СОДЕРЖАНИЕ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

7

КИТ.032.00.00.00.ДП.ПЗ

Разраб.

Ефремов М.А.

Провер.

*Тарасов А. Ф. прпрртиВВ.В.*

Реценз.

Н. контр.

Алтухов А.В.

Утверд.

Тарасов А.Ф.

Пояснительная записка

Лит.

Листов

*X*

ИТ 12-1

[ВВЕДЕНИЕ 9](#_Toc451876375)

[1 ОБЩАЯ ЧАСТЬ 10](#_Toc451876376)

[1.1 Описание предметной области он-лайн геолокации 10](#_Toc451876377)

[1.2 Анализ существующего программного обеспечения для он-лайн геолокации 12](#_Toc451876378)

[1.2.1 Сервис «Мама, я тут!» 12](#_Toc451876379)

[1.2.2 Сервис «ТрекерПлюс» 14](#_Toc451876380)

[1.2.3 Сервис «Где Мои — NAVIXY Android Трекер» 16](#_Toc451876381)

[1.2.4 Сервис «GPShome.ru» 19](#_Toc451876382)

[1.2.5 Сервис «М2М gps-tracker.com.ua» 21](#_Toc451876383)

[1.3 Глоссарий предметной области он-лайн геолокации 25](#_Toc451876384)

[1.4 Математическая модель предметной области он-лайн геолокации 27](#_Toc451876385)

[1.5 Разработка структурно-функциональной модели бизнес-процесса онлайн-геолокации с применением SADT-технологии 28](#_Toc451876386)

[1.6 Разработка средств моделирования приложения он-лайн геолокации 31](#_Toc451876387)

[1.6.1 Разработка логической модели для приложения он-лайн геолокации 31](#_Toc451876388)

[1.6.1.1 Разработка диаграммы прецедентов использования для приложения он-лайн геолокации 31](#_Toc451876389)

[1.6.1.2 Разработка диаграммы классов для процесса он-лайн геолокации 36](#_Toc451876390)

[1.6.1.3 Разработка диаграммы последовательностей для предметной области «он-лайн геолокация» 38](#_Toc451876391)

[1.6.1.4 Разработка модели «сущность-связь» (ER-диаграммы) предметной области «он-лайн геолокация» 40](#_Toc451876392)

[1.6.2 Виды обеспечения функционирования он-лайн геолокации 41](#_Toc451876393)

[1.7 Выводы по разделу 44](#_Toc451876394)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ВЕДЕНИЯ БАЗ ДАННЫХ, ОБРАБОТКИ ДАННЫХ И ФОРМИРОВАНИЯ ОТЧЕТОВ 45](#_Toc451876395)

[2.1 Структура и функциональное назначение классов приложения для он-лайн навигации 45](#_Toc451876396)

[2.2 Детальная разработка модулей приложения для он-лайн геолокации 49](#_Toc451876397)

[ПРИЛОЖЕНИЕ А РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ НА РАЗРАБОТКУ ПРИЛОЖЕНИЯ ОН-ЛАЙН ГЕОЛОКАЦИИ 52](#_Toc451876398)

[А.1 Введение 52](#_Toc451876399)

[А.2 Основание для разработки приложения он-лайн геолокации 52](#_Toc451876400)

[А.3 Назначение разработки для приложения он-лайн геолокации 53](#_Toc451876401)

[А.4 Требования к программному продукту приложения он-лайн геолокации 53](#_Toc451876402)

[А.4.1 Требования к функциональным характеристикам для приложения он-лайн геолокации 53](#_Toc451876403)

[А.4.2 Требования к надежности приложения он-лайн геолокации 54](#_Toc451876404)

[А.4.3 Условия эксплуатации приложения он-лайн геолокации 54](#_Toc451876405)

[А.4.4 Требования к составу и параметрам технических средств приложения для он-лайн геолокации 54](#_Toc451876406)

[А.4.5 Требования к информационной и программной совместимости для приложения он-лайн геолокации 55](#_Toc451876407)

[А.5 Требования к программной документации приложения он-лайн геолокации 55](#_Toc451876408)

[А.6 Технико-экономическая эффективность приложения он-лайн геолокации 56](#_Toc451876409)

[А.7 Стадии и этапы разработки приложения он-лайн геолокации 56](#_Toc451876410)

[А.8 Порядок контроля и приемки приложения для он-лайн геолокации 57](#_Toc451876411)

# ВВЕДЕНИЕ

На рубеже XX-XXI веков глобальные спутниковые радионавигационные системы позиционирования становятся в ряд систем массового обслуживания. Спутниковая радионавигация применяется в авиации, управлении наземным и морским транспортом, картографии, мониторинге газо- и нефтепроводов, высотных сооружений, наблюдениям за смещением материков и многих других отраслях. При этом достигнуты точности определения координат от миллиметров до нескольких десятков метров. Мировое сообщество может пользоваться спутниковыми системами GPS и ГЛОНАСС безвозмездно. Сейчас спутниковая радионавигация присутствует практически в каждом мобильном телефоне.

Интересна идея использования GPS в качестве источника точного времени при проведении разного рода научных экспериментов, нельзя недооценить важность GPS и для спасательных служб.

GPS полезна для управления автомобильными системами навигации. Имея в автомобиле соответствующее оборудование, вы можете путешествовать по незнакомой местности. Введите координаты назначения, и система сама подскажет, где вам необходимо совершить поворот. GPS-оборудование, установленное на инкассаторской машине, позволит следить за ней на всем пути ее следования. Автосигнализацией на основе GPS сегодня никого не удивишь - она стала доступна многим по относительно невысокой цене. Угнанный автомобиль всегда будет "на прицеле" у диспетчерской службы. Перспективы использования системы GPS огромны. Поэтому крайне важно автоматизировать этот процесс.

[Lina.bogdanova1955@mail.ru](mailto:Lina.bogdanova1955@mail.ru)

0996043146

# 1 ОБЩАЯ ЧАСТЬ

## 1.1 Описание предметной области он-лайн геолокации

Спутниковые радионавигационные системы представляют собой всепогодные системы космического базирования и позволяют в глобальных масштабах определять текущие местоположения подвижных объектов и их скорость, а как же осуществлять точную координацию времени.

Принцип действия систем заключается в том, что навигационные спутники излучают специальные электромагнитные сигналы. Аппаратура потребителей, расположенная на объектах, находящихся на поверхности Земли или околоземном пространстве принимает эти сигналы и после специальной обработки вырабатывает данные о местоположении и скорости объекта. На рис. 1.1 представлена спутниковая радионавигационная система как высокотехнологичная информационная система, состоящая из пяти основных сегментов.

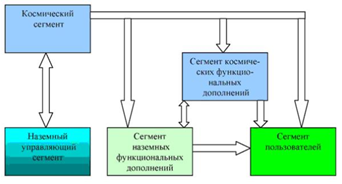


Рисунок 1.1 - Организация спутниковой радионавигационной системы

Наземный управляющий сегмент включает в себя центр управления космическим сегментом, станции слежения за навигационными спутниками (радиолокационные и оптические), аппаратуру контроля состояния навигационных спутников. Управляющий сегмент решает задачи определения, прогнозирования и уточнения параметров движения навигационных спутников, формирования и передачи в бортовую аппаратуру спутников цифровой информации, а также ряд контрольных и профилактических функций. Космический сегмент представляет собой систему навигационных спутников, вращающихся по эллиптическим орбитам вокруг Земли. На каждой орбите находятся несколько спутников. Навигационный спутник имеет на борту радиоэлектронную аппаратуру, излучающую в направлении Земли шумоподобные непрерывные радиосигналы, содержащие информацию необходимую для проведения навигационных определений с помощью аппаратуры потребителя.

Благодаря достаточному количеству навигационных спутников и специальным параметрам радиосигналов аппаратура потребителя может в любое время, при любых погодных условиях принимать излученные спутниками сигналы и определять местоположение, скорость и время.

Сегмент пользователей потенциально может состоять из неограниченного количества спутниковых навигационных приемников, которые принимают сигналы навигационных спутников и производят расчеты текущего местоположения, скорости и времени с погрешностями, определяемыми спутниковой навигационной системой и аппаратурой потребителя. Сегменты наземных и космических функциональных дополнений представляет собой аппаратурно-программные комплексы, предназначенные для обеспечения точности навигационных определений, целостности, непрерывности, доступности и эксплуатационной готовности системы.

Отдельно стоит сказать об времени. В спутниковой радионавигации время играет исключительное значение, поскольку основные навигационные определения производятся по формулам, в которых параметр времени присутствует многократно. Это прежде всего время распространения электромагнитного сигнала от навигационного спутника до потребителя, время "включения" часов спутника, время синхронизации данных передаваемых со спутника, время прохождения электромагнитного сигнала через атмосферу, влияние на время релятивистских эффектов, совмещение шкал времени спутника и потребителя и многое другое.

## 1.2 Анализ существующего программного обеспечения для он-лайн геолокации

Система обеспечения он-лайн геолокации – не тревильная задача, но не смотря на это интернет полон систем, позволяющих её осуществлять:

### 1.2.1 Сервис «Мама, я тут!»

Официальный сайт сервиса: http://mamayatut.ru/

В бесплатный тариф включен всего один трекер. На рисунке 1.2 сайт, который хоть и красочен, но к сожалению — абсолютно не информативен, и не дает практически никакой информации о том, какие настройки есть у программы, что она позволяет делать, а что нет, за сколько времени хранятся данные, и т.д.

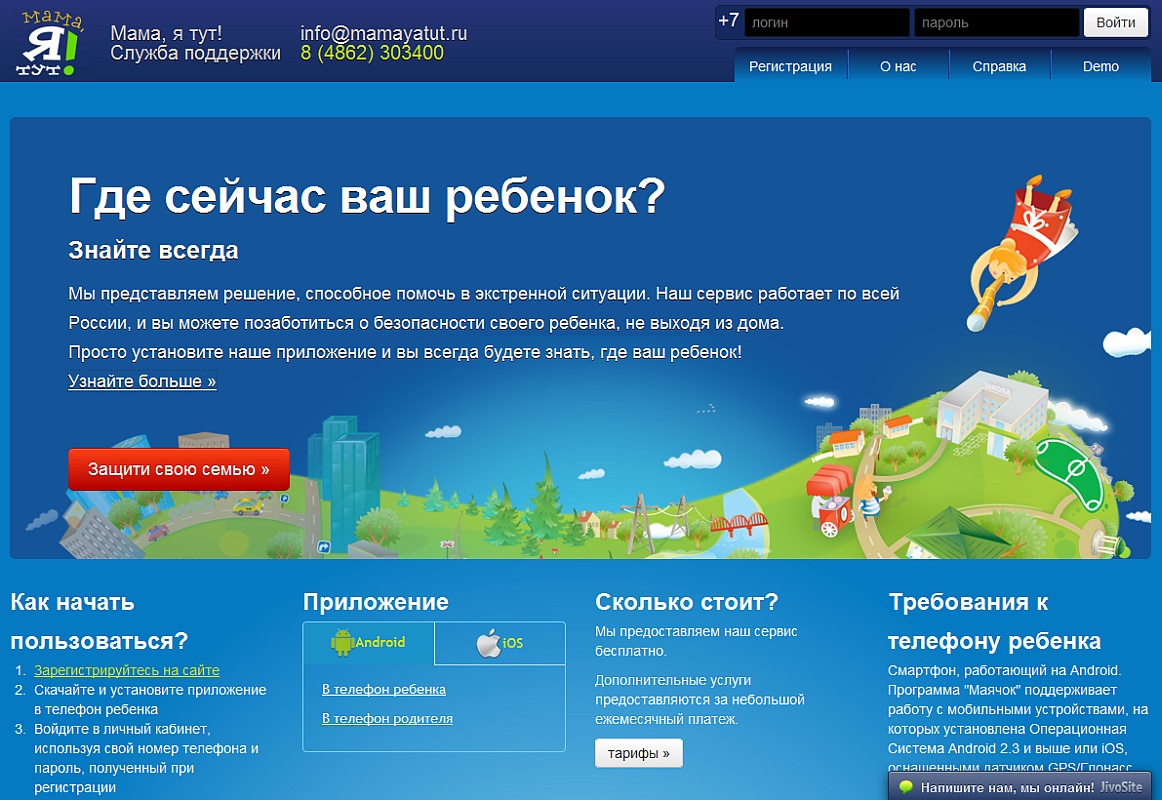


Рисунок 1.2 - Сайт системы он-лайн геолокации «Мама, я тут!»

Есть демоверсия непосредственно онлайн-сервиса карты, которая отображена на рисунке 1.3.

