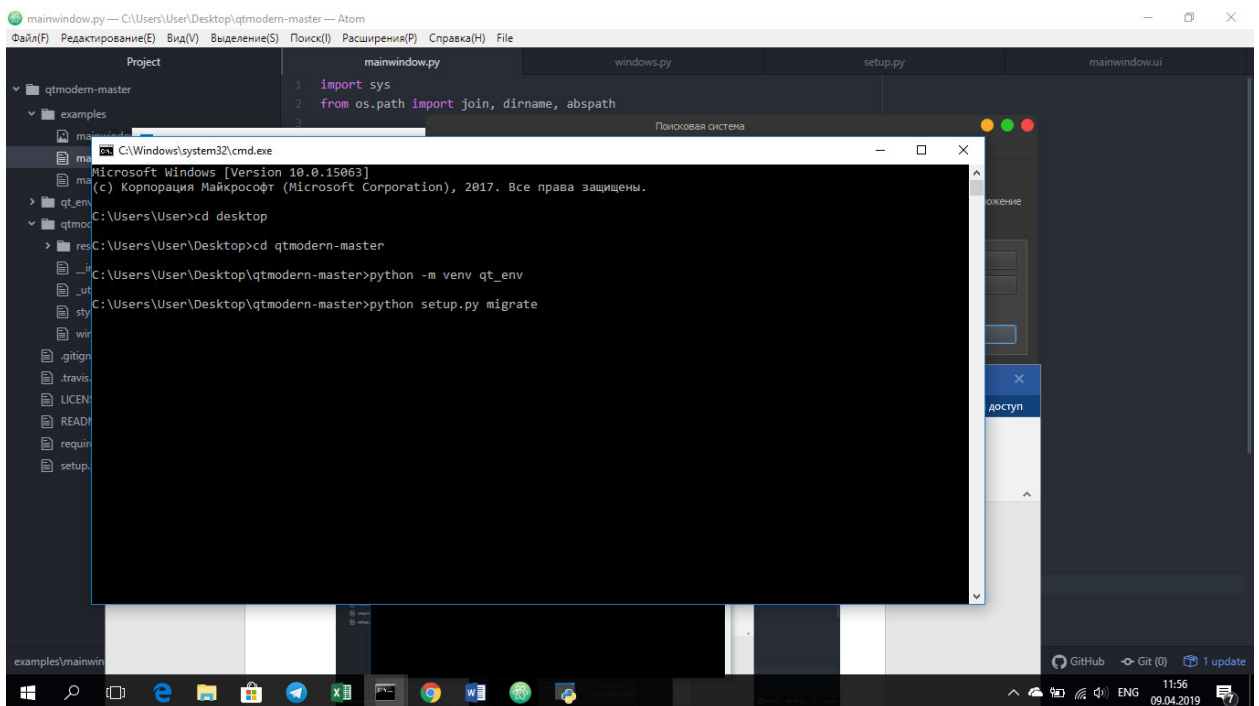


Рисунок 2.20 – Процесс привязки к базе данных SQL

На рисунке 2.20 показан скрипт, который позволяет привязать приложение к базе данных на SQL.

Установка соединения показана на рисунке 2.21.

```

import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="yourusername",
    passwd="yourpassword"
)

print(mydb)

```

Рисунок 2.21- Создание соединения

## 2.7 Автоматизация геолокации

Автоматизации входа в Citrix

Переменная `self` представляет экземпляр самого объекта. Большинство объектно-ориентированных языков передают это как скрытый параметр методам, определенным для объекта; Python нет. В питоне его надо объявлять явно. Экземпляр класса `A` и его методы, будут переданы автоматически [14].

Метод «`__init__`» — это примерно то, что представляет конструктор в Python. Когда вызывается метод, Python создает объект и передает его в качестве первого параметра методу «`__init__`». Любые дополнительные параметры также будут переданы в качестве аргументов - в этом случае вызывается исключение, так как конструктор их не ожидает (см. рисунок 2.22).

```

def __init__(self, url):
    """
    :param url:
    """
    self.url = url
    try:
        self.chrome_option = webdriver.ChromeOptions()
    except BaseException as error:
        print(" Драйвер Chrome не в пути или селен не установлен, или обт ошибки : ", error)
    self.latitude = None
    self.longitude = None

```

Рисунок 2.22 Автоматизации входа в Citrix

Автоматизация процесса показана на рисунке 2.23.

```

def run_automation(self):
    """
    Это на самом деле работает и автоматизирует весь процесс
    """
    try:
        browser = webdriver.Chrome(chrome_options=self.chrome_option)
        browser.get(self.url)
        self.latitude = browser.find_element_by_id("latitude").get_attribute("value")
        self.longitude = browser.find_element_by_id("longitude").get_attribute("value")
        while True:
            if self.latitude != browser.find_element_by_id("latitude").get_attribute(
                "value") and self.longitude != browser.find_element_by_id("longitude").get_attribute("value"):
                self.latitude = browser.find_element_by_id("latitude").get_attribute("value")
                self.longitude = browser.find_element_by_id("longitude").get_attribute("value")
                with open("GeoinformationContainer.txt", "a") as file:
                    file.write("Longitude = " + self.longitude + " широта = " + self.longitude + "\n")
        except BaseException as error:
            print("Пожалуйста, проверьте интернет-соединение :", error)

```

Рисунок 2.23 – Автоматизация процесса

```

browser = webdriver.Chrome(chrome_options=self.chrome_option)
browser.get(self.url)

```

Рисунок 2.24 – Драйвер браузера

На рисунке 2.24 реализован метод, возвращающий текущее значение URL адреса в браузере.

```

self.latitude = browser.find_element_by_id("latitude").get_attribute("value")
self.longitude = browser.find_element_by_id("longitude").get_attribute("value")
while True:
    if self.latitude != browser.find_element_by_id("latitude").get_attribute(
        "value") and self.longitude != browser.find_element_by_id("longitude").get_attribute("value"):
        self.latitude = browser.find_element_by_id("latitude").get_attribute("value")
        self.longitude = browser.find_element_by_id("longitude").get_attribute("value")
        with open("GeoinformationContainer.txt", "a") as file:
            file.write("Longitude = " + self.longitude + " широта = " + self.longitude + "\n")
except BaseException as error:
    print("Пожалуйста, проверьте интернет-соединение :", error)

```

Рисунок 2.25 – Проверка долготы и ширины

Как показано на рисунке 2.25, приложение определяет значения долготы и ширины, которые в дальнейшем будут использоваться для указания координат на карте.

Следует учесть, что данный функционал корректно выполняет свои задачи при условии, что устройство имеет выход в интернет. Поэтому нельзя обойтись без условия, как на рисунке 2.26.

```
except BaseException as error:  
    print("Пожалуйста, проверьте интернет-соединение :", error)
```

Рисунок 2.26 – Проверка наличия интернет-соединения

Далее на рисунке 2.27 создается подключение к поисковым системам на Джанго.

```
if __name__ == '__main__':  
    URL = "https://www.gps-coordinates.net/"  
    TEMP = GeoLocationAutomation(url=URL)  
    TEMP.run_automation()
```

Рисунок 2.27 - Подключение для поиска

## 2.8 Определение IP и местоположения

Объявляем сервисы, которые используются для детализации геолокационных данных устройства (см. рисунок 2.28).

```
from geopy.geocoders import Nominatim  
import sys  
import time
```

Рисунок 2.28 - Сервисы

Далее необходимо реализовать определения адреса в городе и установить местонахождение как на рисунке 2.29 [15].

```

repeat='yes'
while repeat == 'yes' or repeat == 'yes':
    print ("\n[*] Определение IP Aidyn [*]")
    try:
        inputCity=raw_input("[*] Адрес в городе : ")
        geolocator=Nominatim()
        location=geolocator.geocode("%s" % (inputCity))
        for i in(5,4,3,2,1):
            time.sleep(2)
            print ("[*] Поиск ....")
        #time.sleep(2)
        print()
        res=location.address
        res2=location.latitude,location.longitude
        print ("[*] Поиск по адресу : ", res)
        print ("[*] Установление местонахождения : ", res2)

```

Рисунок 2.29 – Определение адреса

Иногда возникает потребность прервать процесс местонахождения. Для этого предусмотрен участок кода, который отображен на рисунке 2.30.

```

except KeyboardInterrupt:
    print ("\n[*] CTRL+C активация")
    sys.exit()
except :
    print ("[*] IP не найден")

```

Рисунок 2.30 – Активация завершения скрипта



## 2.9 Отслеживание IP адреса

```
Выбрать C:\Windows\system32\cmd.exe - python location.py
[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден

[*] Определение IP AIdyn [*]
[*] IP не найден
```

Рисунок 2.31 – Отслеживание IP

На рисунке 2.31 показана командная строка, которая отображает найденные и зафиксированные IP-адреса.

```
C:\Windows\system32\cmd.exe - python web.py
Microsoft Windows [Version 10.0.15063]
(c) Корпорация Майкрософт (Microsoft Corporation), 2017. Все права защищены.

C:\Users\User>cd downloads

C:\Users\User\Downloads>cd mobile-search-master

C:\Users\User\Downloads\mobile-search-master>python web.py
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 104-964-808
* Running on http://127.0.0.1:3000/ (Press CTRL+C to quit)
```

Рисунок 2.32 - Запуск локального сервера

На рисунке 2.32 показан запуск локального сервера, который показывает последние действия, которые проходили на устройстве. (звонки, смс, состояние батарей imei).

