

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 приложения «Умная остановка» на платформе Android

Специальность: 5В070400 – «Вычислительная техника и программное обеспечение»

Выполнил: Битюкова Д.А.      Группа: ВТ-15-2  
Научный руководитель: PhD, доцент Досжанова А.А.

Консультанты:

по экономической части: к.э.н., профессор \_\_\_\_\_ Ж.Г. Аренбаева  
« 13 » \_\_\_\_\_ мая 2019 г.

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

по применению  
вычислительной техники: ст. преп. \_\_\_\_\_ М.Н. Майкотов  
« 14 » \_\_\_\_\_ мая 2019 г.

Нормоконтролер: ст. преп. \_\_\_\_\_ А.А. Айтказина  
« 15 » \_\_\_\_\_ мая 2019 г.

Рецензент: PhD, доцент \_\_\_\_\_ Б.С. Каленова  
« \_\_\_\_ » \_\_\_\_\_ 2019 г.

Алматы 2019

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

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070400 – «Вычислительная техника и  
программное обеспечение»

**ЗАДАНИЕ**

на выполнение дипломного проекта

Студенту Битюковой Дарье Александровне

Тема проекта: Разработка приложения «Умная остановка» на платформе Android

Утверждена приказом по университету № 124 от «26» октября 2018 г.

Срок сдачи законченного проекта «24» мая 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Руководство системы менеджмента качества на предприятии; международные стандарты ИСО-9001, данные преддипломной практики.

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

- аналитическая часть;
- проектная часть;
- экспериментальная часть;
- экономическая часть;
- безопасность жизнедеятельности;
- приложение А. Техническое задание;
- приложение Б. Листинг программы;
- приложение В. Акт внедрения.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 12 таблиц, 33 иллюстрации.

Основная рекомендуемая литература:

1 Медникс З., Дорнин Л. Программирование под Android. Издательство Питер, 2012.

2 Амелин К. С., Граничин О. Н., Кияев В. И., Корявко А. В.. Введение в разработку приложений для мобильных платформ. Издательство ВВМ, 2011.

3 Мельникова О.М.: Смартфоны на Android. Издательство Эксмо, 2013.

4 Каймин В.А. Информатика: Учебное пособие: Изд. 2-е. Издательство РИОР, 2007.

5 Варакин М.В. Разработка мобильных приложений под Android. УЦ Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Аренбаева Ж.Г.	04.03.2019 - 13.05.2019	
Безопасность жизнедеятельности	Бекбасаров Ш.Ш.	27.03.2019 8.05.2019	
Программное обеспечение	Майкотов М.Н.	04.04.2019 14.05.2019	
Нормоконтролер	Айтказина А.А.	02.04.2019 - 15.05.2019	

**ГРАФИК**  
подготовки дипломной проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Аналитическая часть	05.11.2018г. - 22.12.2018г.	
Проектная часть	07.01.2019г. - 30.01.2019г.	
Экспериментальная часть	04.02.2019г. - 13.04.2019г.	

Дата выдачи задания «25» октября 2018 г.

Заведующий кафедрой \_\_\_\_\_ Т.С. Картбаев

Научный руководитель проекта  А.А. Досжанова

Задание принял к исполнению студент  Д.А. Битюкова

## Аңдатпа

Дипломдық жобаның тақырыбы: Android платформасында «». практикалық құндылығы бағыты пайдаланушы, ұтқырлық және ыңғайлы интерфейс қосымшалар ретінде оңайлатылған нысанын пайдалану арқылы іріктеу қабылдау маңызды жеделдету болып табылады. бағдарлама GPS-навигатор жабдықталған осы бағдарламалық қамтамасыз етуді әзірлеу жылы, Алматы қаласында көлік жүйелерін зерттеу және талдау тартылған. Бұл тезис жобаға App құны туралы экономикалық есептеулерді жүргізілді, сондай-ақ кеңсесінде басты проблема ол қайта құру офис жасалды, содан кейін желдету проблема екенін көрсетті өнеркәсіптік қауіпсіздік алдын алу шеңберінде, жасалған.

Диссертациялық жобаның келесі мақсаттары анықталды: пайдаланушының орналасқан жерін қадағалау, маршруттарды іздестіру және оларды картада көрсету үшін өтінімді жасау.