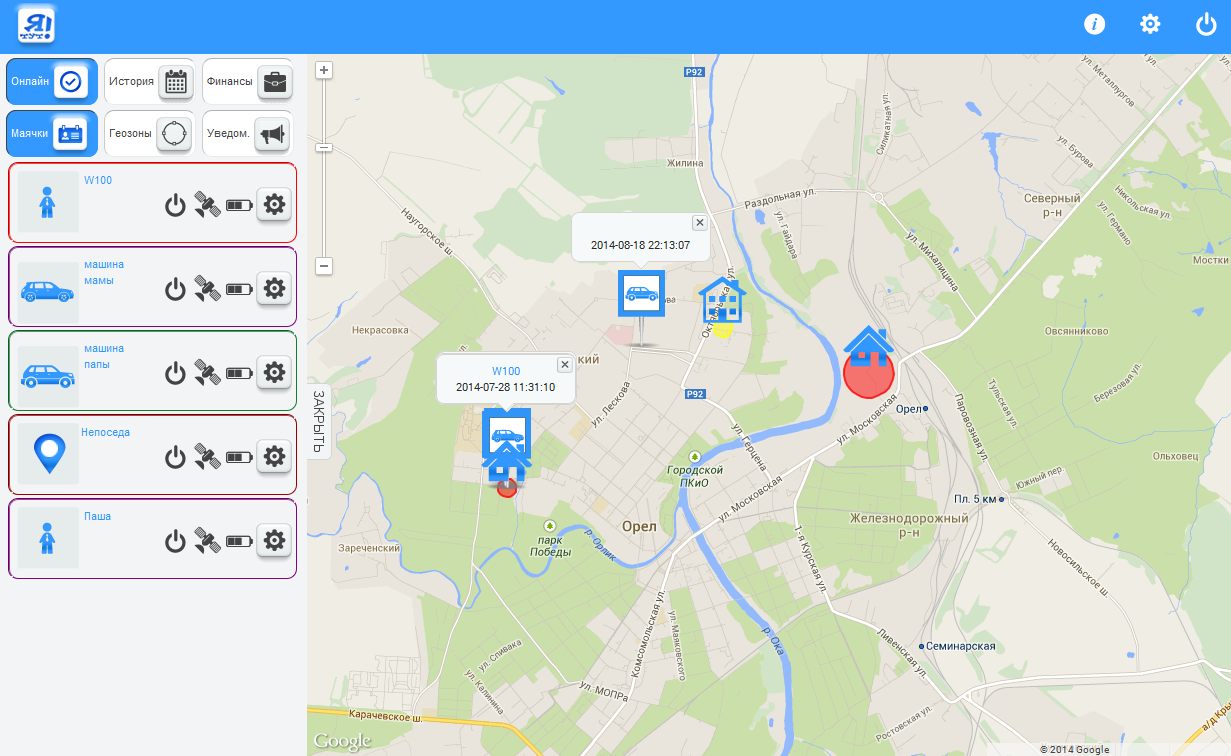


Рисунок 1.3 - Демоверcия системы он-лайн геолокации «Мама, я тут!»

В приложении можно настраивать период работы трекера, например — по будним дням, с 8 утра до 8 вечера, как часто получать координаты (в минутах), и как часто передавать данные. Эти настройки осуществляются из родительской части приложения, показанной на рисунке 1.4.

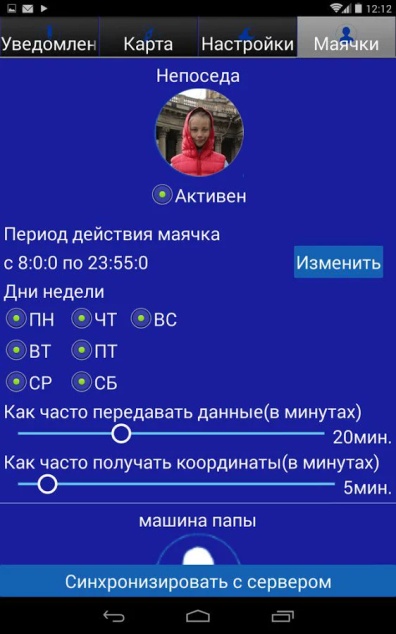


Рисунок 1.4 - Управленческая часть системы «Мама, я тут!»

Также из нее (а не только с компьютера) можно посмотреть координаты маячка, в этом случае положение отображается на Google-картах. Больше особо никаких настроек нет. Запущенный на телефоне ребенка маячок особо себя никак не проявляет, кроме символа GPS.

В ходе тестирования демоверсии периодически возникали проблемы, чувствуется, что сервис сырой. Судя по окну мониторинга история трекинга хранится всего за два дня.

### 1.2.2 Сервис «ТрекерПлюс»

Официальный сайт сервиса: http://trackerplus.ru/

Аналогично предыдущему сервису, в бесплатную версию включен только один объект. Тарифный план сервиса показан на рисунке 1.5.

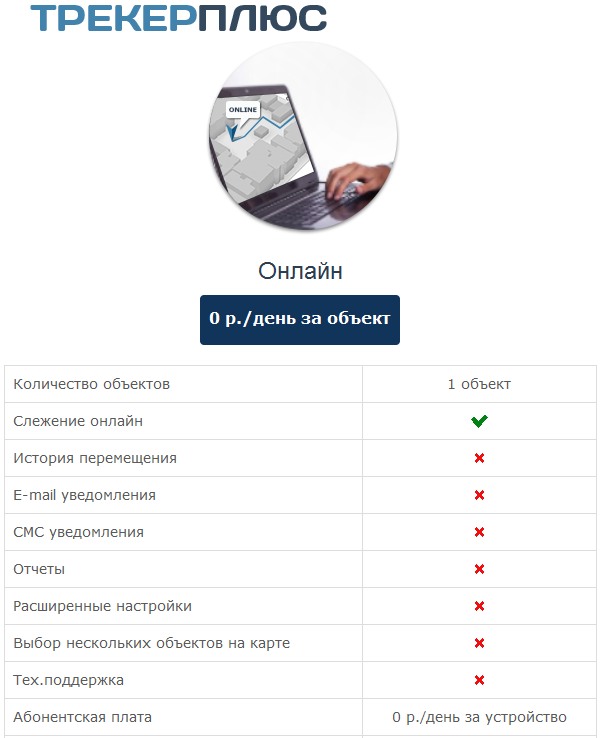


Рисунок 1.5 - Тарифные планы сервиса «Трекерплюс»

В бесплатном режиме, кроме слежения онлайн сервис ничего не может. Т.е. ни истории перемещений, ни расширенных настроек. Из настроек — можно только выбрать, через сколько секунд производить отправку координат (например, каждые 30 секунд), или каково должно быть расстояние между соседними координатами. Режим карты, показанный на рисунке 1.6 иногда подвисает: в режиме наблюдения за объектом — пытается всегда раскрыться на максимальный масштаб.

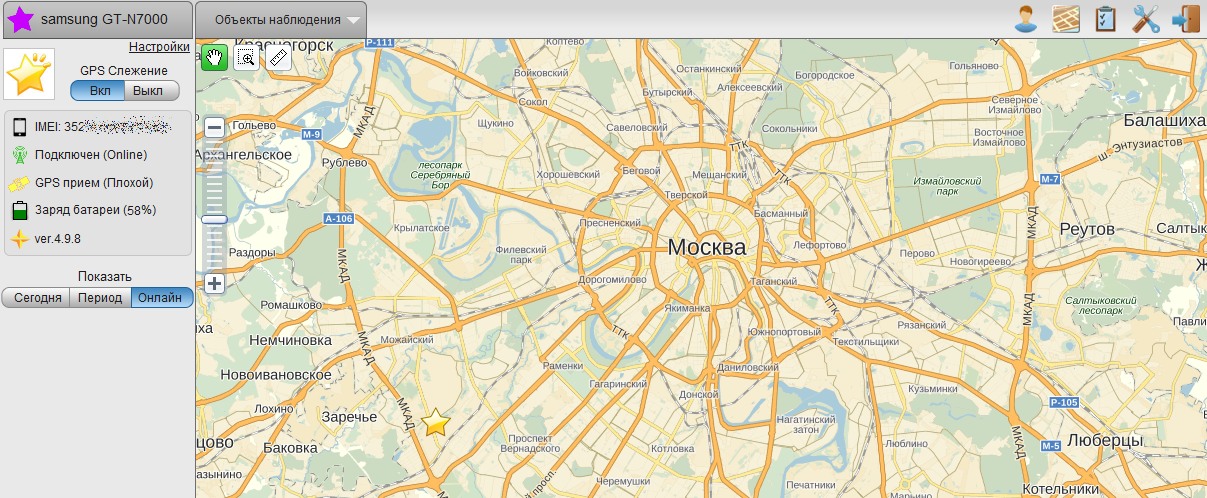


Рисунок 1.6 - Карта устройств сервиса «Трекерплюс»

В связи с описанными ограничениями данный сервис слабо подходит к серьезному использованию.

### 1.2.3 Сервис «Где Мои — NAVIXY Android Трекер»

Официальный сайт сервиса: http://trackerplus.ru/

В отличие от двух предыдущих сервисов, в бесплатной версии поддерживает уже два трекера, а не один. Может отображать местоположение не только по GPS, но и по GSM. В бесплатной версии, как и у других аналогов — сервис трекинга отсутствует, и можно только смотреть текущее местоположение. Сайт сервиса предоставлен на рисунке 1.7.



Рисунок 1.7 - Официальный сайт сервиса «Где Мои»

На рисунке 1.8 отображен demo-режим, который работает, к сожалению, только для платной версии.

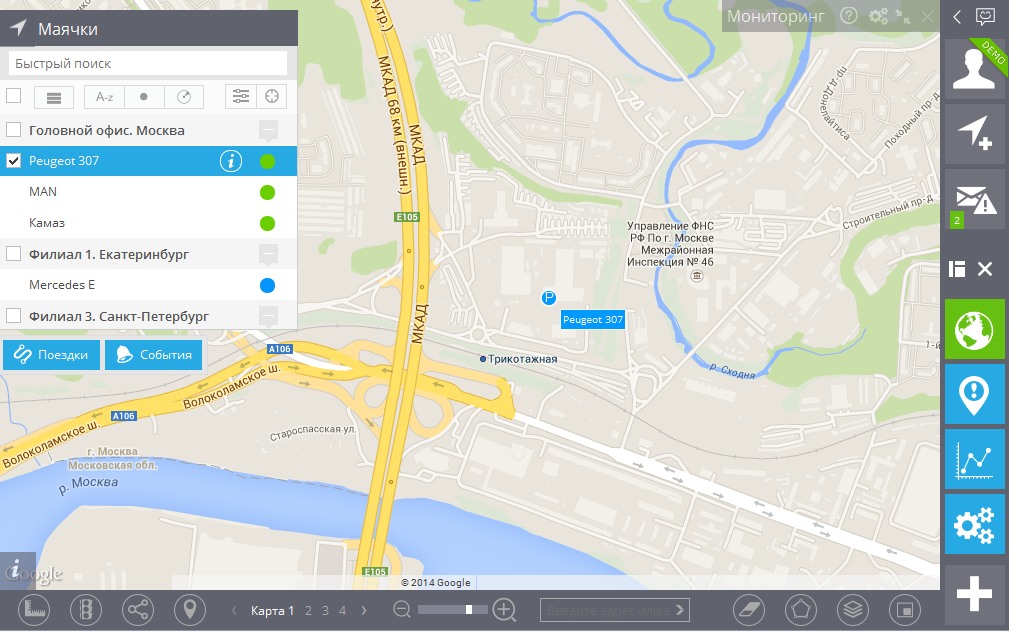


Рисунок 1.8 - Demo-режим сервиса «Где Мои»

Само мобильное приложение, показанное на рисунке 1.9, поддерживает запись точек маршрута по дистанции — через определенное количество метров, по времени — каждые несколько секунд, позволяет настроить интервал отправки данных на сервер, например, каждые 5 минут (но не делает разницы — в движении это, или нет), или поддерживать постоянное соединение с сервером. Другой вопрос, что поскольку в бесплатном виде поддерживается только показ последнего местоположения — это все бессмысленно. Также позволяет запрашивать данные GPS с определенной периодичностью. Проблема сервиса – отсутствие удалённого управления.

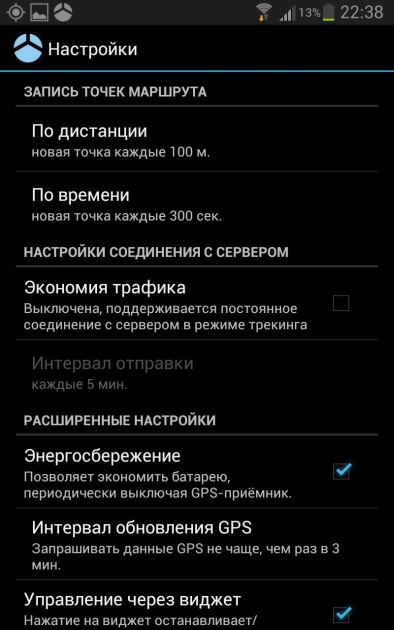


Рисунок 1.9 - Мобильная часть сервиса «Где Мои»

### 1.2.4 Сервис «GPShome.ru»

Официальный сайт сервиса: http://www.gpshome.ru/conn\_android

Продвинутый, по сравнению с аналогами сервис, позволяет отображение сразу трех объектов на карте. Сервис позволяет хранить и отображать на серверной части перемещение каждого объекта в течении 24 часов. Кроме прочего, сервис позволяет загрузить трек. Пример окна работающего сервиса есть на рисунке 1.10.