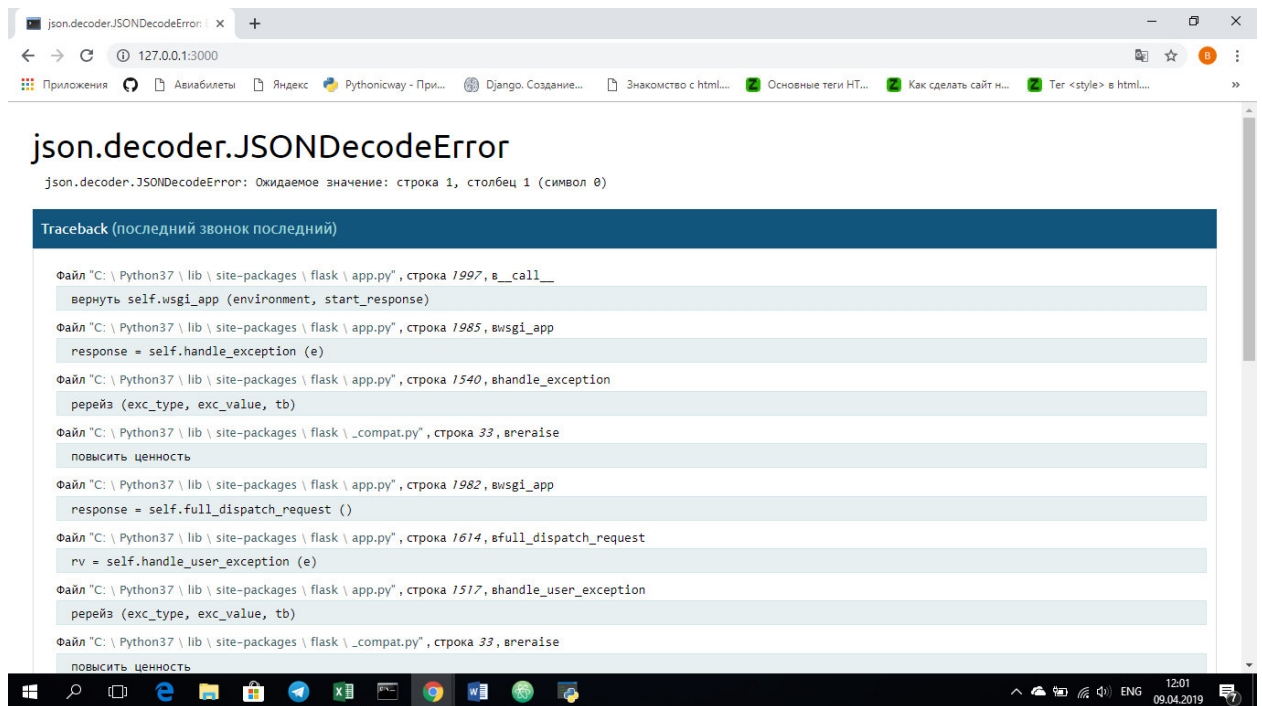


Рисунок 2.33 – Последняя активность

По адресу, который указан на рисунке 2.33, мы определяем последние действия на устройстве.

## 2.10 HTML привязка к шаблону

Приложение, разрабатываемое в рамках данного дипломного проекта, как и любое другое, имеет свой интерфейс.

Интерфейс – это внешнее представление, которое пользователь видит, посещая веб-ресурс или пользуясь приложением на смартфоне или компьютере.

Интерфейс пользователя составляется из языка программирования HTML, языка гипертекстовой разметки.

Элементы тега `<head>` представлены на рисунке 2.34.

```
<title>Мой радар</title>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no">
<meta charset="utf-8">
<link type="text/css" rel="stylesheet" href="/stylesheets/main.css" />
```

Рисунок 2.34 – Элементы тега `<head>`

На этом рисунке показаны три строки, каждая из которых выполняет важные функции.

Первая из них отображает название страницы, которое влияет на поисковую выдачу.

Вторая строка отвечает за отображение страницы в корректных масштабах на мобильных или иных устройствах с отличным разрешением экрана.

Третья строка определяет самую распространенную кодировку.

И наконец последняя строка содержит ссылку на файл стилей, которые влияют на то, как будет отображаться страница.

```
<script>
  // Попробуйте HTML5 геолокации
  if(navigator.geolocation) {
    navigator.geolocation.watchPosition(function(position) {
      data = {
        tag: "{{ tag_id }}",
        latitude: position.coords.latitude,
        longitude: position.coords.longitude,
        accuracy: position.coords.accuracy,
      };
      $.ajax({
        type: 'POST',
        url: '/update',
        data: data,
      });
    }, function() {
      alert('Geolocation failed. Bummer.');
```

Рисунок 2.35 – Скрипт для геолокации

На рисунке 2.35 показана реализация скрипта. В нем используется метод `watchPosition`, который возвращает основные данные.

Помимо долготы и широты возвращаются также значения тег и точность.

Весь процесс работает с помощью Ajax. Он позволяет получать и отправлять данные не перезагружая страницу, что положительно влияет на поведенческий фактор посетителя, который не станет каждый раз перезагружать страницу, лишь бы узнать, изменилось ли что-либо на этот раз.



```

<body>
  {% if tag_id %}
  <p>Ваше местоположение отслеживается на <a href="{{ scheme }}://{{ hostname }}/track?tag={{ tag_id }}" target="_blank">
  { scheme }}://{{ hostname }}/track?tag={{ tag_id }}</a>.</p>
  <p></p>
  {% else %}
  <form action="/" method="post">
    <label for="tag_id">имей код</label>
    <input id="tag_id" autofocus name="tag_id" value="{{ tag_id }}">
    <input class="submit" type="submit" value="Enable Tracking">
  </form>
  {% endif %}
</body>

```

Рисунок 2.36 – Скрипт в пределах тега <body>

На рисунке 2.36 пользователю сообщается, что его местоположение отслеживается.

Прописана форма, которая содержит несколько основных полей, в которые пользователь вносит данные, на основании которых в дальнейшем происходит отслеживание.

При нажатии на кнопку «Enable Tracking» данные отправляются на сервер и запускаются соответствующие скрипты.

## 2.11 Отслеживание

```

<title>мой радар</title>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no">
<meta charset="utf-8">
<link href="/stylesheets/main.css" rel="stylesheet">
{% if latitude %}
<script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true"></script>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>

```

Рисунок 2.37 – Объявление скриптов

На рисунке 2.37 видно, что объявлены мета теги а также настроена кодировка текста на UTF-8, для корректного отображения на различных платформах [16].

Скриптами указываются ссылки на Google карты и Google ajax сервисы. Эти два скрипта являются основными.

```

function initialize() {
  var mapOptions = {
    zoom: 15,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
  };
  var map = new google.maps.Map(document.getElementById('map_canvas'),
    mapOptions);

  var marker = new google.maps.Marker({
    map: map,
    icon: 'data:image/png;base64,iVBORw0KGgoAAAANSUHEuGAAAAoAAAAKAQMAAAC3/F3
+AAAACXBIWXMAAA7DAAAOwNHb6hkAAAAAXNSR0IArs4c6QAAARnQU1BAACxjwv8YQUAAAgY0hSTQAAeiYAAICEAAD6AAAAgOgAAHUwAADqYAAAOpg
gAABdwnLpRPAAAAANQTRFALfvPEv6TAAAAATJREFUCB1jYMAHAAeAAEBGNlaAAAAAE1FTkSuQmCC',
  });
}

```

Рисунок 2.38 – Функция инициализации

Функция, описанная на рисунке 2.38, инициализирует сервисы, данные, а также карту, на которой в дальнейшем будут отображаться устройства, за которыми ведется слежение.