Дипломдық жобаны зерттеу объектісі қалалық трафик саласында карталар мен қазіргі заманғы технологияларға арналған геолокаторды тану құралы болды.

## **Аннотация**

Тема дипломного проекта: «Разработка мобильного приложения на платформе Android «Умная остановка»». Практическая ценность заключается в существенном ускорении принятия выбора маршрута пользователем, мобильности и упрощенном виде использования приложения в виде удобного интерфейса. В разработке данного программного продукта задействовано исследование и анализ транспортных систем города Алматы, также данное приложение оснащено GPS-навигатором. В данном дипломном проекте были проведены экономические расчеты по стоимости приложения, а также произведена профилактика по части производственной безопасности, которые показали, что главной проблемой в офисе являлось проблема с вентиляцией, после чего была произведена реконструкция офиса.

## **Annotation**

Theme of the graduation project: "Developing a mobile application on the Android platform" Smart Stop ". The practical value is to significantly accelerate the adoption of the route by the user, mobility and simplified use of the application in the form of a convenient interface. In the development of this software product involved the study and analysis of transport systems of the city of Almaty, also this application is equipped with a GPS-navigator. In this thesis project, economic calculations were made on the cost of the application, as well as preventive measures were made regarding industrial safety, which showed that the main problem in the office was a problem with ventilation, after which the office was reconstructed.

The following objectives of the thesis project were set: to develop an application to track the user's location, search for routes and display them on the map.

The object of the research of the graduation project was geolocation recognition tools for maps and modern technologies in the field of urban traffic.

## Содержание

Введение	8
1 Анализ предметной области.	10
1.1 Теоретическая база транспортной системы	11
1.2 Обзор мобильных приложений	13
1.3 Анализ существующих казахских и зарубежных разработок	21
1.4 Подведение итогов анализа и постановка задачи	25
2 Описание и обоснование выбора инструментов разработки	27
2.1 Анализ современных языков программирования	27
2.2 Выбор языка программирования Java и среды разработки	29
3 Программная реализация	40
3.1 Проектирование информационной системы	40
3.2 Аспекты реализации	43
3.3 Разработка и описание интерфейса	51
4 Экономическая часть	54
4.1 Определение сложности разработки программного обеспечения	57
4.2 Расчет затрат на разработку программного продукта	60
4.3 Определение договорной цены программного продукта	61
5 Производственная безопасность	63
5.1 Анализ условий труда	67
5.2 Расчет вентиляции	68
Заключение	70
Список литературы	71
Приложение А. Техническое задание	73
Приложение Б. Листинг программы	77
Приложение В. Акт внедрения	81

## Введение

Если маршрут дом – работа – дом не вызывал проблем, то возможность проезда в незнакомую точку вызывала дискомфорт. С мобильным приложением для видимости всех имеющихся маршрутов все будет намного проще и быстрее. Сейчас для любого большого города имеются навигаторы. Таким образом, главной целью данного дипломного проекта является создание геолокационного приложения. Из поставленных целей следует выделить следующие задачи:

- сбор и обработка информации по технологиям создания мобильных приложений;
- выбор средств и языка программирования для разработки приложения;
- этап разработки мобильного приложения;
- дать оценку экономической эффективности внедрения информационной системы.

Пользователям необходимо больше мобильности и удобства, чтобы осуществлять ежедневное передвижение по городу «от двери до двери» с максимальным комфортом. Интеграция и комплексный подход должны стать краеугольным камнем транспортной жизни города, в первую очередь, при использовании пассажиром в поездке различных видов транспорта.

Предметом исследования является повышение эффективности геолокационного применения пользователей города Алматы за счет использования данного приложения.

Практическая ценность заключается в существенном ускорении принятия выбора маршрута пользователем, мобильности и упрощенном виде использования приложения в виде удобного интерфейса.

В процессе написания дипломной работы применены методы: системного подхода к объекту и предмету дипломной работы, анализа, сравнения, конкретизации темы дипломного проекта с помощью практического примера. Работа графически использована с использованием иллюстраций.

В роли статистически-информационной базы были использованы информационно-аналитические статьи, предоставляющие полную информацию о положении и состоянии транспортных ситуаций города Алматы, специализированные и информативные сайты.