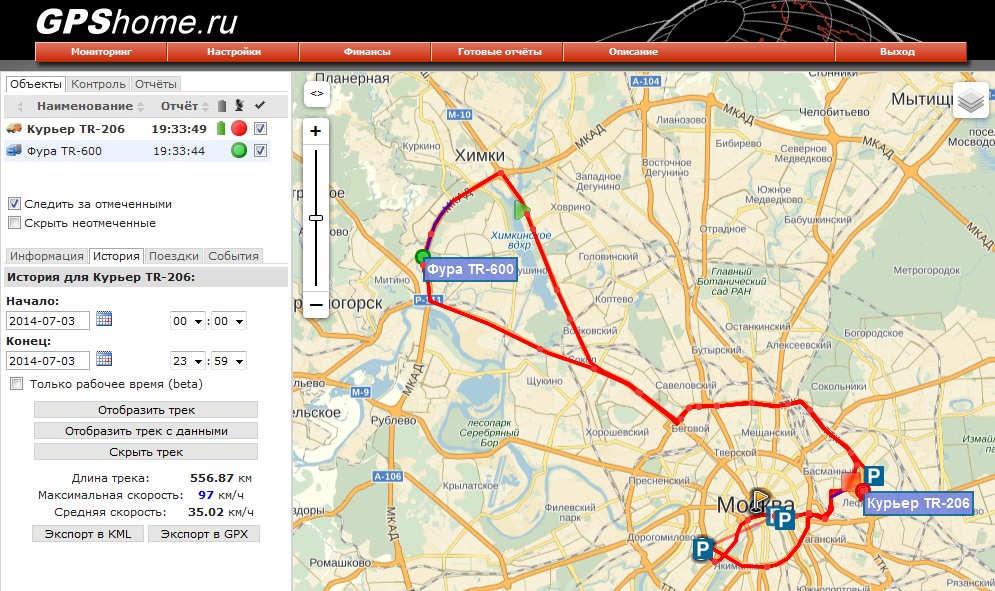


Рисунок 1.10 - Официальный сайт сервиса «GPSHome.ru»

Мобильная часть, показанная на рисунке 1.11, позволяет настроить частоту отчетов в покое, частоту отчетов в движении.

Через серверную часть настраивается минимальная скорость движения, минимальное время остановки, и радиус остановки. Стоит заметить, что в ходе тестирования сервиса немного неточно работал датчик обнаружения движения: в состоянии покоя сервис несколько раз запрашивал точку, а при движении – наоборот, иногда не реагировал.

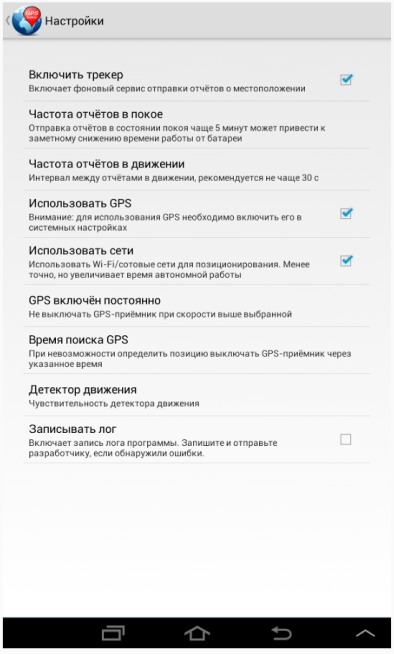


Рисунок 1.11 - Мобильная часть сервиса «GPSHome.ru»

Необходимо отметить, что данные для позиционирования определяются не только по GPS, но и по GSM — причем достаточно точно. Но только для последнего местоположения, если GPS был отключен — то трек по GSM данным построить не получится. В целом — сервис лучше предыдущих трех, особенно учитывая возможность вывода всего трека за сутки. Наблюдать — просто, трек построить — тоже просто.

### 1.2.5 Сервис «М2М gps-tracker.com.ua»

Официальный сайт сервиса: http://gps-tracker.com.ua/

Судя по описанию, можно сделать вывод, что это одна из самых мощных программ подобного типа. Официальный сайт на рисунке 1.12.



Рисунок 1.12 - Официальный сайт сервиса «M2M GPS»

Список возможностей включает опций больше, чем в предыдущих сервисах:

* до 5 объектов
* режим определения местоположения по базовым станциям при отсутствии сигнала GPS (LBS режим)
* умный режим экономии батареи (используется акселерометр для определения движения)
* выключение использования GPS сигнала при неиспользовании телефона (лежит на столе)
* режим «Автостарт», автоматический запуск службы одновременно с загрузкой телефона
* режим «Пароля администратора», запрет на остановку приложения и изменения настроек
* определение угла поворота для более точного расчета пройденного расстояния

Из недостатков можно заметить довольно сложный механизм настройки. Для добавления всего одного девайса нужно было пройти регистрацию, которая заняла около 30 минут. Да и в бесплатном режиме доступно всего одно устройство и 24 часа истории. Окно предупреждения об этом показано на рисунке 1.13.

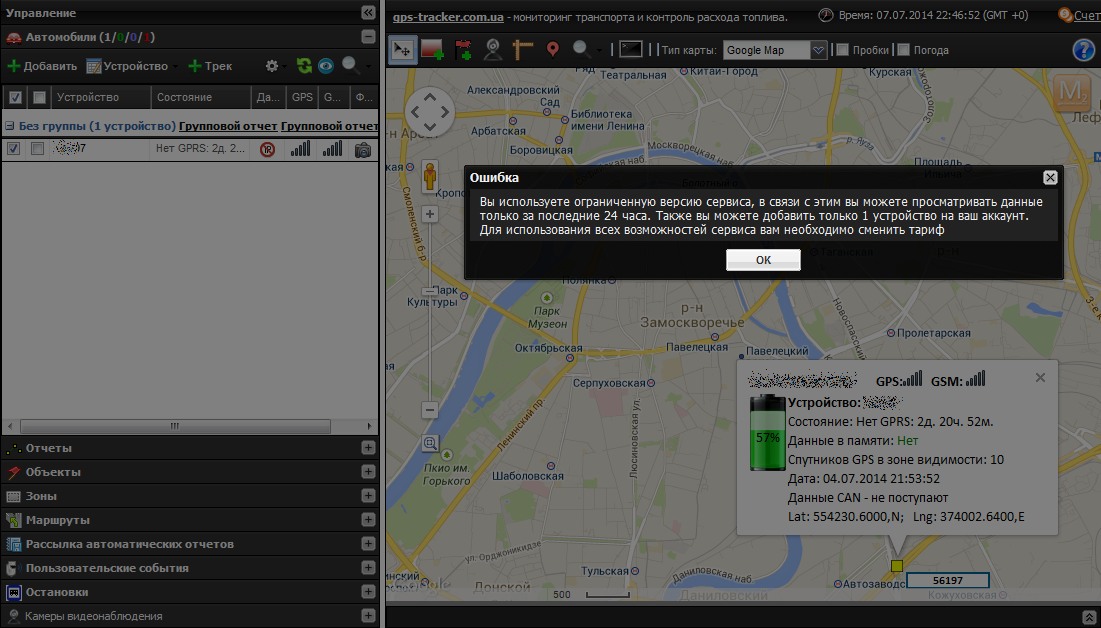


Рисунок 1.13 - Процесс настройки сервиса «M2M GPS»

В мобильной же версии история хранится явно дольше (точный период не указан). Но, к сожалению, мобильная часть иногда подтормаживает. Кроме этого клиентская часть имеет массу настроек, но все они, в принципе — не сложные для понимания. На рисунке 1.14 изображен режим настройки мобильной части.



Рисунок 1.14 - Разнообразие настроек мобильной части сервиса «M2M GPS»

Режим пароля администратора — к сожалению, работает не так как нужно. В стандартном менеджере запущенных приложений можно без особых проблем «убить» приложение. Еще из минусов — то, что нет записи точек по дистанции, нет включения в заданное время, нет возможности управлять приложением удалённо. Можно отдельно выбрать интервал времени простановки точек в зависимости от того, движется ли объект, или телефон просто лежит на столе. По батарее – сервис расходует заряд чуть активнее чем аналоги.

Еще одно очевидное преимущество – запрос точки при повороте. При плавных поворотах это не особо помогает, но в случае, если скорости большие, и повороты — резкие — отображение ближе к действительности.

Исходя из анализа аналогов, можно выделить факторы, критичные для нашей предметной области:

* Возможность управлять периодом работы;
* Количество устройств, которое поддерживает сервис;
* Возможность управлять временем запроса точки;
* Просмотр трека передвижения на клиентском приложении;
* Время хранения истории на сервисе;
* Определение угла и запрос при его детектировании;

Составим сводную таблицу (таблица 1.1) по указанным факторам для всех проанализированных нами сервисов.

Таблица 1.1 - Сводная таблица аналогов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Название | Управление периодом работы | Количество доступных устройств | Упр. врем. запроса точки | Просмотр карты из моб. прил. | Время хранения истории | Определение угла |
| Мама, я тут! | + | 1 | + | + | 48 ч. | - |
| Трекерплюс | - | 1 | + | - | 0 ч. | - |
| Где Мои | - | 2 | + | - | 0 ч. | - |
| GPSHome.ru | - | 3 | + | - | 24 ч. | - |
| M2M GPS | - | 1 | + | - | 24 ч. | + |

## 1.3 Глоссарий предметной области он-лайн геолокации

ГЕОЛОКАЦИЯ - определение реального географического местоположения электронного устройства, например, радиопередатчика, сотового телефона или компьютера, подключённого к интернету. Словом, «геолокация» может называться как процесс определения местоположения такого объекта, так и само местоположение, установленное таким способом. Часто для целей геолокации используется та или иная система позиционирования, и часто бывает важнее определить местоположение в виде, легко воспринимаемом человеком (например, почтовый адрес), нежели точные географические координаты.

СПУТНИКОВАЯ СИСТЕМА НАВИГАЦИИ - система, предназначенная для определения местоположения ([географических координат](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D0%BA%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82%D1%8B)) наземных, водных и воздушных объектов. Спутниковые системы навигации также позволяют получить скорости и направления движения [приёмника сигнала](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B2%D0%B8%D0%B3%D0%B0%D1%82%D0%BE%D1%80_(%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE)). Кроме того, могут использоваться для получения точного времени. Такие системы состоят из космического оборудования и наземного сегмента (систем управления). В настоящее время только две спутниковых системы обеспечивают полное и бесперебойное покрытие земного шара — [GPS](https://ru.wikipedia.org/wiki/GPS) и [ГЛОНАСС](https://ru.wikipedia.org/wiki/%D0%93%D0%9B%D0%9E%D0%9D%D0%90%D0%A1%D0%A1).

GPS - спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение во всемирной системе координат. Позволяет в любом месте [Земли](https://ru.wikipedia.org/wiki/%D0%97%D0%B5%D0%BC%D0%BB%D1%8F) (исключая приполярные области), почти при любой погоде, а также в околоземном космическом пространстве определять местоположение и скорость объектов.

GPS-ТРЕКЕР - устройство приёма-передачи данных для [спутникового](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D1%83%D1%82%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3) контроля [автомобилей](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D1%83%D1%82%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%B0), людей или других объектов, к которым оно прикрепляется, использующее [GPS](https://ru.wikipedia.org/wiki/GPS) для точного определения местонахождения объекта. GPS-трекер содержит [GPS-приёмник](https://ru.wikipedia.org/wiki/GPS-%D0%BF%D1%80%D0%B8%D1%91%D0%BC%D0%BD%D0%B8%D0%BA), с помощью которого он определяет свои координаты, а также передатчик на базе [GSM](https://ru.wikipedia.org/wiki/GSM), передающий данные по [GPRS](https://ru.wikipedia.org/wiki/GPRS), [SMS](https://ru.wikipedia.org/wiki/SMS) или на базе [спутниковой связи](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D1%83%D1%82%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B2%D1%8F%D0%B7%D1%8C) для отправки их на серверный центр, оснащённый специальным программным обеспечением для спутникового контроля. Кроме GPS-приёмника и передатчика важными техническими элементами трекера является GPS-антенна, которая бывает как внешняя так и встроенная в трекер; аккумуляторная батарея; встроенная память.

ТРЕКИНГ - система, предоставляющая собой некую копию систем позиционирования и ориентации, существующих в природе. «Естественные» системы трекинга в реальном мире — [органы чувств](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD_%D1%87%D1%83%D0%B2%D1%81%D1%82%D0%B2) человека. Например, зрение помогает человеку определить, где он находится относительно других предметов и людей. По сути, синоним геолокации.

ГЛОБАЛЬНАЯ НАВИГАЦИОННАЯ СПУТНИКОВА СИСТЕМА (ГЛОНА́СС) - [советская](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B2%D0%B5%D1%82%D1%81%D0%BA%D0%B8%D0%B9_%D0%A1%D0%BE%D1%8E%D0%B7)/российская [спутниковая система навигации](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D1%83%D1%82%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BD%D0%B0%D0%B2%D0%B8%D0%B3%D0%B0%D1%86%D0%B8%D0%B8), разработана по заказу [Министерства обороны СССР](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D0%B5%D1%80%D1%81%D1%82%D0%B2%D0%BE_%D0%BE%D0%B1%D0%BE%D1%80%D0%BE%D0%BD%D1%8B_%D0%A1%D0%A1%D0%A1%D0%A0). Одна из двух функционирующих на сегодня систем глобальной спутниковой навигации (китайская система спутниковой навигации [Бэйдоу](https://ru.wikipedia.org/wiki/%D0%91%D1%8D%D0%B9%D0%B4%D0%BE%D1%83) на данный момент функционирует как региональная). ГЛОНАСС предназначена для оперативного навигационно-временного обеспечения неограниченного числа пользователей наземного, морского, воздушного и космического базирования. Доступ к гражданским сигналам ГЛОНАСС в любой точке земного шара, на основании указа [Президента РФ](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B7%D0%B8%D0%B4%D0%B5%D0%BD%D1%82_%D0%A0%D0%BE%D1%81%D1%81%D0%B8%D0%B9%D1%81%D0%BA%D0%BE%D0%B9_%D0%A4%D0%B5%D0%B4%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8), предоставляется российским и иностранным потребителям на безвозмездной основе и без ограничений.