Обновление карты происходит в режиме реального времени. Т.е. при передвижении точки на карте, в определенном радиусе (в зависимости от установленного масштаба) загружаются все близлежащие объекты, такие как здания, улицы и т.д.

Код, отвечающий за обновление карты показан на рисунке 2.39

```

updateMap(map, marker, {{ latitude }}, {{ longitude }});

```

Рисунок 2.39 – Обновление карты

```

setInterval(function () {
  $.ajax({ url: "/get?tag={{ tag_id }}", success: function(data){
    updateMap(map, marker, data.latitude, data.longitude);
  }, dataType: "json", timeout: 12000 });
}, 15000);
}

```

Рисунок 2.40– Установка интервала

Устанавливается интервал, через который происходит проверка, нужно ли обновить карту.

Передача информации реализована в json формате, что обеспечивает высокую скорость и высокую степень защищенности.

Непосредственно функция, которая обновляет карту и создает новую позицию отслеживаемого на ней объекта, показана на рисунке 2.41.

```
function updateMap(map, marker, lat, long) {
    var pos = new google.maps.LatLng(lat, long);
    marker.setPosition(pos);

    map.setCenter(pos);
}
```

Рисунок 2.41 – Функция updateMap()

На рисунке 2.42 показан метод добавления карты и ее обработчика на структуру страницы.

```
google.maps.event.addDomListener(window, 'load', initialize);
</script>
{% endif %}
</head>
<body>
    {% if error %}
    <p>{{ error }}</p>
    {% else %}
    <div id="map_canvas"></div>
    {% endif %}
</body>
</html>
```

Рисунок 2.42 – Функция addDomListener

## 2.12 Стартовая страница

```
<head>
    <title>Мой радар</title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <link type="text/css" rel="stylesheet" href="/stylesheets/main.css" />
</head>
```

Рисунок 2.43 – Тег <head>

Объявление заголовка, поддержки мобильной графики, а также файла разметки страницы представлены на рисунке 2.43.

```
<body>

  <form action="/" method="post">
    <label for="tag_id">имей код</label>
    <input id="tag_id" autofocus name="tag_id" value="{{ tag_id }}">
    <input class="submit" type="submit" value="Enable Tracking">
  </form>

</body>
```

Рисунок 2.44 – Содержимое тега <body>

На рисунке 2.44 показана реализация формы, на которой вводится такая информация, как:

- имеи код;
- имя.

Кроме того отображается кнопку, при нажатии на которую данные отправляются на сервер, для ее дальнейшей обработки.

```
import webapp2
import jinja2
import os
import json
import re

from google.appengine.ext import db
```

Рисунок 2.45 – Импорт сервисов

На рисунке 2.45 показываються сервисы, которые необходимо импортировать для корректности работы приложения.

```
jinja_environment = jinja2.Environment(
    loader=jinja2.FileSystemLoader(os.path.dirname(__file__)))
```

Рисунок 2.46 – создание виртуальной среды

На рисунке 2.46 показан код, который выполняет создание виртуальной среды.

На рисунке 2.47 изображен класс Tag, который определяет свойства долготы и широты

```
class Tag(db.Model):
    latitude = db.FloatProperty()
    longitude = db.FloatProperty()
    accuracy = db.IntegerProperty()
```

Рисунок 2.47 – класс Tag

```
class MainPage(webapp2.RequestHandler):
    def get(self):
        template_values = {
            'tag_id': ''
        }

        template = jinja_environment.get_template('index.html')
        self.response.out.write(template.render(template_values))
```

Рисунок 2.48 – Класс MainPage

На рисунке 2.48 показан класс MainPage, который выполняет запуск основного шаблона и другие сервисы.

```
def post(self):
    if os.environ.get('HTTP_HOST'):
        hostname = os.environ['HTTP_HOST']
    else:
        hostname = os.environ['SERVER_NAME']

    scheme = os.environ.get('wsgi.url_scheme', 'http')

    template_values = {
        'tag_id': re.sub(r'\W+', '', self.request.get('tag_id')),
        'hostname': hostname,
        'scheme': scheme,
    }
    template = jinja_environment.get_template('track.html')
    self.response.out.write(template.render(template_values))
```

Рисунок 2.49 – Определение Post



Определение post выполняет передачу и получение той или иной информации об устройстве, за которым ведется слежение.

Такие данные как tag\_id, hostname, scheme определяют и идентифицируют устройство, будь то гаджет, часы, смартфон или ноутбук.

```
class Tracker(webapp2.RequestHandler):
    def get(self):
        tag_id = re.sub(r'\W+', '', self.request.get('tag'))
        if tag_id != "":
            position = Tag.get_by_key_name(tag_id)

            if position != None:
                template_values = {
                    'tag_id': tag_id,
                    'latitude': position.latitude,
                    'longitude': position.longitude,
                    'accuracy': position.accuracy
                }
            else:
                template_values = {
                    'error': "Нет информации для отслеживания этого тега."
                }
            template = jinja_environment.get_template('tracker.html')
            self.response.out.write(template.render(template_values))
```

Рисунок 2.50 – класс Tracker

На рисунке 2.50 показан класс Tracker.

С помощью этого класс проводится определение местоположение устройства, проверка его последнего подключения к сети интернет, а также сохранение последних данных перед обрывом доступа к сети.

```
class Update(webapp2.RequestHandler):
    def post(self):
        tag_id = re.sub(r'\W+', '', self.request.get('tag'))
        latitude = self.request.get('latitude')
        longitude = self.request.get('longitude')
        accuracy = self.request.get('accuracy')

        if tag_id != "" and latitude != "" and longitude != "" and accuracy != "":
            tag_id = str(tag_id)

            tag = Tag(key_name=tag_id)
            tag.latitude = float(json.loads(latitude))
            tag.longitude = float(json.loads(longitude))
            tag.accuracy = int(json.loads(accuracy))
            tag.put()
```

Рисунок 2.51 - класс Update

На рисунке 2.51 видно, как работает класс Update, который обновляет всю информацию, которая дает представление о местоположении в реальном времени.



```
class GetLocation(webapp2.RequestHandler):
    def get(self):
        tag_id = self.request.get('tag')
        if tag_id != "":

            position = Tag.get_by_key_name(tag_id)

            if position != None:
                self.response.out.write(json.dumps({
                    'latitude': position.latitude,
                    'longitude': position.longitude}))

app = webapp2.WSGIApplication([('/', MainPage),
                               ('/track', Tracker),
                               ('/update', Update),
                               ('/get', GetLocation)],
                              debug=False)
```

Рисунок 2.52 – класс GetLocation

## 2.13 Интерфейс

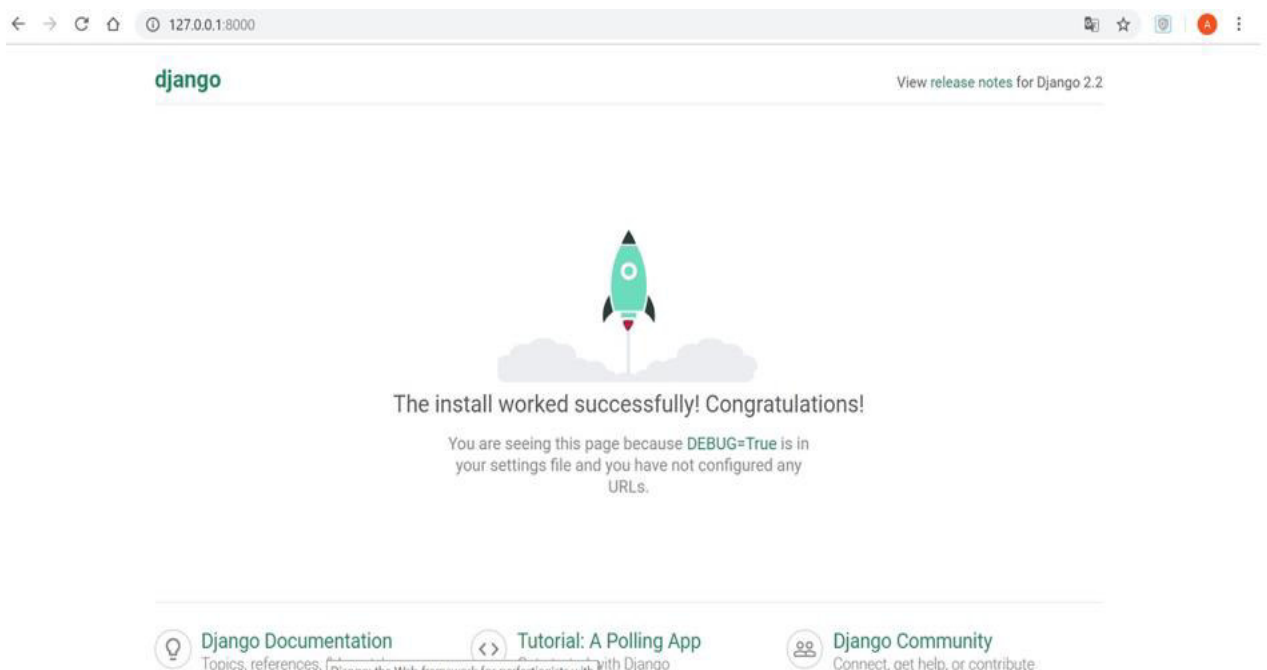


Рисунок 2.53 – Интерфейс регистрации пользователя

Новые пользователи могут зарегистрироваться на данной странице. После регистрации их данные попадают на сервер и устройство, добавляется в базу данных. Таким образом, в дальнейшем оно может быть распознано и отслежено на дальнем расстоянии в реальном времени.