В дипломный проект включены такие пункты как введение, 5 пунктов, заключение, список литературы и приложения А,Б,В. Введение включает актуальность, предмет, объект, цель, задачи, методику и практическую значимость дипломной работы. В первой главе продемонстрирован анализ и теоретическая база определений, в ней описаны понятия, проведен сравнительный анализ уже с существующими приложениями. Вторая глава описывает значимость и обоснованность выбора инструмента разработки. Третья глава содержит описание данного программного продукта. В



заклучении выделены и описаны основные результаты и выводы данного дипломного проекта.

## **1 Анализ современного состояния вопроса**

### **1.1 Теоретическая база транспортной системы**

Социальный транспорт считается одним из основных элементов существования населенных пунктов. И если кто-то гарантирует комфортное, чистое и не опасное движение с наименьшим количеством выбросов и по действующим тарифам, то в этом случае такой вид социального транспорта является ежедневным и необходимым атрибутом существования тысяч людей.

Преимущество использования индивидуального автомобильного транспорта-это представления современных условий в городе Алматы. Эту ситуацию абсолютно естественно можно оценить, при этом так же как и у населения нет шансов приобрести целый комфорт, используя социальные средства, и по этой причине любой человек мечтает получить собственный автомобильный транспорт.

Так же как и в городе преобладает индивидуальный автомобильный транспорт это было образование пробок и сдвиги в худших экологических условиях мегаполиса. Это приводит к складкам на путях, высокой закупорке, а на уровне градостроительства потребность в автомобилях начинает доминировать над необходимыми процессами пешеходов.

Работа акимата города Алматы заключается в формировании планов по улучшению нашей сферы, планов экологической семьи и реализации таких изменений, а также о введении специальных автобусных полос на дорогах мегаполиса Алматы, внедрение микроавтобуса в Газе, наиболее многофункциональных и практичных дорожных денег,

Кто также взял за основу, чтобы внедрить концепцию легкорельсовых транспортных средств (трамвая) в городе. Общество, работающее в этой сфере, пытается внедрить новейшие стандарты с целью сокращения выбросов в муниципальном транспорте, так как в рамках массового полностью стабильного формирования социальных транспортных средств является необходимым условием, которое оказывает большое влияние на общее материальное благополучие сообщества и находятся вокруг личности экосистем.

Алматы-огромный мегаполис, с большим количеством способностей, жители которого в кратчайшие 20 лет вырастут огромными, чем полмиллиона человек. Для целей мегаполиса на начальном участке возникают аналогичные трудности, так как перегруженность дорог, социального транспорта недостаточно для осуществления устойчивой перевозки пассажиров, а увеличение количества отдельных транспортных средств также усугубляет ситуацию на путях.

Алматинцам, как и большему количеству населения нынешних населенных пунктов, следует быть более мобильными и комфортными от социальных транспортных средств для того, чтобы осуществлять ежедневное движение по городам "от двери до двери" с наибольшим удобством.

Ассоциация и системный аспект должны быть базовыми камнями автотранспортных средств существования мегаполиса, в первую очередь, наличие использования пассажирам, путешествующим на различных. Компания АСТ Kazakhstan сделали анализ по оценке реформ, собрали мнения Алматинцев, данный анализ показал, что частным транспортом пользуются 52 % опрошиваемых, общественным транспортом пользуются 49% , сервис такси используют 16%, к услугам метро обращаются 9%. Только для 4% алматинцев велосипед является одним из способов передвижения. И ещё 5% отметили, что вообще не пользуются никаким видом транспорта. 58% опрошиваемых довольны работой общественного транспорта. Опрошиваемые довольны такими тезничсекими внедрениями как электронно системой оплаты 28%, внедрение более удобных автобусов 28%, также качесвто обслуживания 9%. Алматинцы отметили, что благодаря данным внедрениям уменьшилось время пути за счет автобусной линии, были введены новые маршруты, также увеличился объем транспорта.