СПУТНИКОВЫЙ МОНИТОРИНГ ТРАНСПОРТА - система мониторинга [подвижных объектов](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82), построенная на основе систем [спутниковой навигации](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D1%83%D1%82%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BD%D0%B0%D0%B2%D0%B8%D0%B3%D0%B0%D1%86%D0%B8%D1%8F), оборудования и технологий [сотовой](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B2%D1%8F%D0%B7%D1%8C) и/или [радиосвязи](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B8%D0%BE%D1%81%D0%B2%D1%8F%D0%B7%D1%8C), [вычислительной техники](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и [цифровых карт](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BA%D0%B0%D1%80%D1%82%D0%B0). Спутниковый мониторинг транспорта используется для решения задач [транспортной логистики](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B3%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0) в системах управления перевозками и автоматизированных системах управления автопарком.

## 1.4 Математическая модель предметной области он-лайн геолокации

?????????????????????????????????????????????????????????????????????

## 1.5 Разработка структурно-функциональной модели бизнес-процесса онлайн-геолокации с применением SADT-технологии

Определение границ бизнес-процесса разработки программного продукта «Онлайн-геолокации» позволяет определить внешний интерфейс автоматизированной системы. Опишем границы бизнес-процесса «онлайн геолокации» при помощи спецификаций входов и выходов системы, а также условий начала и завершения этого бизнес-процесса.

На вход поступает запрос на текущую позицию.

На выходе процесса формируется траектория движения объекта на географической карта.

Эксперт по навигации обеспечивает использование gps-трекера для нахождения текущей геолокации.

Эксперт задействован на всех этапах бизнес-процесса. Он определяет с каким периодом делать запрос на текущую позицию, осуществляет непосредственно запрос и заносит полученные данные на географическую карту.

Рассмотрим подробнее работу эксперта по управлению навигацией.

Эксперту приходит запрос об определении текущей геопозиции. Эксперт, получив инструкции по навигации, анализирует необходимость запроса в данный момент времени. Если выполнения запроса требуется, эксперт его выполняет, в обратном случае отбрасывает. После выполнения проверки, эксперт должен выполнить запрос с помощью gps-трекера.

В случае, если физический доступ обеспечен, эксперт получает текущую позицию. После удачного определения текущей геопозиции эксперт должен занести текущую точку на географическую карту и провести прямую от последней известной точки, полученная траектория движения будет использоваться локально, эксперту необходимо учитывать инструкции – например, чтобы точка была адекватна в соответствии с перемещением эксперта.

Выполнив предыдущий процесс, эксперт должен отправить сообщения с текущей координатой удалённый сервис, который ведет учёт передвижения. Сообщение эксперт должен отправлять любым удобным для себя способом.

Описанный процесс можно представить в виде структурно-функциональной диаграммы (рисунок 1.15).

C:\Users\Mikhail\Downloads\SADT SA Level0 (1).png

Рисунок 1.15 – Структурно-функциональная диаграмма нулевого уровня для бизнес-процесса «Он-лайн геолокации»

Описание структурно-функциональной диаграммы нулевого уровня для бизнес-процесса «Он-лайн геолокации» приведено в таблице 1.2.

Таблица 1.2 – Описание структурно-функциональной диаграммы нулевого уровня для бизнес-процесса «Он-лайн геолокации»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Входные данные | Управление | Исполнитель | Выходные  Данные |
| А0 | Запрос текущей позиции | Физический доступ к спутникам  Инструкции по навигации | Географическая карта  Gps-трекер  Эксперт по управлению навигацией  Удалённый сервис навигации | Карта для личного использования  Траектория движения всех зарегистрированных экспертов |

На структурно-функциональной диаграмме первого уровня (рисунок 1.16) приведены основные активности системы, детализирующие основной бизнес-процесс «Он-лайн геолокации».

Процесс 1 «Обработать и проверить необходимость запроса», происходит после поступления запроса позиции. На выходе получаем проверенный запрос.

Процесс 2 «Запрос текущей позиции» выполняется после проверки запроса на корректность. Зависит от физического доступа к gps-спутникам. После удачно выполненного запроса, мы получает точку с текущей позицией.

Процесс 3 «Отметка позиции на карте» выполняется после удачного запроса и получения текущей точки. Выход распараллеливается. Один выход является траекторией на карте для личного пользования. Второй выход направляет сообщение с координатой удалённому сервису навигации.

Процесс 4 «Отметка позиции на карте от данного эксперта» выполняется после получения сообщения с координатой от эксперта. Выводом является траектория движения для каждого эксперта в отдельности.

В таблице 3 приведено описание детализирующей структурно-функциональной диаграммы первого уровня бизнес-процесса «Он-лайн геолокации» с указанием входов и выходов каждого вида работ, условий их выполнения и исполнителей.

Таблица 1.3 – Описание детализирующей структурно-функциональной диаграммы первого уровня бизнес-процесса «Он-лайн геолокации»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование операции (активности) | Управление активностью | | Входы  (документы, данные и др.) | Выходы  (документы, данные и др.) | Ответственный за операцию |
| При каких условиях начинается | Чем регламентируется и завершается |
| Обработать и проверить необходимость запроса | Запрос позиции | Проверена необходимость запроса | Инструкция по навигации | Запрос позиции | Эксперт по управлению навигацией |
| Запрос текущей позиции | Запрос прошел проверку | Запрос выполнен успешно | Запрос позиции | Текущая координатная точка | Gps-трекер  Эксперт по управлению навигацией |
| Отметка позиции на карте | Успешно выполнен запрос позиции | Точка отмечена на карте | Текущая геопозиции | Траектория на личной карте  Сообщение с координатой | Эксперт по управлению навигацией  Географическая карта |

Продолжение таблицы 1.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Отметка позиции на карте от эксперта | При получении сообщения с координатой | Точка отмечена на карте | Сообщение с координатой | Траектория движения всех  экспертов | Географическая карта  Удалённый сервис |

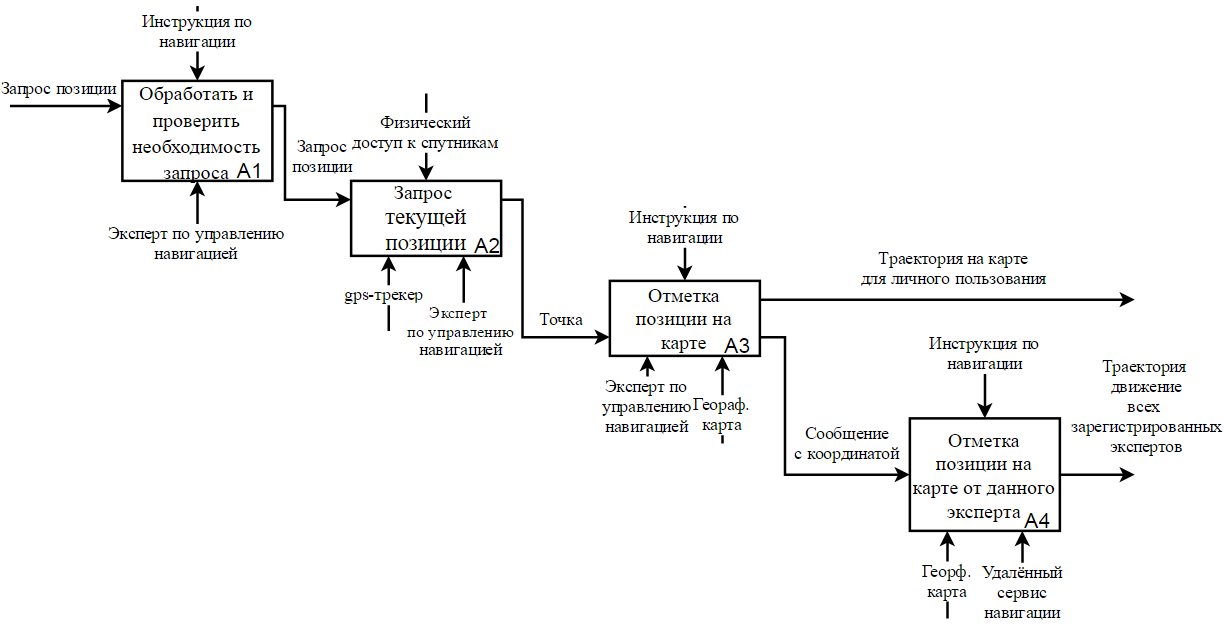


Рисунок 1.16 – Детализирующая структурно-функциональная диаграмма первого уровня бизнес-процесса «Он-лайн геолокации

## 1.6 Разработка средств моделирования приложения он-лайн геолокации

### 1.6.1 Разработка логической модели для приложения он-лайн геолокации

#### 1.6.1.1 Разработка диаграммы прецедентов использования для приложения он-лайн геолокации

Диаграмма прецедентов (Use case diagram) – диаграмма, на которой показано множество прецедентов и актеров, а также отношений между ними [4].

Конструкция или стандартный элемент языка UML вариант использования применяется для спецификации общих особенностей поведения системы или любой другой сущности предметной области без рассмотрения внутренней структуры сущности. Каждый вариант использования определяет последовательность действий, которые должны быть выполнены проектируемой системой при взаимодействии ее с соответствующим актером. Диаграмма вариантов может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов [4].

Диаграмма вариантов (прецедентов) эксперта по управлению навигацией приведена на рисунке 1.17.

Эксперт должен обеспечивать процесс он-лайн навигацию. В процесс входят проверка запросов позиции в соответствии с предписанными инструкциями. Эксперт делает непосредственно запрос на получение текущей геопозиции.

Эксперт должен заносить полученные геоданные на географическую карту. Этот процесс так же включает отсылку данных об геопозиции удалённому сервису, для обеспечения удалённой навигации.

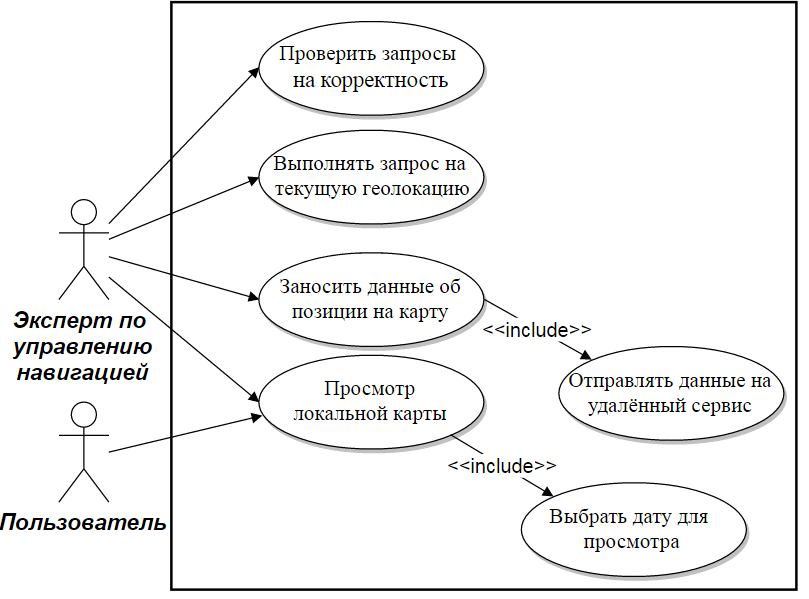
Кроме этого, эксперт по управлению навигацией может в любой момент просмотреть локальную карту для просмотра свой траектории движения.

Для решения поставленных задач выделим такие прецеденты: Проверить запросы корректность, Выполнять запрос на текущую геопозицию, Заносить данные об позиции на карту, Отправлять данные на удалённый сервис, Просматривать локальную карту.

Так же имеется актер пользователь, который при наличии прав может просматривать карту, составленную экспертом.

На основании описанных выше прецедентов использования выполним построение диаграммы прецедентов (рисунок 1.17).

Таким образом, на разработанной диаграмме вариантов использования показаны 5 прецедентов, а также отношения между ними.

 Рисунок 1.17 – Диаграмма вариантов (прецедентов) «он-лайн геолокация»

Выполним описание прецедента «Проверить запросы на корректность» (таблица 1.4).