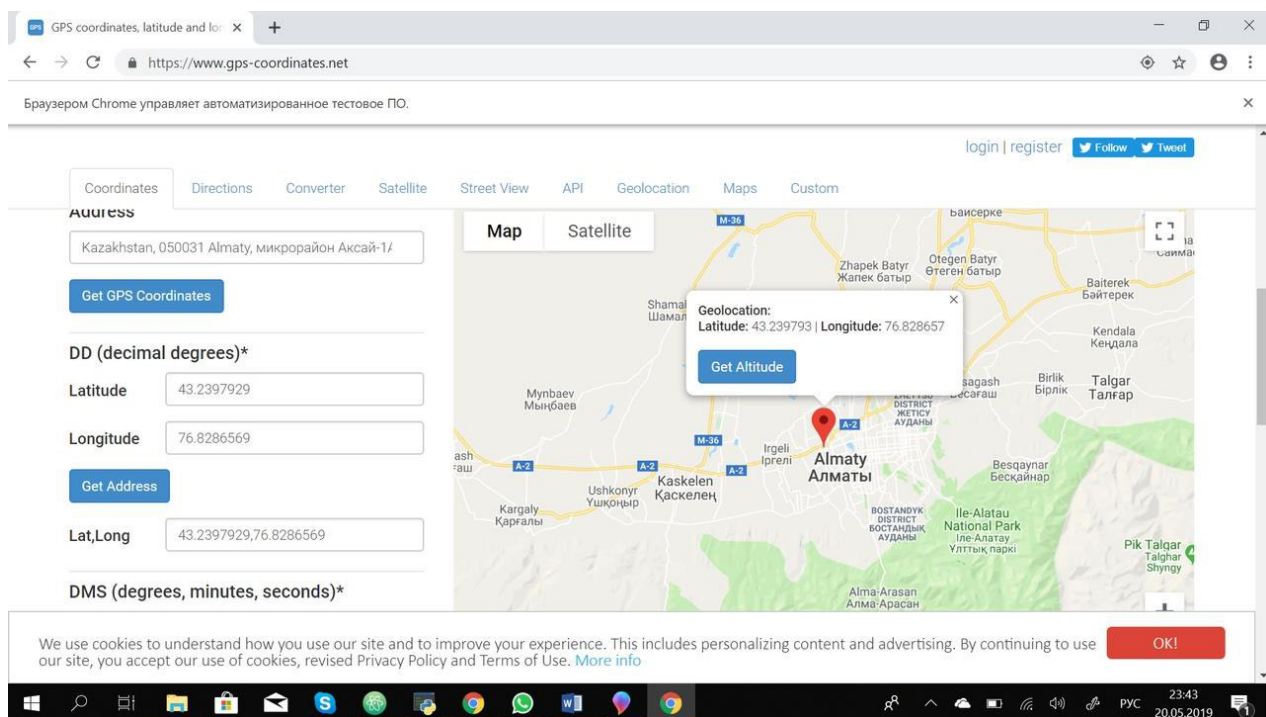


Рисунок 2.54 – Главная страница приложения

На данной странице представлен главный функционал приложения. Разметка страницы состоит из двух колонок. Первая включает в себя данные об устройстве и его местоположении. Вторая колонка содержит карту, с отмеченными на ней флажками. Флажки в данном случае, изображают местоположение в данный момент времени.

### 3 Ожидаемые технико-экономические показатели

Тема дипломного проекта – «Разработка системы слежения и поиска потерянных мобильных устройств».

Целью экономического обоснования дипломного проекта является расчет и анализ затрат, необходимых для создания и реализации системы слежения и поиска потерянных мобильных устройств, изучить затраты на этапах проектирования, разработки, внедрения и функционирования системы; определить условия и сроки окупаемости затрат. Обосновать рациональность проектирования данного веб-приложения.

В данной главе производится экономический расчет коммерческой стоимости исследования проекта. Расчеты учитывают расходы на создание и внедрение информационной системы.

Учитывая, что количество пользователей интернета растет, а также беря во внимание перспективность и не большую величину затрат на использование интернет-ресурсов, вопрос создания системы наиболее актуальным, т.к. имеет ряд преимуществ. Помимо всего этого, следует взять во внимание, что приложение затрагивает очень важную тематику, связанную с социальными группами людей, к которым относятся дети, старики и люди с ограниченными возможностями, а потому имеет довольно широкий охват пользователей.

Отсутствует необходимость в больших рабочих зонах, и большого количество сотрудников. не имеет ограничений по зоне действия, т.к. потенциальный пользователь имеет доступ к веб-приложению из любой точки мира, не требует мощной аппаратуры или специфичного программного обеспечения. Приложение совместимо со всеми современными смартфонами под ОС Android.

Приложение должно быть легким в использовании, иметь понятный функционал, т.к. целевой аудиторией являются не профессиональные IT-специалисты, а простые пользователи гаджетов.

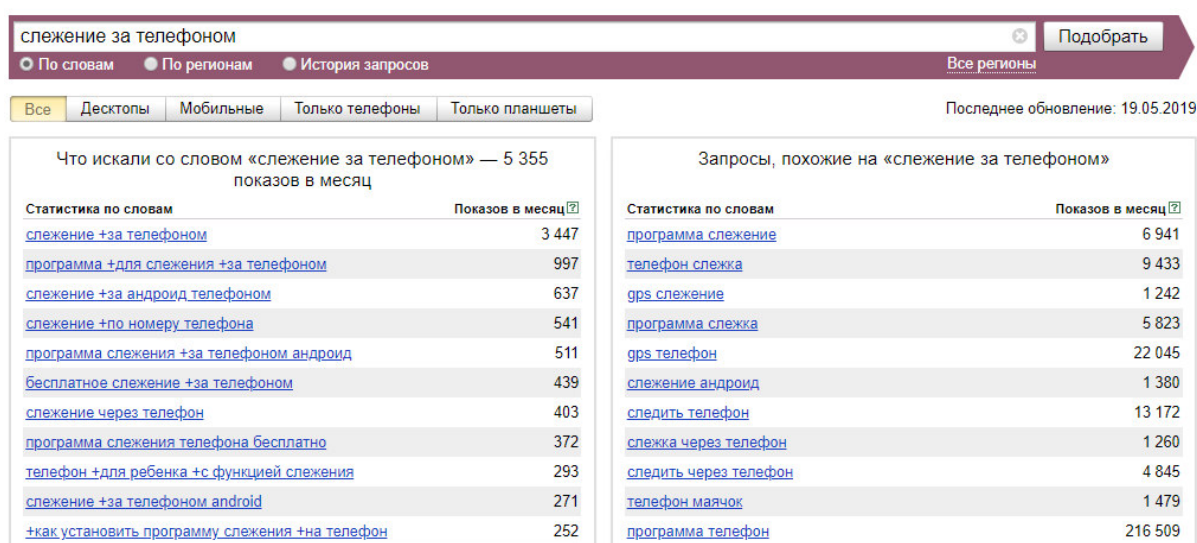


Рисунок 3.1 - Статистика поисковых запросов в wordstat.yandex

Исходя из статистики, за последний месяц в одной из поисковых систем совершали поиск по фразе «слежение за телефоном» 5355 человек.

### 3.1 Расчет трудоемкости разработки программного продукта

Для реализации проекта необходимы финансовые, материальные и трудовые ресурсы;

Для определения трудоемкости разработки приведен перечень всех основных этапов и видов работ, которые должны быть выполнены.

Форма разделения работ по этапам с указанием трудоемкости их выполнения приведена в таблице 3.1 [17].