Стабильный муниципальный автотранспорт-эта концепция обеспечиваетнеизменное качество, мобильность и удобство использования перевозки с целью вообще резидентов в долгосрочной возможности, в то же время оказывает положительное влияние на окружающую местность, а также социальная и финансовая стабильность регионального общества в полном объеме. В развитых странах мира вопрос о доступности общественного транспорта является одним из самых важных вопросов в создании городской среды. Доступность – свойство транспорта, зданий, помещений, объектов и территорий, позволяющее беспрепятственно достичь места и воспользоваться услугой. При разработке решений по формированию доступной среды в различных областях основными являются несколько принципов:

- обеспечение беспрепятственного передвижения, достижение места назначения и возможность воспользоваться услугами;
- предотвращение возможности получения травм, причинения вреда другим людям и (или) окружающей среде, обеспечение соблюдения различных правил и норм;
- обеспечение доступности не только в границах отдельных объектов и элементов этих объектов, но и связей между ними;
- создание условий для минимальных затрат и/или компенсации усилий при передвижении, получении услуг и информации;
- обеспечение своевременного получения и осознания информации одновременно различными способами ее передачи (визуальными, акустическими, тактильными);
- использование интерактивных систем и средств передачи и обмена информацией.

Общественный транспорт в Казахстане хорошо развит в крупных городах. В большинстве городов транспортная система состоит из автобусов и маршруток. Местный транспорт не представляет особой сложности для

туристов, несмотря на то, что многие из указателей и информационных табло – на казахском. Трудностей с оплатой проезда не возникнет, в большинстве автобусов и маршруток деньги нужно передать водителю или, что встречается реже, кондуктору. Единственная система метро в Казахстане расположена в Алма-Ате. Состоит Алма-Атинский метрополитен из одной линии с 7 станциями. Главным транспортом в городах Казахстана являются автобусы. Кроме автобусов по тем же направлениям зачастую ездят и маршрутки. Останавливаются автобусы на всех остановках своего маршрута.

Самый популярный вид общественного транспорта в России, как и в Казахстане – автобусы. Они есть в большинстве даже самых отдаленных населенных пунктов. В крупных городах – это развитые автобусные сети с постоянными маршрутами и расписанием. В деревнях и селах обычно есть один-два автобуса, которые связывают соседние населенные пункты и ходят несколько раз в день с большими перерывами. Наряду с автобусами в России есть троллейбусы и трамваи, которые распространены гораздо меньше: первые есть всего в 85 российских городах, а трамваи ходят только по 61 населенному пункту. Во многих городах также можно воспользоваться услугами маршрутных такси. Билеты на общественный транспорт всегда можно купить при посадке у водителя или кондуктора. В некоторых городах есть специальные кассы по продаже проездных билетов и абонементов. Поездка на городском автобусе, трамвае или троллейбусе стоит от 15 до 30 рублей в зависимости от города. А цены на билеты на междугородные автобусы могут достигать нескольких тысяч.

В бывшей советской стране, ныне входящей в состав Евросоюза - Латвии, общественный транспорт представлен автобусами, трамваями и троллейбусами. В Риге, столице государства, на сегодняшний день действует 84 маршрута. Одна из важных характеристик рижского общественного транспорта - то, что ходит он строго по расписанию, буквально по минутам.

Главным общественным транспортом Нью-Йорка является метрополитен. Этот сабвей считается одним из самых крупных в мире и по длине путей, и по количеству пассажиропотока. Метро Нью-Йорка насчитывает почти 500 станций, а протяженность путей превышает 1000 километров. Автобусы в Нью-Йорке связаны с метро и пригородными электричками.

Многие путешественники и туристы сходятся во мнении, что общественный транспорт Японии устроен просто и в то же время удобно. Это результат многолетней работы лучших умов, смыслящих в инженерии и логистике. Учитывая густонаселенность городов Страны восходящего солнца, метро, электрички, трамваи образуют громадный и широко разветвленный транспортный механизм. Принципы работы метро во многих городах схожи. Самыми перегруженными традиционно являются метрополитены в Токио и Осаке. Основной транспорт в Японии - электричка или метро. Автобусы тоже есть, и их немало. Почти все пользуются проездными типа «Онай» в Алматы.