Таблица 1.4 – Описание прецедента «Проверить запросы на корректность»

|  |
| --- |
| Описание прецедента |
| Прецедент – Эксперт по управлению навигацией |
| Основной исполнитель – *Эксперт по управлению навигацией*  Заинтересованные лица –  *Эксперт по управлению навигацией* |
| Предусловия:   * Поступает новый запрос на определение геопозиции |
| Входные данные:   * Запрос на геопозицию |
| Основной успешный сценарий:   * Запрос прошел проверку |

Продолжение таблицы 1.4

|  |
| --- |
| Описание прецедента |
| Частота выполнения:   * Каждый раз, когда выполняется запрос на геопозицию |
| Постусловия (результаты):   * Запрос прошел проверку |
| Выходные данные:   * Запрос, готовый выполнятся |

Выполним описание прецедента «Выполнять запрос на текущую геопозицию» (таблица 1.5).

Таблица 1.5 – Описание прецедента «Выполнять запрос на текущую геопозицию»

|  |
| --- |
| Описание прецедента |
| Прецедент – Выполнять запрос на текущую геопозицию |
| Основной исполнитель – *gps-трекер*  Заинтересованные лица – *Эксперт по управлению навигацией* |
| Предусловия:   * Проверенный запрос, который необходимо выполнить |
| Входные данные:   * Запрос на выполнение |
| Основной успешный сценарий:   * Gps-трекер получает текущую координату * Эксперт получает результаты определения геопозиции |
| Частота выполнения:   * Каждый раз при успешной проверки входящего запроса |
| Постусловия (результаты):   * Позиция полученны |
| Выходные данные:   * Сохраненные геоданные |

Выполним описание прецедента «Заносить данные об позиции на карту» (таблица 1.6).

Таблица 1.6 – Описание прецедента «Заносить данные об позиции на карту»

|  |
| --- |
| Описание прецедента |
| Прецедент – Заносить данные об позиции на карту |
| Основной исполнитель – *Эксперт по управлению навигацией*  Заинтересованные лица – *Эксперт по управлению навигацией* |
| Предусловия:   * Была получена корректная геопозиция |
| Входные данные:   * Геопозиция эксперта |
| Основной успешный сценарий:   * Эксперт заносит данные на карту |
| Частота выполнения:   * Каждый раз, когда приходит корректные геоданные от трекера |
| Постусловия (результаты):   * Геоданные занесены на карту |
| Выходные данные:   * Географическая карта |

Выполним описание прецедента «Отправлять данные на удалённый сервис» (таблица 1.7).

Таблица 1.7 – Описание прецедента «Отправлять данные на удалённый сервис»

|  |
| --- |
| Описание прецедента |
| Прецедент – Отправлять данные на удалённый сервис |
| Основной исполнитель – *Эксперт по управлению навигацией*  Заинтересованные лица – *Удалённый навигационный сервис* |
| Предусловия:   * Эксперт занес на карту текущую геопозицию |
| Входные данные:   * Геопозиция эксперта |

Продолжение таблицы 1.7

|  |
| --- |
| Описание прецедента |
| Основной успешный сценарий:   * Эксперт отправляет геоданные на сервис |
| Частота выполнения:   * Каждый раз, когда приходит корректные геоданные от трекера |
| Постусловия (результаты):   * Геоданные занесены сервису |
| Выходные данные:   * Географическая карта с траекториями передвижений всех экспертов |

Выполним описание прецедента «Просмотр локальной карты» (таблица 1.8).

Таблица 1.8 – Описание прецедента «Просмотр локальной карты»

|  |
| --- |
| Описание прецедента |
| Прецедент – Просмотр локальной карты |
| Основной исполнитель – *Эксперт по управлению навигацией*  Заинтересованные лица – *Пользователь* |
| Предусловия:   * Актер открыл карту |
| Входные данные:   * Запрос эксперта |
| Основной успешный сценарий:   * Актер получил карту с историей трекинга |
| Частота выполнения:   * Каждый раз, когда актер сделает запрос |
| Постусловия (результаты):   * Актер получил карту |
| Выходные данные:   * Географическая карта с историей трекинга |

#### 1.6.1.2 Разработка диаграммы классов для процесса он-лайн геолокации

Диаграмма классов – это UML-диаграмма, которая предназначена для отображения статической структуры модели системы [4].

Центральное место в объектно-ориентированном анализе и проектировании занимает разработка логической модели системы в виде диаграммы классов.

Класс (class) в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов [4]. Диаграмма классов отражает статический аспект программы. Большинство существительных в описании предметной области являются претендентом на классы. Также претендентов на классы можно найти в описании прецедентов.

Диаграммы классов обычно содержат следующие сущности:

* классы;
* интерфейсы;
* кооперации;
* отношения зависимости, обобщения и ассоциации.

Диаграммы классов применяют для моделирования статического вида системы с точки зрения проектирования. В этом представлении удобнее всего описывать функциональные требования к системе – услуги, которые она предоставляет конечному пользователю [4].

На основе описания предметной области и описаний прецедентов в рассматриваемой предметной области можно выделить следующие классы:

* Эксперт
* Инструкция геолокации
* Локальная карта
* Геоточка
* Gps-трекер
* Сервис мессенджер

Для процесса он-лайн геолокации разработана диаграмма классов (рисунок 1.18).



Рисунок 1.18 – Диаграмма классов для «он-лайн геолокации»

#### 1.6.1.3 Разработка диаграммы последовательностей для предметной области «он-лайн геолокация»

Диаграмма последовательности – это UML-диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления. Ее недостатком является то, что она занимает слишком много места [4].

Диаграмма, на которой изображено упорядоченное во времени взаимодействие объектов. В частности, на ней изображаются участвующие во взаимодействии объекты и последовательность сообщений, которыми они обмениваются [4].

Все объекты на диаграмме последовательности образуют некоторый порядок, определяемый очередностью или степенью активности объектов при взаимодействии друг с другом.

Эксперт отправляет на проверку сообщение с запросом, после ответа он выполняет запрос на позицию, получив ответ он отрисовывает на карте полученные геоточки. Отрисовав точки на карте эксперт с помощью Сервис Мессенджера отправляет свои геоданные на сервис. На рисунке 18 показана диаграмма последовательности для прецедента «он-лайн геолокация».

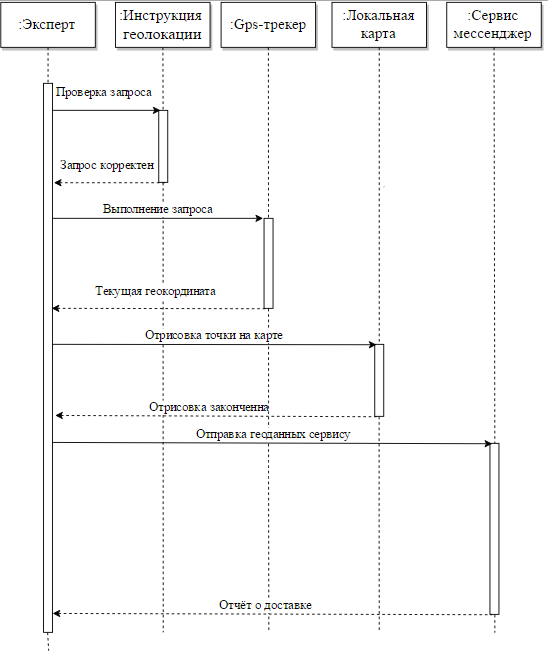


Рисунок 1.19 – Диаграмма последовательности для прецедента «он-лайн геолокация»

#### 1.6.1.4 Разработка модели «сущность-связь» (ER-диаграммы) предметной области «он-лайн геолокация»

Модель сущность-связь (ER-модель) — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-диаграммы используются для разработки и определения данных, а также для установления отношений между ними. В ходе анализа предметной области выделяется набор сущностей и связей между ними. Выполним построение информационной модели он-лайн геолокации в виде ER-диаграммы (рисунок 19).

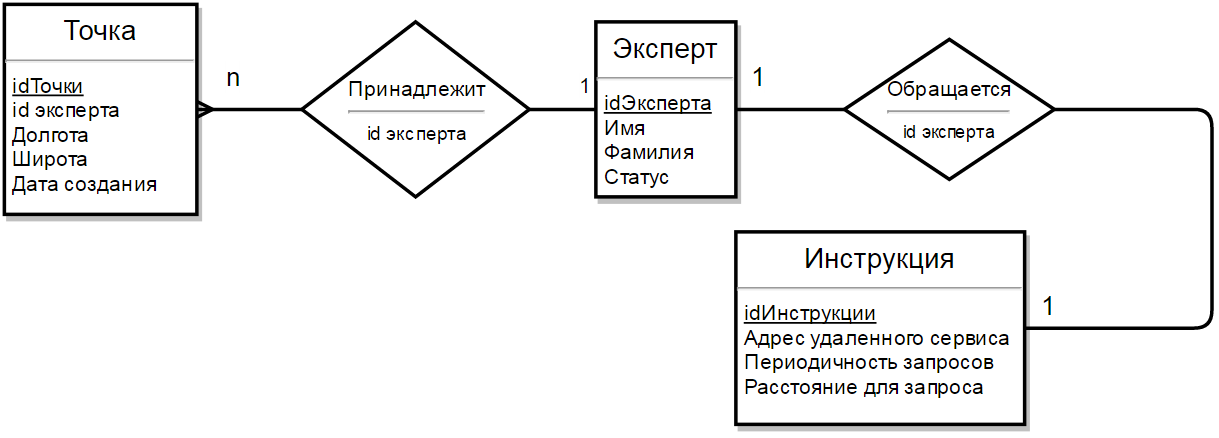


Рисунок 1.20 – ER-диаграмма процесса он-лайн геолокации

В ходе анализа предметной области были выделены основные сущности: «Точка», «Эксперт», «Инструкции», а также связи: «Принадлежит», «Обращается». Всего 3 сущности и 2 двухсторонние связи.

Сущность «Точка» хранит в себе информацию об id эксперта, у которого хранятся точки, о широте, долготе и дате создания точки. В качестве первичного ключа в этой сущности введен атрибут «idТочки», который представляет собой уникальный номер.

Сущность «Эксперт» хранит в себе данные имени и фамилии, а так же статус эксперта. В качестве первичного ключа в этой сущности введен атрибут «idЭксперта», который представляет собой уникальный номер.

Сущность «Инструкции» хранит информацию о адресе удалённого сервиса, на который отправляются данные, а также периодичность и расстояние для выполнения запросов. Первичным ключом является поле «idИнструкции».

Исходя из имеющихся данных, мы можем определить ключевые понятия предметной области, для создания модели сущность-связь.

R1 (Точка) = (idТочки, idЭксперта, Долгота, Широта, Дата создания).

R2 (Принадлежит) = (idЭксперта).

R3 (Эксперт) = (idЭкперта, Имя, Фамилия, Статус).

R4 (Обращается) = (idЭксперта).

R5 (Инструкция) = (idИнструкции, Адрес удалённого сервиса, Периодичность запроса, Расстояния запроса).

### 1.6.2 Виды обеспечения функционирования он-лайн геолокации

Для реализации приложения онлайн геолокации мы будем использовать такие программные средства: .NET C#, фреймворк Xamarin и базы данных SQlite и MySql.

Тема дипломного проекта – «Проект программного комплекса для он-лайн геолокации». Проект будет реализовываться в среде разработки – Microsoft Visual Studio.

Сравнение программных продуктов (по десятибалльной шкале) приведено в таблице 1.9.

Таблица 1.9 – Сравнение программных продуктов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование критерия | C# | C++ | Delphi | Kj |
| Поддержка мультиплатформенности | 8 | 6 | 4 | 0,20 |
| Поддержка баз данных | 8 | 7 | 7 | 0,25 |
| Скорость выполнения программ | 7 | 8 | 4 | 0,10 |
| Поддержка объектно-ориентированных средств | 10 | 6 | 8 | 0,10 |
| Наличие дополнительных компонентов | 10 | 8 | 4 | 0,10 |
| Простота применения | 9 | 5 | 7 | 0,10 |
| Устойчивость к нагрузкам | 8 | 8 | 4 | 0,15 |

Экспертная оценка определяется по формуле (1.1).

, (1.1)

где Qi – общая оценка программного продукта;

Kj – весовой коэффициент критерия;

aij – оценка i – того программного средства по j – тому критерию.

Рассчитаем общие оценки программных средств:

C#:

Q=8\*0,20+8\*0,25+7\*0,10+10\*0,10+10\*0,10+9\*0,10+8\*0,15=8,4.

C++ 6.0:

Q=6\*0,20+7\*0,25+8\*0,10+6\*0,10+8\*0,10+5\*0,10+8\*0,15=6,85.

Delphi 7:

Q=4\*0,20+7\*0,25+4\*0,10+8\*0,10+4\*0,10+7\*0,10+4\*0,15=5,45.

Как показал расчет, наиболее целесообразно выбирать в качестве средства разработки язык программирования C#:

Сравнение СУБД (по десятибалльной шкале) приведено в таблице 1.10.

Таблица 1.10 – Сравнение СУБД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование критерия | BerkleyDB | LevelDB | SQLite | Kj |
| Open Source | 8 | 8 | 8 | 0,10 |
| Мультиплатформенность | 6 | 6 | 10 | 0,20 |
| Мобильность | 8 | 7 | 8 | 0,15 |
| Поддержка реляционной модели | 7 | 1 | 10 | 0,10 |
| Поддержка стандарта SQL | 5 | 9 | 9 | 0,15 |
| Средства для редактирования | 7 | 8 | 8 | 0,10 |
| ORM | 5 | 9 | 8 | 0,20 |

Экспертная оценка определяется по формуле (1.1) для каждой СУБД:

BerkleyDB:

Q=8\*0,10+6\*0,20+8\*0,15+7\*0,10+5\*0,15+7\*0,10+5\*0,20=6,35.