Таблица 3.1 - Трудоемкость этапов работ

Этап разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП, ч.
1. Техническое задание	1. Постановка задачи	3
	2. Сбор материалов и анализ существующего программного обеспечения	3
	3. Определение требований к системе	1
2. Эскизный проект	Функционал, строение и дизайн приложения	100
3. Технический проект	1. Выбор инструментальных средств	1
	2. Определение требований к аппаратному обеспечению	1
4. Рабочий проект	1. Разметка таблиц структуры БД	10
	2. Программирование	320
	3. Тестирование и отладка	50
	4. Создание модулей синхронизации с иными сторонними сервисами	25
	5. Наполнение контентом	6
ИТОГО трудоемкость выполнения проекта		520

Поскольку количество часов активной работы по разработке программного продукта равно 520, а в сутки на разработку выделялось 8 часов, следовательно, срок выполнения проекта равен 65 суток. Для дальнейших расчетов время разработки программного продукта округляем до двух месяцев.

### 3.2 Расчет затрат на разработку программного продукта

Общая сумма затрат на материальные ресурсы ( $Z_M$ ) определяется по формуле:

$$Z_M = \sum_{i=1}^n P_i * C_i \quad (3.1)$$

где  $P_i$  - расход  $i$ -го вида материального ресурса, натуральные единицы;

$C_i$  - цена за единицу  $i$ -го вида материального ресурса, тг;

$i$  - вид материального ресурса;

$n$  - количество видов материальных ресурсов.

Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 3.2.

Таблица 3.2 - Затраты на материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
1. Картридж для принтера	шт.	1	10500	10 500
2. Бумага	пачка	1	400	400
ИТОГО затраты на материальные ресурсы				10 900

Общая сумма затрат [18] на электроэнергию ( $Z_Э$ ) рассчитывается по формуле:

$$Z_Э = \sum_{i=1}^n M_i \times K_i \times T_i \times C \quad (3.2)$$

где  $M_i$  - паспортная мощность  $i$ -го электрооборудования, кВт;

$K_i$  - коэффициент использования мощности  $i$ -го электрооборудования (принят  $K_i=0.7$ );

$T_i$  - время работы  $i$ -го оборудования за весь период разработки ПП ч;

$C$  - цена электроэнергии, тг/кВт×ч;

$i$  - вид электрооборудования;

$n$  - количество электрооборудования.

Затраты на электроэнергию приведены в таблице 3.3.

Таблица 3.3 - Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэфф-т использ-я мощности	Время работы оборуд-я для разработки ПП, ч	Цена электро энергии, $\frac{т}{кВт \times ч}$	Сумма, тг
1. ПК Samsung	0,4	0,7	320	16,5	1478,4
2. ПК Apple	0,2	0,7	25	16,5	57,75
3. Ноутбук HP	0,2	0,7	75	16,5	173,25
4. Ноутбук Lenovo	0,2	0,7	100	16,5	231
5. Модем	0,2	0,7	520	16,5	1201,2
Итого затраты на электроэнергию					3141,35

Общая сумма затрат на оплату труда ( $Z_{TP}$ ) определяется по формуле:

$$Z_{TP} = \sum ЧС_i \times T_i \quad (3.3)$$

где  $ЧС_i$  - часовая ставка  $i$ -го работника, тг;

$T_i$  - трудоемкость разработки ПП, чел.×ч;  $i$  - категория работника;

$n$  - количество работников, занятых разработкой ПП.

Часовая ставка программиста составляет 800 (тг/ч), трудоемкость разработки – 320 ч.

Часовая ставка WEB-дизайнера 800 (тг/ч), трудоемкость – 100 ч.

Часовая ставка контент-менеджера 400 (тг/ч), трудоемкость – 25.

Часовая ставка научного руководителя составляет 1000 (тг/ч), трудоемкость – 75 ч.

$$Z_{TP} = 800*320+800*100+400*25+1000*75 = 421000\text{тг.}$$

Затраты на оплату труда приведены в таблице 3.4.

Таблица 3.4 - Затраты на оплату труда

Категория работника	Квалиф-ция	Трудоемкость разработки ПП, ч	Часовая ставка, тг/ч	Сумма, тг
1. Программист	Senior-developer	320	800	256 000
2. WEB - дизайнер		100	800	80000
3. Контент-менеджер		25	400	10000

Продолжение таблицы 3.4

4. Научный руководитель	Руководитель проекта	75	1000	75000
ИТОГО затраты на оплату труда				421 000

Сумма годовых амортизационных отчислений определяется по формуле:

$$A = \text{Перв. стоимость} * \text{Норма амортизации} / 100, \quad (3.4)$$

Амортизационные отчисления приведены в таблице 3.5.

Таблица 3.5 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг.	Годовая норма амортизации, %	Сумма амортизации в год, тг.	Сумма амортизации за 2 месяца, тг.
Модем D-LINK	4000	40	1600	265
ПК Samsung	150000	40	60000	10 000
ПК Apple	300000	40	120000	20 000
Ноутбук Lenovo	200000	40	80000	13 333
Ноутбук HP	100000	40	40000	66 667
ОС Windows 10	бесплатно			
CP Visual Studio	бесплатно			
ИТОГО амортизация основных фондов				110 265

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ:

$$H_{A_i} = \frac{100}{T_{N_i}} \quad (3.5)$$

Сумма амортизации за один месяц =  $A / 12$ .

Сумма амортизационных отчислений за три месяцев равна 110 265тг.

В статью «Прочие затраты» включаются расходы на арендную плату, включая коммунальные платежи, затраты на лицензирование и



сертификацию, расходы на рекламу, канцелярские и прочие хозяйственные расходы.

Стоимость аренды помещения на месяц равна 40000 тг. (в эту сумму включены коммунальные услуги).

Арендная плата рассчитывается по формуле:

$$АП = Ca * S, \quad (3.6)$$

где  $Ca$  – срок аренды;

$S$  – стоимость аренды за 1 месяц.

$$АП = 40\,000 * 2 = 80\,000 \text{ тг.}$$

Расходы на интернет, месячная оплата которого составляет 4500 тг равны:

$$P_{и} = 2 * 4\,500 = 9\,000 \text{ тг.}$$

Оплата услуг хостинг компании, предоставляющей аренду дискового пространства (5 000 тг/Gb в год) и доменных имен (1800 тг за 1 доменное имя), в данном проекте предусмотрено 3 Gb дискового пространства и 1 доменное имя:

$$P_{х} = 5\,000 * 2 + 1\,800 = 11\,800 \text{ тг.}$$

Прочие хозяйственные расходы составляют 5 000 тг.

Прочие затраты = 80000 + 9000 + 5 000 + 11800 = 105 800 тг.

Социальный налог, согласно Налоговому кодексу РК, составляет 9,5 % от затрат на оплату труда. Пенсионные отчисления не облагаются социальным налогом.

$$CO = (Z_{тр} - O_{п}) * 0,095,$$

где  $O_{п}$  – отчисления в пенсионный фонд, 10% от ФОТ;

$Z_{тр}$  – затраты на оплату труда.

Для программиста:

$$O_{п} = Z_{тр} * 10\% = 256\,000 * 0,1 = 25\,600 \text{ тг.}$$

$$O_{с} = (256\,000 - 25\,600) * 0,095 = 21\,888 \text{ тг.}$$

Для WEB-дизайнера:

$$O_{п} = Z_{тр} * 10\% = 80\,000 * 0,1 = 8\,000 \text{ тг.}$$

$$O_{с} = (80\,000 - 8\,000) * 0,095 = 6\,840 \text{ тг.}$$

Для контент-менеджера:

$$O_{п} = Z_{тр} * 10\% = 2\,400 * 0,1 = 240 \text{ тг.}$$

$$O_c = (2400 - 240) * 0,095 = 205,2 \text{ тг.}$$

Для дип. Рук.:

$$O_{п} = Z_{тр} * 10\% = 75\,000 * 0,1 = 7\,500 \text{ тг.}$$

$$O_c = (75\,000 - 7\,500) * 0,095 = 6\,412,5 \text{ тг.}$$

Итого: 42 545,7 тг.

На основании полученных данных по отдельным статьям в таблице 3.6 приведена смета затрат на разработку ПП

Таблица 3.6 - Смета затрат на разработку ПП

Статьи затрат	Сумма, тг
1. Материальные затраты, в том числе:	
- материалы	10 900
- электроэнергия	4818,66
2. Затраты на оплату труда.	421 000
3. Отчисления на социальные нужды.	42 545,7
4. Амортизация основных фондов.	110 265
5. Прочие затраты.	155 300
<b>ИТОГО по смете</b>	<b>744 829,36</b>

### 3.3 Определение возможной (договорной) цены программного обеспечения

Величина возможной (договорной) цены ПП должна устанавливаться с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена ( $C_d$ ) для прикладных ПП рассчитывается по формуле:

$$C_d = Z_{нир} * (1 + (P/100)), \quad (3.7)$$

где  $Z_{нир}$  - затраты на разработку ПП (из таблицы 3.6), тг;

$P$  - средний уровень рентабельности ПП. % (принято 25%).

$$C_d = 744\,829,36 * (1 + 0,25) = 931\,036,7 \text{ тг.}$$

Цена реализации с учетом НДС рассчитывается по формуле:

$$C_p = C_d + C_d * \text{НДС.}$$

НДС, согласно Налоговому кодексу РК, составляет 12 %.

$$C_p = 931\,036,7 + 931\,036,7 * 0,12 = 1\,042\,761,1 \text{ тг.}$$

### 3.4 Расчет срока окупаемости программного продукта

По статистике за последние девять месяцев в Казахстане было зафиксировано 25000 краж мобильных телефонов. В среднем стоимость одного смартфона – 80000 тг. В общей сложности, в результате потерь или краж телефонов люди теряют около 2 000 000 000 тг.

В результате использования системы слежения и поиска потерянных мобильных устройств возможна экономия расходов на покупку новых смартфонов более чем в 2 раза, т.к. поддержка веб-приложения не требуется, а своим функционалом приложение создает шансы возврата смартфонов в 2 раза [19].

После введения в эксплуатацию системы слежения, расходы сокращаются в 2 раза. Соответственно расходы теперь составляют 1 000 000 000 тг.

Расчетный срок окупаемости продукта можно найти по формуле:

$$T_{\text{ок}} = C / \text{Э}, \quad (3.8)$$

где С - затраты на разработку и внедрение системы, тенге;

П - прибыль, тенге/месяц.

$$T_{\text{ок}} = 1\,042\,761,1 / 1\,000\,000\,000 = 0,010 \text{ (месяцев)}.$$

В данном случае срок окупаемости проекта составит менее месяца.

### 3.5 Оценка социально-экономических результатов

К социально-экономическим показателям функционирования информационной системы является обработка таких факторов как:

- качество процесса управления;
- длительность и сроки проектирования программного продукта;
- расходы на реализацию приложения и эксплуатацию программного продукта;
- количеству разработчиков.

Эффективность программного продукта определяется его качеством и эффективностью процесса разработки и сопровождения. С позиции специалиста эффективность процесса разработки обуславливается отсутствием необходимости в мощном оборудовании. Все что необходимо специалисту для осуществления процесса разработки – компьютер средней мощности с доступом к интернету. С позиции использования ресурсов пользователи не ограничены ничем, кроме необходимости иметь доступ к интернету [20].

Разработка системы слежения занимает 520 часов (2 месяца). Из них 9 часов занимает процесс постановки задачи и проектирование приложения, 100

– функционал, строение и дизайн приложения, 2 – выбор инструментария, 411 – разработка БД, программирование, тестирование и отладка, модули и наполнение контентом. В процессе разработки участвуют 4 человека-программист, WEB-дизайнер, контент-менеджер, Project manager (дипломный руководитель). Расходы на реализацию программного продукта составляют 744 829,36 тенге. Договорная цена – 931 036,7 тг. Цена реализации- 1, 042 761тг. Сумма эксплуатации программного продукта составляют 4000 тенге в год, т. к. это сумма размещения сайта на хостинге [21].

Данные вложения обязательно окупятся за счёт значительного социально-экономического результата. Минимизируются трудовые затраты и время обработки результатов.

Высокая эффективность сопровождения веб-приложения обуславливается беспрепятственной расширением его функционала. Например, нет необходимости приостанавливать работу приложения для внесения функционала латинизации EXCEL документов. Перспективность актуальности веб-конвертера поддерживается наличием соответствующих законодательных актов о переходе казахского алфавита на латинскую графику.

Использование конечного программного продукта не требует какой-то особенной подготовки или квалификации. Любой сотрудник, имеющий опыт использования веб-ресурсов, справится с поставленной задачей.

## 4 Безопасность жизнедеятельности

### 4.1 Анализ условий труда

В дипломной работе мы проектируем геоинформационную систему для логистического центра. Для её эксплуатации требуется участие людей, поэтому необходимо помещение типа «операторская», в котором будет работать шесть сотрудников. В помещении есть шесть рабочих мест с установленным на нём компьютером и мониторами. Все оборудование устанавливается в помещении длиной  $A = 17$  м и шириной  $B = 11$  м, высота потолка  $H = 5$  м.

Для создания благоприятных условий труда, помещение для работы должно соответствовать установленным нормам. На данный момент на предприятии имеется помещение размерами  $17 \times 11 \times 5$  м. Естественное освещение обеспечивают оконные проемы в количестве 4 штук размером  $7,2$  м<sup>2</sup>. Помимо естественного источника света, в помещении есть и искусственные источники освещения, представляющие собой лампы накаливания мощностью  $60$  Вт [22]. Естественное и искусственное освещение помещения соответствуют нормам. В виду отсутствия специальных производственных приборов и установок, уровень шума в помещении соответствует нормам.

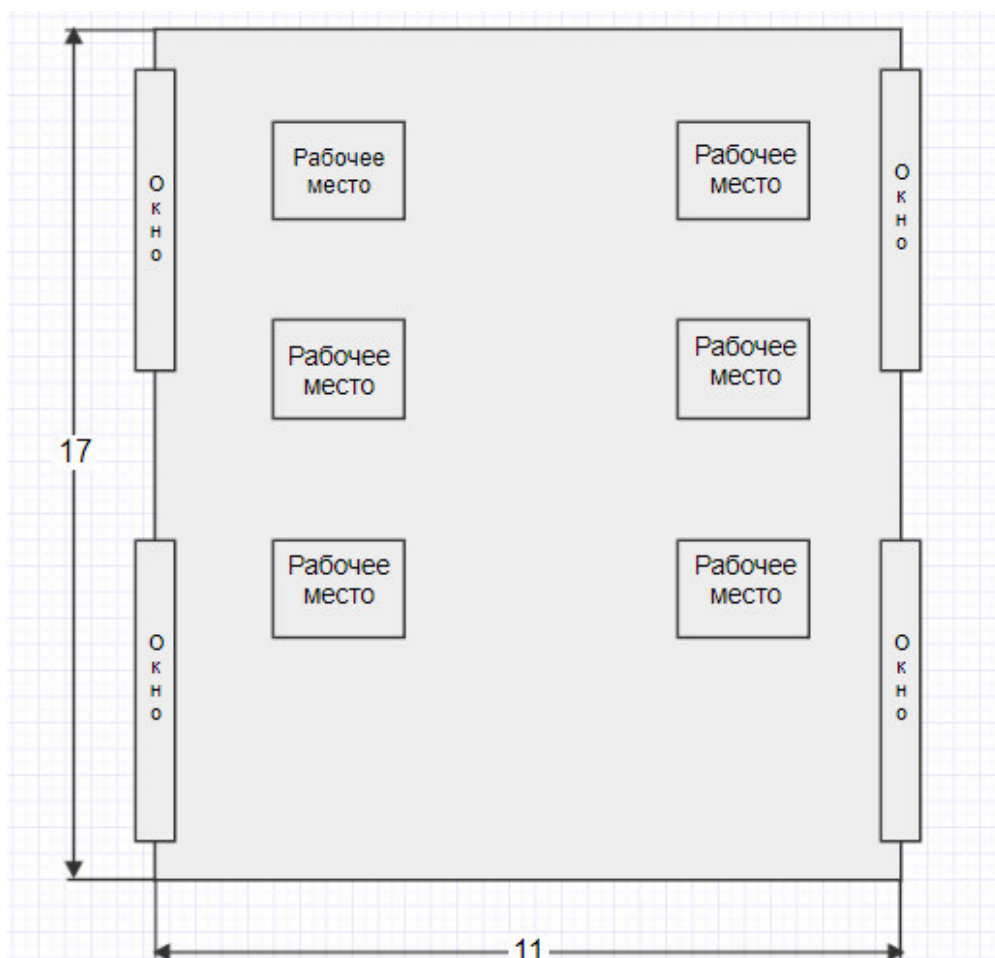


Рисунок 4.1 – Схема помещения

Оценка микроклимата на производстве выполняется в рабочей зоне, в пространстве, высота которого равна до 2 метров над поверхностью уровня пола, на которой расположено место пребывания сотрудников. Из таблицы 4.1 [10], определяем категорию работ: в нашем случае, как легкая I,а. Параметр влажности воздуха оценивается содержанием водяных паров [23].