LevelDB:

Q=8\*0,10+6\*0,20+7\*0,15+1\*0,10+9\*0,15+8\*0,10+9\*0,20=7,10.

SQLite:

Q=8\*0,10+10\*0,20+8\*0,15+10\*0,10+9\*0,15+8\*0,10+8\*0,20=8,75.

Как показал расчет, наиболее целесообразно выбирать в качестве СУБД SQLite

## 1.7 Выводы по разделу

В первом разделе дипломной работы был проведен анализ предметной области «он-лайн геолокация». Проанализирована работа эксперта по работе с навигацией, рассмотрены основные понятия в данной предметной области. Разработаны SADT-диаграммы и диаграммы UML: диаграмма прецедентов, диаграмма классов, диаграмма последовательности и ER-диаграмма.

# 2 СПЕЦИАЛЬНАЯ ЧАСТЬ. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ВЕДЕНИЯ БАЗ ДАННЫХ, ОБРАБОТКИ ДАННЫХ И ФОРМИРОВАНИЯ ОТЧЕТОВ

## 2.1 Структура и функциональное назначение классов приложения для он-лайн навигации

Программное обеспечение для он-лайн навигации включает в себя следующие основные классы [6]:

MainActivity – класс, который управляет работой сервисной части приложения, так же из этого класса, который является главным интерфейсом, можно вызвать окно локальной карты.

LocationService – класс-сервис, который обеспечивает фоновую работу приложения и который вызывает класс, управляющий логикой приложения.

LogicManager – класс, вызываемый сервисной частью приложения, имеет в себе всю логическу часть работы приложения.

Point – класс, выполняющий роль вспомогательного, хранит в себе данные о точке.

UdpServer – класс, отвечающий за отправку и синхронизацию данных между клиентом и сервисной частью.

Configuration – класс, выполняющий роль хранилища для актуальных настроек приложения (минимальный таймаут между запросами, минимальная дистанция).

SensorListener – класс, выполняющий роль детектора срабатывания указанного сенсора, нужного для логики класса LogicManager.

LocationListener – класс, имплементирующий интерфейс, который может выполнять запросы к gps-спутникам и слушать их ответ.

Выполним описание некоторых основных методов, которые находятся в используемых классах (таблицы 2.1 – 2.8).

Таблица 2.1 – Описание методов класса MainActivity

|  |  |
| --- | --- |
| Метод | Описание |
| protected override void OnCreate(Bundle bundle) | Метод, вызываемый при старте приложения. Инициализирует класс. |
| public void Subscribe() | Метод, вызываемый для подписки класса MainActivity с сервисной частью приложения. |
| private void DisplayAddress(Address address) | Метод, реагирующий на событие смены адреса. Выводит новый адрес на экран. |

Таблица 2.2 – Описание методов класса LocationService

|  |  |
| --- | --- |
| Метод | Описание |
| public override StartCommandResult OnStartCommand(Intent intent, StartCommandFlags flags, int startId) | Метод, вызываемый при старте сервиса. Возвращает логику перезапуска в случае ошибки. |
| public override void OnDestroy() | Метод, вызываемый при закрытии сервисной части приложения. |
| public void SendToast(string message) | Метод, выполняющий отправку сообщений пользователю. |

Продолжение таблицы 2.3 – Описание методов класса LogicManager

|  |  |
| --- | --- |
| Метод | Описание |
| public void ForceRequestLocation(LocationManager locationManager) | Метод, предоставляющий интерфейс вызова позиции из вне. |
| private void OnSensorChangedEvent() | Метод-событие, срабатывающее при изменении статуса сенсора. |
| private void GetLocation(LocationCallReason reason) | Основной метод, выполняющий вызов gps-модуля для определения текущей геопозиции. |
| public void OnLocationChanged(Location location) | Метод, вызываемый при смене позиции. |
| private void SaveInBase(Point point) | Метод, который вызывает метод сохранения в базе для точки. |
| private void SendToServer(Point point) | Метод, отправляющий точку udp-серверу для отправки на сервис. |

Таблица 2.4 – Описание методов класса Point

|  |  |
| --- | --- |
| Метод | Описание |
| public string GetMessageToSend() | Метод, возвращающий сообщение, готовое для отправки на сервис. |
| public int SaveInBase() | Метод, выполняющий сохранение точки в базе данных. |
| public static int GetGreatestAck() | Метод, выполняемый для получения максимального значения идентификатора, нужного для синхронизации данных. |

Таблица 2.5 – Описание методов класса UdpServer

|  |  |
| --- | --- |
| Метод | Описание |
| public void Add(Point point) | Метод, необходимый для добавления сообщения в очередь на отправку. |
| public void Ack(Point point) | Метод, вызываемый после получения отчёта о доставке от сервиса. |
| private void OnTimerCall(object state) | Метод, вызываемый по таймеру. Служит для проверки очереди. |
| private int Send(string message) | Метод, оправляющий сообщение на удалённый сервис. |
| private void Listen() | Метод, вызываемый для прослушки порта на входящие сообщения. |
| public void Stop() | Метод, вызываемый при попытке остановить прослушку. |

Таблица 2.6 – Описание методов класса Configuration

|  |  |
| --- | --- |
| Метод | Описание |
| public Configuration(long minTime, int minDistance) | Конструктор, выполняющий синхронизацию настроек с базой. |

Таблица 2.7 – Описание методов класса SensorListener

|  |  |
| --- | --- |
| Метод | Описание |
| public void OnSensorChanged(SensorEvent e) | Метод-событие, вызываемый при смене статуса сенсора. |

Таблица 2.8 – Описание методов класса LocationListener

|  |  |
| --- | --- |
| Метод | Описание |
| private void Initialize() | Метод, выполняющий инициализацию класса. |
| public void RequestLocation(long minTime, int minDistance) | Метод, который периодически отправляет запрос к gps-модулю. |
| public void SingleRequestLocation() | Метод, выполняющий единичный запрос на позицию к gps-модулю. |
| public void OnLocationChanged(Location location) | Метод, вызываемый при смене получении повой геопозициию |

## 2.2 Детальная разработка модулей приложения для он-лайн геолокации

Рассмотрим более детально реализацию основных методов приложения для он-лайн геолокации.

Класс MainActivity – класс, выполняющий роль графического интерфейса. Код OnCreate метода приведен на рисунке 2.1:

protected override void OnCreate(Bundle bundle)

{

base.OnCreate(bundle); //вызов базовой имплементации OnCreate

SetContentView(Resource.Layout.Main); //Установка главного графического окна

\_addressText = FindViewById<TextView>(Resource.Id.address\_text);

\_locationText = FindViewById<TextView>(Resource.Id.location\_text);

\_errorText = FindViewById<TextView>(Resource.Id.textErrorInfo);

//определение текстовых переменных главного графического окна

SubscribeUi(); //метод, для подписки событий окна с методами класса

}

Рисунок 2.1 – Метод OnCreate класса MainActivity

private async void OnLocationChanged(Location location)

{

\_counter++; //счетчик для мониторинга

\_currentLocation = location; //присвоение внутренней переменной класса

\_locationText.Text = \_counter + "Lat:" + \_currentLocation.Latitude + " Lon:" + \_currentLocation.Longitude; //изменение полученными данными строки

var address = await ReverseGeocodeCurrentLocation(); //геокодирование полученного адресса

DisplayAddress(address); //отображение полученного адресса

}

Рисунок 2.2 – Метод OnLocationChanged класса MainActivity

Для каждых того, чтобы срабатывал метод OnLocationChanged (рисунок 2.2) нам необходимо запустить сервисную часть приложени и подписаться на её события, что и показано на рисунке 2.3:

public void Subscribe()

{

if (!\_binder.GetDemoService().IsStarted) //проверка статуса сервиса

\_binder.GetDemoService().StartService(new Intent(this, typeof(LocationService))); //старт сервисной части приложения

if (!\_isBinding)

{

\_isBinding = true;

\_binder.GetService().LogicManager.OnLocationChangedEvent += OnLocationChanged; //подписывание на событие OnLocationChangedEvent

\_binder.GetService().OnError += OnError; //подписывание на OnError

}

}

Рисунок 2.3 – Метод Subscribe класса MainActivity

После запуска сервисной части выполняется метод OnStartCommand, который показан на рисунке 2.4:

[Obsolete("deprecated")]

public override StartCommandResult OnStartCommand(Intent intent, StartCommandFlags flags, int startId)

{

IsStarted = true;

var builder = new Notification.Builder(this) //создание нотификации сервиса

.SetContentTitle("Personal Tracker")

.SetContentText("Service is working. Coming soon to click event!")

.SetSmallIcon(Resource.Drawable.Icon)

.SetLargeIcon(BitmapFactory.DecodeResource(Resources, Resource.Drawable.IconBlack));

var notification = builder.Build();

StartForeground(startId, notification); //запуск сервиса с «вечной» нотификацей

SendToast("Service was started"); //отправка сообщения об удаче

Log.Debug(Tag, "OnStartCommand called at {2}, flags={0}, startid={1}", flags, startId, DateTime.UtcNow); //запись старта в дебаг-журнал

return StartCommandResult.Sticky; //возвращение логики поведения сервиса

}

Рисунок 2.4 – Метод OnStartCommand класса LocationService

После запуска сервиса, приложению необходимо инициализировать класс, выполняющий логику приложения, что и показано на рисунке 2.5:

public void Initialize()

{

LogicManager = new LogicManager(UniqueId, (LocationManager)GetSystemService(LocationService)); //инициализация логики

LogicManager.OnError += OnError;

LogicManager.InitializeSendProcess(); //отправление старых точек

SendToast("Service was initialized");}

Рисунок 2.5 – Метод Initialize класса LocationService

На рисунке 2.5 вызывается метод для оправки старых сообщений, он показан на рисунке 2.6:

public void InitializeSendProcess()

{

try

{

var dbPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal), "trackerdb.db3"); //получение адреса базы данных

var db = new SQLiteConnection(dbPath); //создание подключения

db.CreateTable<Point>(); //создание таблицы, если нету

//linq-запрос на точки, которые не отправлены на удалённый сервис

var points = db.Table<Point>().Where(p => p.Acked == false);

if (points.ToList().Count == 0)

return;

foreach (var p in points)

{

\_udpServer.Add(p); //добавление найденный точек в очередь отправки

}

}

catch (Exception e)

{

OnError(e); //сигнализированние об возникшей ошибке

}

}

Рисунок 2.6 – Метод InitializeSendProcess класса LogicManager

Основной метод, выполняющий запрос на позицию показан на рисунке 2.7:

private void GetLocation(LocationCallReason reason)

{

if (\_locationManager == null)

return;

if (IsRequestSendeed) //проверка, выполнен ли предыдущий запрос

{

return;

}

switch (reason)

{

case LocationCallReason.Step: //причина вызова - StepDetector

{

if (LastLocationCall < DateTime.Now.AddMinutes(-StepTimeOutMinutes)) //проверка на таймауты

{

ForceRequestLocation(\_locationManager); //запроса на позицию

}

break;

}

case LocationCallReason.Timer: //причина вызова - таймер

{

if (LastLocationCall < DateTime.Now.AddHours(TimerTimeOutHour)) //проверка на таймауты

{

ForceRequestLocation(\_locationManager); //запрос на позицию

}

break;

}

default:

{

throw new ArgumentOutOfRangeException(nameof(reason), reason, null); //инициирование исключения, в случае неизвестной причины вызова

}

}

}

}

Рисунок 2.7 – Метод GetLocation класса LogicManager

Для обработки события, которое вызывается при нахождении точки существует метод OnLocationChanged, реализация которого показана на рисунке 2.8:

public void OnLocationChanged(Location location)

{

LastLocationCall = DateTime.Now;

IsRequestSendeed = false;

if (location == null)

{

Console.WriteLine("Unable to determine your location. Try again in a short while."); //вывод предупреждения

}

else

{

OnLocationChangedEvent(location); //вызов события изменения позиции

var point = new Point(\_uniqueId, location); //создание точки

SaveInBase(point); //сохранение точки в базе данных

SendToServer(point); //отправка точки на удалённый сервис

}

}

Рисунок 2.8 – Метод OnLocationChanged класса LogicManager

Для сохранения данных, в классе Point есть метод SaveInBase, его реализация показана на рисунке 2.9:

public void SaveInBase()

{

var dbPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal), "trackerdb.db3"); //создание пути к базе данных

var db = new SQLiteConnection(dbPath); //создание подключения к базе

db.CreateTable<Point>(); //создание таблицы, если нету

db.Insert(this); //запись точки в базу данных

}

Рисунок 2.9 – Метод SaveInBase класса Point

Сохранив точку в базе данных, приложение отправляет её на удалённый сервис, методы, которые это реализую показаны на рисунке 2.10:

public void Add(Point point) //метод для добавления сообщений в буфер

{

lock (\_messageBuffer)

{

\_messageBuffer.Add(point); //добавление сообщения в буфер отправки

}

}

public void Ack(Point point) //метод, вызываемы при получении отчёта о доставке

{

lock (\_messageBuffer)

{

var item = \_messageBuffer.Single(p => p.Ack == point.Ack);

\_messageBuffer.Remove(item); //поиск сообщения и удаление его из буфера

}

OnTimerCall(null);

}

private void OnTimerCall(object state) //метод для проверки буфера

{

\_timer.Change(TimeIntervalInMilliseconds, Timeout.Infinite);

if (\_messageBuffer.Count > 0)

{

var task = Task.Run(() => Send(\_messageBuffer[0].GetMessageToSend()));

}//запуск в асинхронном режиме метода отправки сообщения из буфера

}

private void Send(string message) //метод отправки сообщения

{

var remoteEp = new IPEndPoint(0, 0);

\_udpClient.Connect(\_host, \_port);

\_udpClient.Send(Encoding.UTF8.GetBytes(message), Encoding.UTF8.GetBytes(message).Length);

}

Рисунок 2.10 – Методы класса UdpServer, используемые для отправки точки на удаленный сервис

## 2.3 Элементы интерфейса приложения он-лайн геолокации

Приложение, созданная для он-лайн геолокации, имеет название «Personal Tracker».