На данный момент условия труда на предприятии, не соответствуют нормам. Необходимо провести реконструкцию текущей аспирационной системы для обеспечения благоприятных условий труда. Для того чтобы все микроклиматические условия поддерживались в соответствии с требованиями «Санитарных норм, микроклимата производственных помещений» и в целях нормальной работы оборудования требуется установка кондиционера. Показатели норм микроклимата показаны в таблице 4.1 на основе [10].

Таблица 4.1 – Нормы микроклимата производственных помещений при выполнении работ средней тяжести

Категория работ	Период года	Температура, °С		Нормальная влажность, %		Скорость движения воздуха, м/с	
		Оптим.	Допус.	Оптим.	Допус.	Оптим.	Допус.
Лёгкая I,а	Холодный	18-20	17-23	40-60	75	0,2	0,1 не более
	Тёплый	21-23	18-27	40-60	65, при 26 °С	0,3	0,2-0,4

## 4.2 Расчет тепловых нагрузок в помещении

В помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние) [24].

$$Q_{\text{отр}} = V_{\text{пом}} \cdot X_o \cdot (t_{\text{Нрасч}} - t_{\text{Врасч}}), \text{ Вт} \quad (4.1)$$

где  $V_{\text{пом}}$  – объем помещения, м<sup>3</sup>:

$$V_{\text{пом}} = 17 \cdot 11 \cdot 5 = 935 \text{ м}^3;$$

$X_o$  – удельная тепловая характеристика, Вт/м<sup>3</sup>°С:

$$X_o = 0.42 \text{ Вт} / \text{м}^3 \text{ } ^\circ\text{С} ;$$

$t_{\text{Нрасч}}$  – наружная температура (параметр А). Для холодного периода – средняя температура самого холодного месяца в 13 часов, для теплого периода – средней температуре самого жаркого месяца в 13 часов.

$t_{\text{Врасч}}$  – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года  
 $t_{Нрасч} = 29,4 \text{ }^{\circ}\text{C}$   
 $t_{Врасч} = 24 \text{ }^{\circ}\text{C}$   
 $Q_{огр} = 935 \cdot 0,42 \cdot 5,4 = 2120,58 \text{ Вт}$

Для холодного времени года  
 $t_{Нрасч} = -9 \text{ }^{\circ}\text{C}$   
 $t_{Врасч} = 20 \text{ }^{\circ}\text{C}$   
 $Q_{огр} = 935 \cdot 0,42 \cdot |-29| = 11388,3 \text{ Вт}$

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсивность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле

$$Q_p = (q^I F_0^I + q^II F_0^{II}) \cdot \beta_{с.з.} \quad (4.2)$$

где  $q^I$ ,  $q^{II}$  – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м<sup>2</sup>;

$F_{Io}$ ,  $F_{IIo}$  – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м<sup>2</sup>;

$\beta_{с.з.}$  – коэффициент теплопропускания.

$\beta_{с.з.} = 0.15$

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение  $F_{Io} = F_o$ ;  $F_{IIo} = 0$ , (5.3) [1]:

$$Q_p = q^I F_o \cdot \beta_{с.з.} = (q_{пр} + q_{рр}) \cdot K_1^c \cdot K_2 \cdot \beta_{с.з.} \cdot n \cdot S_o, \text{ Вт} \quad (4.3)$$

где  $q_{пр}$ ;  $q_{рр}$  – тепловые потоки от прямой рассеянной радиации, Вт/м<sup>2</sup>. По таблице для СЗ после полудня в 12-13 ч. при расположении З:

$q_{пр} = 63 \text{ Вт/м}^2$ ;  $q_{рр} = 65 \text{ Вт/м}^2$ ;

$F_o = n S_o = 4 \cdot 7,2 = 28,8 \text{ м}^2$  – площадь светового проема ( $n$  – число окон;  $S_o$  – площадь 1 окна);

$K_1$  – коэффициент затемнения остекления переплетами ( $K_{C1}$  – для облученных проемов).

По таблице:

$K_{C1} = 0.72$ ;



$K_2$  – коэффициент загрязнения остекления. По таблице :

$$K_2 = 0.9.$$

Тогда:

$$Q_p = (63+65) \cdot 0.72 \cdot 0.9 \cdot 0.72 \cdot 0.15 \cdot 28.8 = 358.32 \text{ Вт}$$

#### 4.3 Расчет внутренних тепловых нагрузок в помещении

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях слагаются в основном из тепла [25]:

- выделяемого людьми;
- выделяемого лампами и осветительными, электробытовыми приборами;
- выделяемого компьютерами, печатающими устройствами фотокопировальными машинами пр.

В производственных и технологических помещениях различного назначения дополнительными источниками тепловыделений могут быть: нагретое производственное оборудование, горячие материалы, в том числе жидкости и различного рода полуфабрикаты, продукты сгорания и химических реакций.

Теплопоступления от людей зависят от интенсивности выполняемой работы и параметров окружающего воздуха. Тепло, выделяемое человеком, складывается из ощутимого (явного), то есть передаваемого в воздух помещения путем конвекции и лучеиспусканий, и скрытого тепла, затрачиваемого на испарение влаги с поверхности кожи и из легких.

По таблице летом при  $24^\circ\text{C}$  один мужчина выделяет явного тепла 67 Вт, а общего – 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$$Q_l^y = 67 * 3 + 67 * 3 = 402 \text{ Вт.}$$

А выделение общего тепла:

$$Q_l^o = 102 * 3 + 102 * 3 * 0,85 = 566,1 \text{ Вт.}$$

По таблице зимой при  $20^\circ\text{C}$  один мужчина выделяет явного тепла 82 Вт, а общего – 103 Вт. Тогда выделение явного тепла в помещении составит:

$$Q_3^y = 82 * 3 + 82 * 3 = 492 \text{ Вт.}$$

А выделение общего тепла:

аспирационный тепловой кондиционер

$$Q_3^o = 103 * 3 + 103 * 3 * 0,85 = 571,65 \text{ Вт.}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле (4.4) [1]:

$$Q_{осв} = \eta \cdot N_{осв} \cdot F_{пол}, \text{ Вт} \quad (4.4)$$

где  $\eta$  – коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп  $\eta=0.5-0.6$ );

$N_{осв}$  – установленная мощность ламп ( $N=60 \text{ Вт/м}^2$ );

$F_{пол}$  – площадь пола:  $F_{пол} = 17 \cdot 11 = 187 \text{ м}^2$

Тогда:

$$Q_{осв} = 0,5 \cdot 60 \cdot 187 = 5610 \text{ Вт.}$$

Тепло, выделяемое производственным оборудованием, определяется по формуле (4.5) [1]:

$$Q_{об} = N_{ум} \cdot K \quad (4.5)$$

$$Q_{об} = 1,8 \cdot 11 \cdot 0,95 = 18,81 \text{ кВт}$$

Теплопритоки, возникающие за счет находящейся оргтехники, - это 30% мощности оборудования:

$$Q_{орг} = 1,8 \cdot 11 \cdot 0,3 = 5,94 \text{ кВт}$$

#### 4.4 Расчет теплового баланса помещения

На основании выполненных расчетов составим баланс теплопоступлений в помещении:

Лето:

$$Q_{изб} = 358,32 + 402 + 5610 + 18810 + 5940 - 2120,58 = 33240,9 \text{ Дж}$$

Зима:

$$358,32 + 492 + 5610 + 18810 + 5940 + 11388,3 = 42598,62 \text{ Дж}$$

Так как тепловой баланс для зимы больше летнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле:

$$Q_n = (Q_{изблето} \cdot 860) / V_{пом} = (42598,62 \cdot 860) / 935 = 39,2 \text{ ккал/м}^3$$

При  $>20 \text{ ккал/м}^3$ ,  $=8 \text{ }^\circ\text{C}$ ,

Определение количества воздуха, необходимое для поступления в помещение:

$$L = (Q_{изб} \cdot 860) / (C \cdot \gamma) = (42598,62 \cdot 860) / (0,24 \cdot 8 \cdot 1,206 \cdot 10^4) = 1693,56 \text{ м}^3/\text{час}$$

где  $C = 0,24 \text{ ккал/(кг}^\circ\text{C)}$  – теплоемкость воздуха,

$\gamma = 1,206 \cdot 10^4 \text{ кг/м}^3$  – удельная масса приточного воздуха.