Данная программа при открытии предоставляет пользователю главное меню, изображенное на рисунке 2.11.

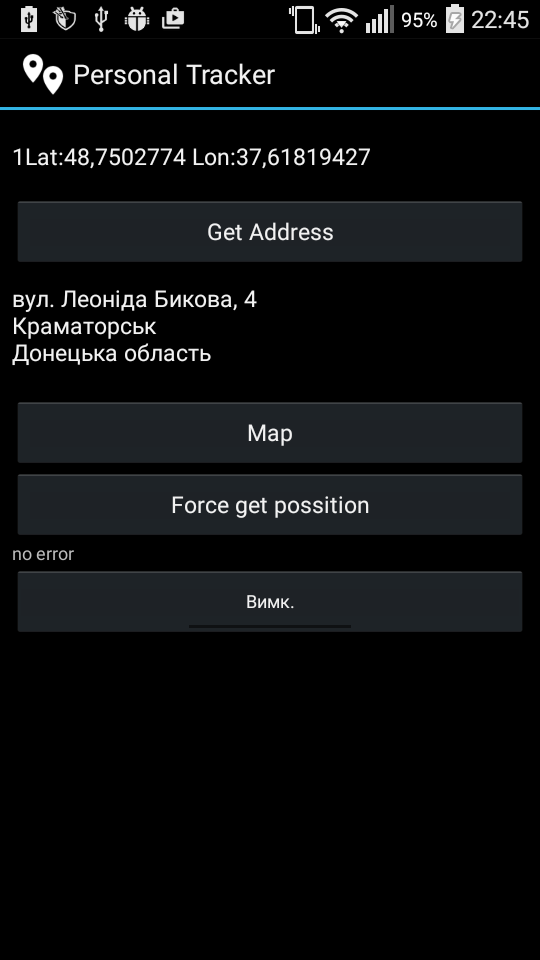


Рисунок 2.11 – Основное окно приложения он-лайн геолокации

Основное окно приложения управляет сервисной частью приложения. Для того, чтобы запустить сервис, необходимо нажать кнопку «Вимк.» (язык кнопки зависит от локальных настроек телефона). Окно приложения после активации сервиса изображено на рисунке 2.12.

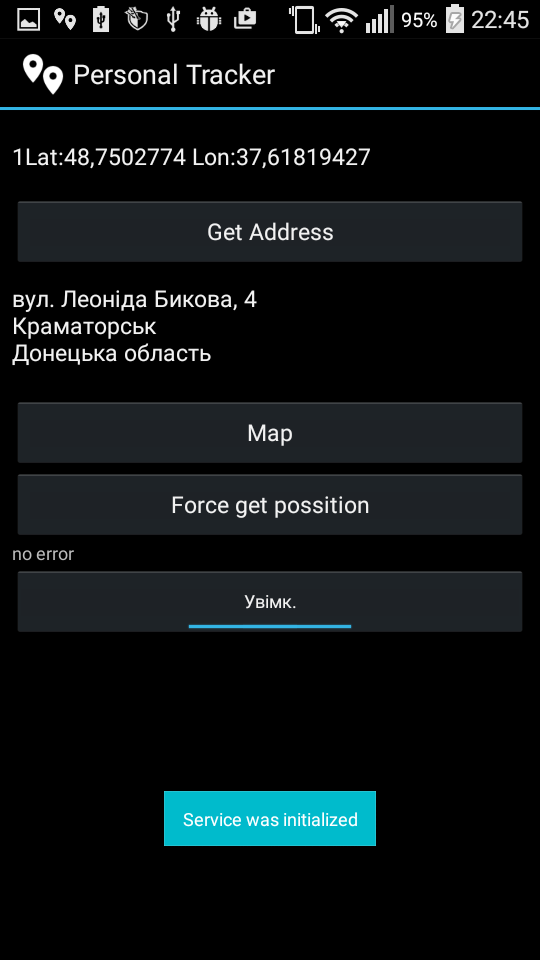


Рисунок 2.12 – Основное окно приложения, после активации сервиса

Запуск сервиса сгенерировал нотификацию, которая оповещает пользователя о том, что сервисная часть приложения стабильно работает. Окно нотификации изображено на рисунке 2.13.

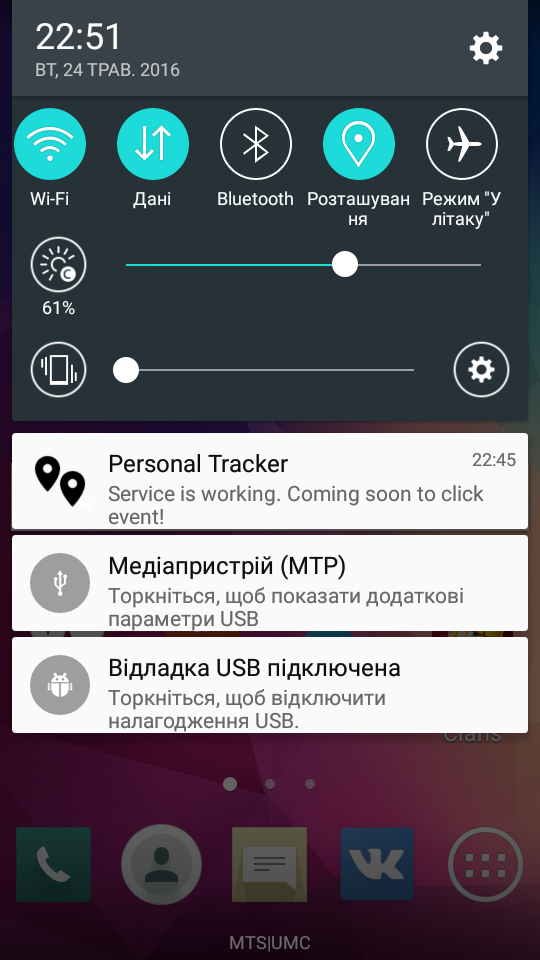


Рисунок 2.13 – Нотификация приложения «Personal Tracker»

Кроме управления сервисной частью приложения, основное окно позволяет отрыть локальную карту траектории движения устройства. Траектория строится на основе точек, которые были сохранены в мобильной базе данных устройства. Перед просмотром карты, необходимо выбрать дату, за которую пользователь хочет просмотреть историю передвижения. Окно выбора даты изображено на рисунке 2.14.

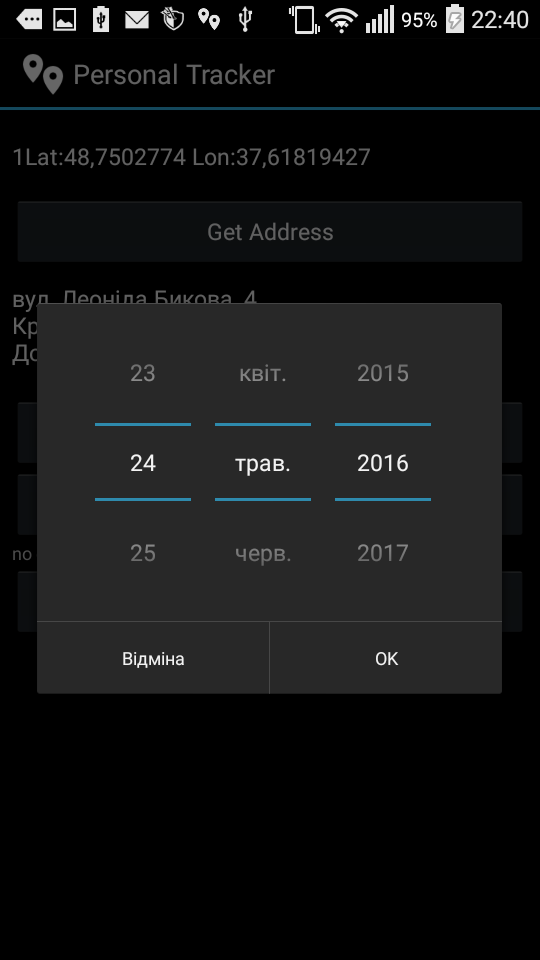


Рисунок 2.14 – Окно выбора даты

Выбрав дату приложение выполняет sql-запрос к локальной базе данных. Использование ORM для SQLite позволило написать запрос синтаксисом языка C#. На рисунке 2.15 изображен метод, выполняющий запрос к ORM системе на основе выбранной даты.

private List<Point> GetPointsByDate(DateTime date)

{

var dbPath = Path.Combine(System.Environment. GetFolderPath(System.Environment.SpecialFolder.Personal), "trackerdb.db3");

var db = new SQLiteConnection(dbPath);

var points = new List<Point>();

points.AddRange(db.Table<Point>().Where(point => point.GpsTime.Date == date.Date));

return points;

}

Рисунок 2.15 – Метод, выполняющий запрос точек к ORM SQLite

Тем не менее ORM выполняет обычной sql-запрос, который можно без проблем просмотреть. Аналогичный код на sql изображен на рисунке 2.16

SELECT \*

FROM Point p

WHERE (p.GpsTime = @time)

ORDER BY p.id desc;

Рисунок 2.16 – SQL запрос точек из SQLite базы.

Выполнив запрос приложение открывает окно карты с построенной траекторией движения. Окно карты, которой изображена траектория движения пользователя изображена на рисунке 2.17.

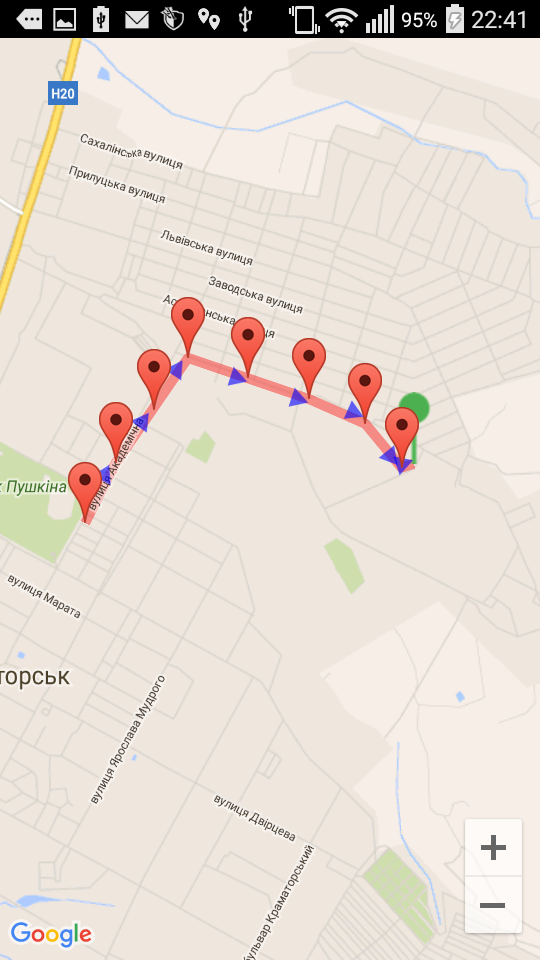


Рисунок 2.17 – Окно карты с траекторией движения пользователя

Нажав на любой маркер на траектории, пользователь может получить информации о том пользователе, который эту точку отобразил на карте, о времени точки и скорости движения устройства в момент сохранения точки. Окно, открывающиеся после нажатия на маркер изображено на рисунке 2.18.

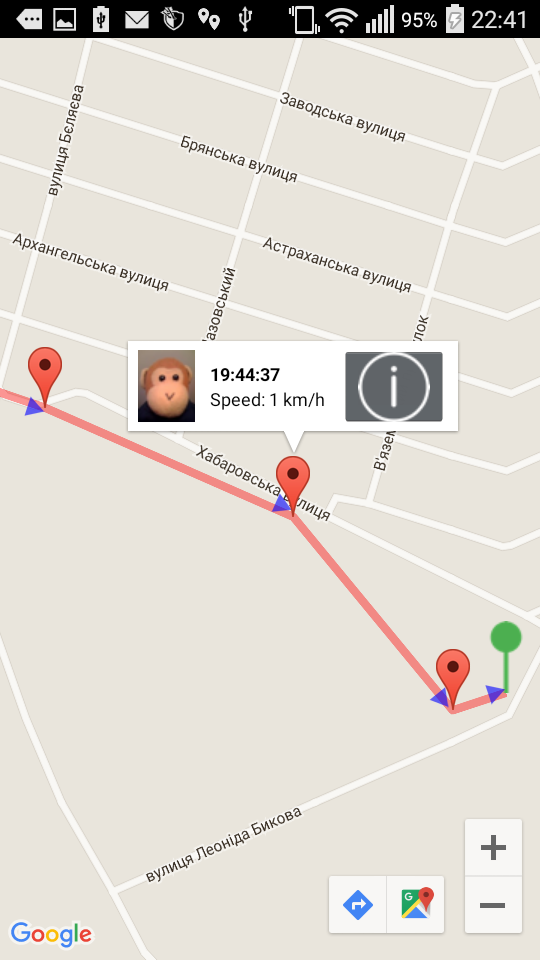


Рисунок 2.18 – Карта с всплывающий окном информации

Для периодических запросов приложению необходим файл конфигурации, который находится в локальной базе данных в таблице Configuration. Метод с обращением к ORM изображен на рисунке 2.19. Аналогичный ему sql-запрос на рисунке 2.20.

var dbPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Personal), "trackerdb.db3");

var db = new SQLiteConnection(dbPath);

db.CreateTable<Configuration>();

var config = db.Get<Configuration>(0);

Рисунок 2.19 – Метод, выполняющий запрос конфигурации к ORM SQLite

SELECT \*

FROM Configuration c

WHERE (c.id = 0);

Рисунок 2.20 – SQL запрос конфигурации из SQLite базы.

Карта с траекторией передвижения на удаленном сервисе отображена на рисунке 2.21

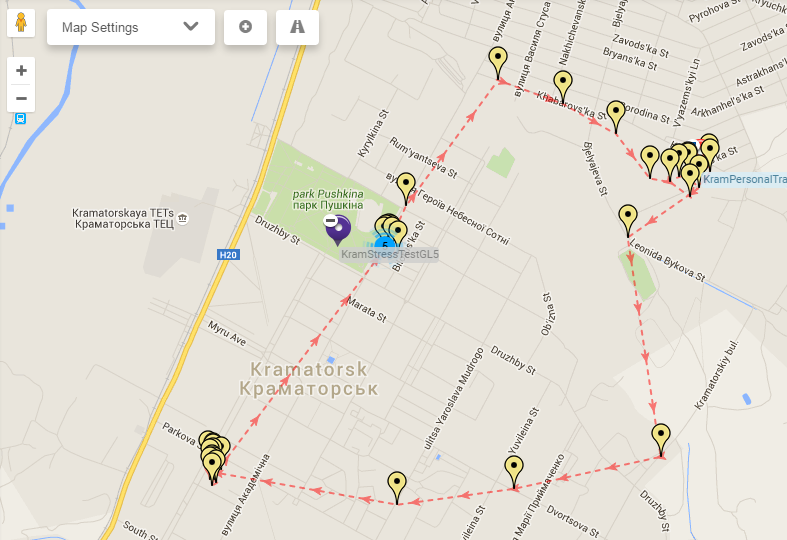


Рисунок 2.21 – Карта на удаленном сервисе

Навигация по датам на удаленном сервисе отображена на рисунке 2.22.

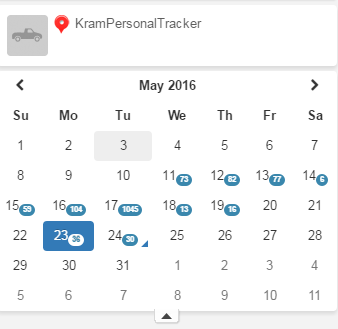


Рисунок 2.22 – Фильтр по датам на удалённом сервисе

Так же, как и в локальной версии, сервис поддерживает функционал нажатия на маркер с просмотром информации по точке. Это показано на рисунке 2.23.

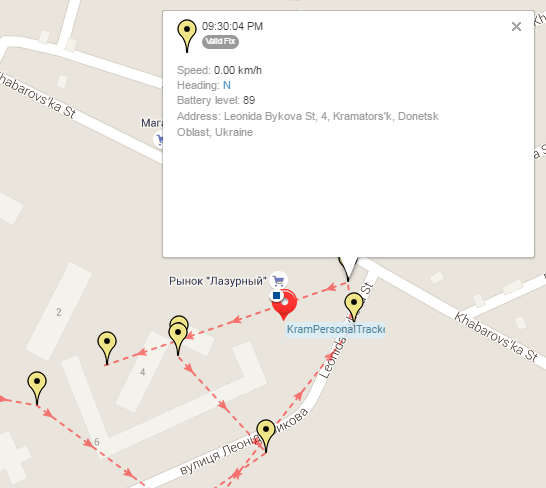


Рисунок 2.23 – Всплывающее окно информации маркера

## 2.4 Руководство пользователя приложения он-лайн геолокации

Перед использованием приложения, пользователь, если он хочет использовать возможности работы с удалённым сервисом должен связаться с администрацией сервиса и указать imei адрес своего устройства для подключения его в систему.

Установив apk-приложение на своё устройство, пользователь, открыв приложение увидит основное окно. Основное окно приложения со сносками находится на рисунке 2.24.

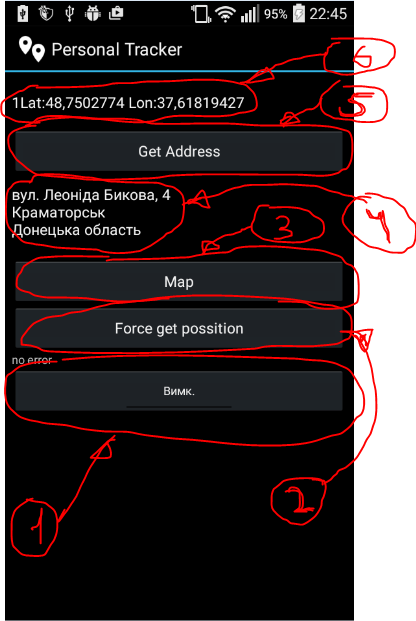


Рисунок 2.23 – Всплывающее окно информации маркера

Описание основного окна приложения:

1. Кнопка, выполняющая роль управления сервисной частью приложения. Её активность\неактивность отражает статус приложения.
2. Кнопка, выполняющая внеочередной запрос на получение геопозиции.
3. Кнопка, вызывающая локальную карту.
4. Адрес, полученный в результате геокодирования полученной геопозиции.
5. Кнопка, принудительно запускающая процесс геокодирования (может понадобится, если на момент получения новой точки отсутствовала связь).
6. Адрес последней полученной геоточки.

Запустив сервисную часть, приложение начнет получать координаты устройства и записывать их в локальную базу данных.

Выполнив нажатие на кнопку 3 – «Map» приложение откроет локальную карту. На рисунках 2.17 и 2.18 содержится пример локальной карты.

Для того, чтобы в фоновом режиме удостоверится в стабильной работе приложения следует обратить внимание на нотификацию телефона, которая изображена на рисунке 2.13. Если нотификация есть, и она соответствует примеру, значит сервис работает стабильно.

Если пользователь зарегистрировался на удалённом сервисе, то он может просматривать историю своих или других пользователей в системе. На рисунках 2.21-2.23 изображены интерфейсы удалённого сервиса

# ПРИЛОЖЕНИЕ А РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ НА РАЗРАБОТКУ ПРИЛОЖЕНИЯ ОН-ЛАЙН ГЕОЛОКАЦИИ

Приложееие Б - ведомость

## А.1 Введение

Название программного продукта – «Personal Tracker». Данный разрабатываемый программный продукт предназначен для осуществления он-лайн трекинга как одного пользователя в частности, так и синхронизации с сервером для геолокации всех пользователей системы.

## А.2 Основание для разработки приложения он-лайн геолокации

Разработка приложения для он-лайн геолокации выполняется на основе индивидуального задания к дипломной работе, выданного на кафедре КИТ Донбасской государственной машиностроительной академии по приказу ректора академии.

## А.3 Назначение разработки для приложения он-лайн геолокации

Приложение «Personal Tracker» разработано для того, чтобы упростить ручную работу записи текущей позиции при грузоперевозках, работе в труднодоступных местах, для родительского контроля или других задач, где полезно использовать системы геолокации.

## А.4 Требования к программному продукту приложения он-лайн геолокации

### А.4.1 Требования к функциональным характеристикам для приложения он-лайн геолокации

Приложение по осуществлению он-лайн геолокации должно выполнять следующие функции:

* обеспечить ведение базы данных для полученных геолокационных точек на клиентской части;
* обеспечить проверку запросов на корректность;
* обеспечить проверку выходных данных;
* обеспечивать синхронизацию базы данных между клиентом и сервером;
* обеспечить авторизованным пользователям отображение траектории движения на карте для клиентской и серверной части;
* обеспечить фильтр отображаемых данных по дате;
* обеспечить управление приложением клиентом и удалённо.

### А.4.2 Требования к надежности приложения он-лайн геолокации

Приложение для он-лайн геолокации должно требовать незначительный объем ресурсов системы (место на жестком диске, оперативная память), нормально выполнять свой основной функционал, а также:

* устойчиво функционировать и не приводить к сбоям операционной системы;
* быстро обрабатывать запросы пользователя;
* обеспечивать обработку некорректной работы приложения с выдачей соответствующих сообщений;
* корректно извлекать значения из базы данных.

### А.4.3 Условия эксплуатации приложения он-лайн геолокации

Условия эксплуатации приложения он-лайн геолокации определяется СанПиН 2.2.2 545-96 «Гигиенические требования к видеодисплейным терминалам, персональным вычислительным машинам и организации работы».

### А.4.4 Требования к составу и параметрам технических средств приложения для он-лайн геолокации

Для клиентской части:

Минимальные параметры:

* ОС Android 5.0.2 или старше;
* Процессор Qualcomm Snapdragon 400;
* ОЗУ 256 МБ;
* Встроенный Gps-модуль;

Для серверной части:

Требования к параметрам программного продукта определяются требованиями к функционированию платформы .NET Framework 4.5 и фреймворка Xamarin, а также операционной системой (минимальные параметры):

* ОС Microsoft Windows 7 или старше;
* процессор Intel i3 2.50 GHz или аналогичный Athlon:
* ОЗУ 1 ГБ;
* видеоадаптер с памятью 32 МБ;
* клавиатура и мышь.

### А.4.5 Требования к информационной и программной совместимости для приложения он-лайн геолокации

Для нормальной работы приложения он-лайн геолокации необходимо наличие платформы .NET Framework 4.0 или mono соответствующей версии или выше. В качестве ОС необходимо наличие ОС Windows 7 или выше, или Linux с поддержкой платформы mono. Для работы с данными необходимо использование файлов базы данных, которые можно редактировать с помощью любого MySql редактора для серверной или SQLite редактора для клиентской.

## А.5 Требования к программной документации приложения он-лайн геолокации

Программная документация должна включать:

* техническое задание;
* рабочий проекта он-лайн геолокации на языке UML;
* исходный код с комментариями;
* руководство пользователя по установке;
* руководство пользователя по эксплуатации.

## А.6 Технико-экономическая эффективность приложения он-лайн геолокации

?????????????????????????????????????????????????????????????????????

ВЕНТЦЕЛЬ МАТ СТАТИСТИКА ТЕОРИЯ ВЕРОЯТНОСТЕЙ

ФУНКЦИЧ ЛОТНОСТ РАСПРЕДЕЛЕНИЯ

## А.7 Стадии и этапы разработки приложения он-лайн геолокации

Стадии и этапы разработки должны соответствовать ГОСТ 19.101-77 и состоять из следующих пунктов: техническое задание, эскизный проект, технический проект, рабочий проект, стадия внедрения.

Стадии и этапы разработки приложения для он-лайн геолокации приведены в таблице А.1.

Таблица А.1 – Стадии и этапы разработки приложения он-лайн геолокации

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Этап | Содержание | Срок |
| 1 | Техническое задание | Описание характеристик приложения он-лайн геолокации. Формирование требований. Выделение стадий и этапов разработки. | 26.03  –  29.03 |
| 2 | Эскизный  проект | Предварительная разработка структуры входных и выходных данных для приложения. Разработка логической модели работы приложения для соответствующих прецедентов. Разработка диаграммы прецедентов, диаграммы классов предметной области «он-лайн геолокация» и диаграммы последовательностей. | 30.03  –  05.04 |
| 3 | Технический проект | Уточнение структуры входных и выходных данных. Разработка ER-диаграммы предметной области «он-лайн геолокации». Проектирование экранных форм. | 06.04  –  15.04 |
| 4 | Рабочий проект | Реализация приложения для он-лайн геолокации. Проведение предварительных испытаний. Разработка документации. | 16.04  – 15.05 |

Продолжение таблицы А.1

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Этап | Содержание | Срок |
| 5 | Внедрение | Подготовка и передача приложения для он-лайн геолокации. Внесение корректировок в программное обеспечение и документацию. | 15.05  –  20.05 |

## А.8 Порядок контроля и приемки приложения для он-лайн геолокации

Контроль корректности функционирования и пригодности программы к эксплуатации выполняется совместно разработчиком и заказчиком, на основании результатов тестирования программного продукта, предоставляемых разработчиком.