Определение кратности воздухообмена:

$$N=L/V_{\text{пом}}=1693,56/935=15,2 \text{ час-1}$$

#### 4.5 Выбор кондиционера. Схема расположения

Исходя из полученных данных, выберем прецизионный кондиционер с верхней подачей воздуха модели SUA 0351, который будет удовлетворять полученным требованиям по необходимому количеству воздуха.

Таблица 4.2 – Характеристики кондиционера

Электропитание	Расход воздуха, внутри блок	Расход воздуха, внешний блок	Произв. По теплу	Произв. По холоду	Мощность компрессора	Электронагреватель	Увлажнитель	Расход пара	высота	ширина	глубина	масса
В/Ф/Гц	м <sup>3</sup> /ч	м <sup>3</sup> /ч	кВт	кВт	кВт	кВт	кВт	Кг/ч	мм	мм	мм	Кг
400/3/5+N	4020	5970	19,7	19,7	5,5	7,6	4,3	3	1740	1200	450	260

Кондиционер с воздушным охлаждением, состоящий из двух блоков: внутреннего блока (собственно кондиционера), в котором расположены компрессор, испаритель, вентилятор и автоматика; внешнего блока – выносного конденсатора или теплообменника. Воздух подается сверху непосредственно в помещение, а забирается через лицевую панель.

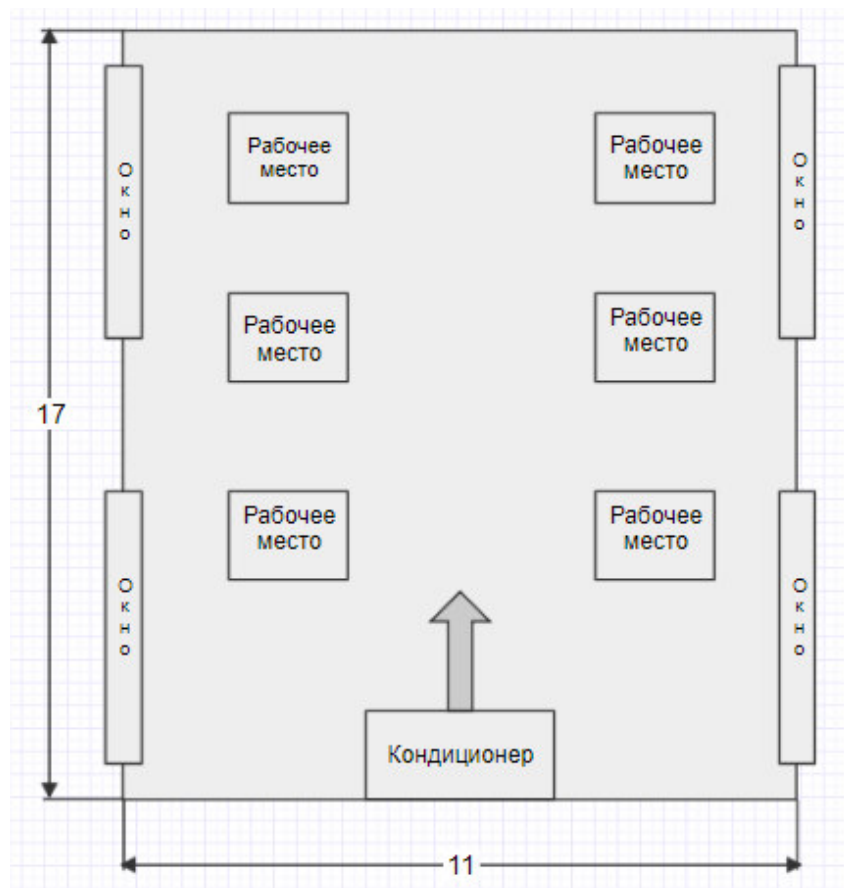


Рисунок 4.2 – Схема расположения кондиционера в производственном помещении

#### 4.6 Вывод

Проанализировав полученные данные можно сделать вывод, что текущая вентиляционная система не соответствовала требуемым нормам, в следствии чего были рассчитаны тепловые нагрузки в помещении, наружные и внутренние. По расчетам была выбрана модель кондиционера с подходящими характеристиками. Из расчетов видно, что при достаточно маленьком пространстве и большом количестве человек и оборудования, количество избыточного тепла очень высоко, что предполагает установку достаточно мощной системы кондиционирования [26].

Обеспечение воздушного комфорта в жилых и производственных помещениях зависит от систем аспирации, вентиляции, отопления и кондиционирования воздуха. Задача кондиционирования воздуха состоит в выполнении вентиляции и отопления, а также в поддержании таких параметров воздушной среды, при которых каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма чувствовал бы себя комфортно, не замечая влияния этой среды.

## Заключение

В процессе исследования был изучен вопрос пропаж и потерь сотовых телефонов, потерь и краж детей и других уязвимых слоев населения. Изучены способы борьбы с данной проблемой.

Проведен сравнительный анализ существующего программного обеспечения и выявлены их недостатки:

- необходимость модернизации алгоритмов, влияющих обширность функциональности программного обеспечения;
- необходимость расширения возможных сторонних интеграция;
- необходимость упрощения модернизации и сопровождения программного продукта.

На основании анализа актуальных проблем были поставлены задачи, которые в последующем были успешно выполнены, протестированы и введены в эксплуатацию.

Было создано веб-приложение на платформе Atom и выполнены задачи:

- реализовать функционал отслеживания устройства на дистанционном уровне;
- реализовать удаленное управление данными устройства;
- реализовать скрытность наличия ПО на устройствах;
- интегрировать систему слежения с другими существующими системами.

Проведен анализ экономических показателей веб-конвертера, в результате которого выявлено, что внедрение данного веб-приложения сокращает расходы компании, а срок окупаемости проекта занимает меньше полугода.

Рассмотрены возможные необходимые меры по защите и охране труда в виде расчета вентиляции.

Подводя итоги, следует отметить, что приложение является актуальным, так как отвечает современным требованиям рынка программного обеспечения не только в рамках Республики Казахстан, но и за его пределами.

Результат работы – приложение – легко интегрируемое, простое в использовании и сопровождении программное обеспечение, которое приносит пользу человечеству.

## Список литературы

1. Сайт <https://medium.com/@mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b>
  2. Сайт <https://www.educba.com/uses-of-django/>
  3. Сайт <https://2ip.ru/geoip/>
  4. Сайт <https://play.google.com/store/apps/details?id=org.findmykids.app>
  5. Сайт <https://www.djangoproject.com/>
  6. Сайт <https://www.datarealm.com/blog/five-advantages-disadvantages-of-mysql/>
  7. Сайт <https://quintagroup.com/cms/python/flask>
  8. Сайт <https://tech.yandex.ru/maps/mapsapi/#maps>
  9. Сайт <https://hackernoon.com/setting-up-a-python-development-envi>
  10. Сайт <https://pythonhow.com/using-atom-as-a-python-editor/>
  11. Сайт <https://atom.io/>
  12. Сайт <https://docs.python.org/3/tutorial/venv.html>
  13. James Chambers, David Paquette, Simon Timms Python Application Development: Building an application in four sprints (Developer Reference), 2016, – 70с
  14. Andrew Lock Python in Action, 2018. – 136с
  15. Michael Fitzgerald Introducing Regular Expressions, 2012. – [100-225с]
  16. Джозеф А., Бен А. Python 6.0. Карманный справочник, 2015. – 117с
  17. Mark J. Price MySQL 3 and Django: Modern Cross-Platform Development, 2016. – 216с
  18. Сайт <https://www.scnsoft.com/blog/web-application-architecture#types-of-web-app-architecture>
  19. Сайт <https://lonewolfonline.net/list-net-culture-country-codes/>
  20. Сайт <https://www.justinmind.com/blog/ui-design-principles-2018-the-new-rules/>
  21. Сайт [https://www.tutorialspoint.com/mvc\\_framework/mvc\\_framework\\_introduction.htm](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm)
  22. Сайт <https://docs.microsoft.com/ru-ru/dotnet/standard/base-types/regular-expression-language-quick-reference>
  23. Сайт <https://www.nuget.org/packages/DocumentFormat.OpenXml/>
  24. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования.
  25. Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 - Информационные системы – Алматы: АУЭС, 2013
- Расчет естественной вентиляции: Учебно-методическое пособие «Безопасность жизнедеятельности» выполнение раздела дипломных проектов /Мананбаева С.Е., Санатова Т.С., Бегимбетова А.С., Бекмуратова Н.К. – Алматы: АУЭС, 2016. – 15с