В Турции, а именно в Стамбуле общественный транспорт пользуется высоким спросом. Это можно объяснить не только достаточно большой плотностью населения, но и огромным и непрерывным туристическим потоком. Все, что касается оплаты в автобусах, трамваях и прочих видах городского транспорта, переведено в электронный формат. Большой популярностью пользуется IstanbulKart - пластиковая карта, с которой списываются деньги при оплате проезда. На нее загружаешь 20 лир на специальном терминале. Они установлены повсюду: на улице, в аэропорту, киосках, магазинах. С этой картой ездят в метро, трамвае и автобусах. В транспорте установлены специальные девайсы, к которым нужно прикладывать карту. Карта показывает не количество поездок, а сколько лир осталось на балансе.

## **1.2 Обзор мобильных приложений**

Операционную систему Android изобрели в 2002 году. Компанию Google впечатлили работы программиста Э.Рубина. Начальным этапом разработки операционной системы предназначенных для устройств связи разрабатывала компания Android Inc., вскоре данную разработку купила компания Google..

Компания Google представила разработки данной операционной системы в ноябре 2007 года.

Это было новшеством в IT технологиях, так как Linux в это время использовался только на домашних компьютерах и ноутбуках, поэтому возникла идея создать в мобильной версии.

Начальная версия операционной системы была предоставлена 23 сентября 2008 года под названием Android 1.0 Astroboy. Это была первая устойчивая модификация концепция с имеющимися инструментами для программистов и Android Market

HTC Dream – телефон, который применил новую операционную систему. Данный телефон имел такие плюсы как дешевизна и новизна, что очень привлекало страны такие как США и Европу.

Версии описанные выше были предназначены только для мобильных устройств, поэтому зашла речь о выходе версии для планшетов. Операционная версия, где можно использовать свои мышки, клавиатуры и джойстики для более мобильной работы.

22 февраля 2011 года для них вышла своя версия операционки — Android 3.0 Honeycomb. Были добавлены такие концепции проектирования как адаптация для широких экранов, интересный и обновленный интерфейс.

Motorola XOOM- устройство, которое на которой тестировали данную версию операционной системы.

Далее в операционной системе вносились более заметные дополнения, 19 октября 2011 в версии Android 4.0 Ice Cream Sandwich был изменен до неузнаваемости интерфейс, система была названа Holo. Создавать PrintScreen,

осуществлять разблокировку с помощью лица, стабильность работы, работы с памятью, осуществление оповещений – все это было в новой версии.

С обновленной версией начали выпускать и новые мобильные устройства - Galaxy Nexus – телефон, который начал пользоваться популярностью. Но Galaxy Nexus имел проблемы с обновлением версий, поэтому нельзя назвать данный выпуск удачным, версия Android 4.4 KitKat не подходила данному телефону

Google Now – функция, которая пришла с последующими выпущенными 27 июня 2012 году версиями 4.1, 4.2 и 4.3. Отображение прогноза погоды, ближайших мест, результатами матчей – функции, которые были добавлены в новых версиях. Была упрощена установка родительского контроля. Miracast – синхронизация телефона и экрана, воспроизведение всего, что происходит на экране.

Для планшетов была разработана версия 4.1 - Asus Nexus 7. Планшеты востребованы и в наше время. С версией 4.2 выпустили планшет с более широким разрешением экрана Samsung Nexus 10, компания LG выпустила мобильный телефон Nexus 4, этот смартфон очень схож по характеристикам iPhone 6. С версией 4.3 был выпущен смартфон Asus Nexus 7 2013. Этот смартфон и на сегодняшний день считают востребованным и удобным в сравнении цены и качества.

31 октября 2013 была выпущена версия Android 4.4 KitKat, названа она так в честь заключения контракта с Nestle. Google Now функция, которую сейчас называют «Алисой» работает более стабильно и вызывается командой «Ok, Google!». В систему были внедрены смайлики Emoji. Облачный принтер – новая возможность печати текстовых файлов с мобильного устройства. Новыми программными разработками стало внедрение предкомпилятора ART, поддержка ИК-портов и шагомеров.

Мобильные устройства на Андроиде благодаря своим удобным использованием и доступностью стали опережать Apple, Windows Phone. В это же время Google совместно с LG выпустили один из самых мощнейших мобильных устройств - Nexus 5. Преимущества заключались в удобном разрешении экрана, обновлениях, которые устанавливаются в один клик, сильных технических характеристиках.

3 ноября 2014 года вышла версия 5.0/5.1 Lollipop, эту версию до сих пор хотят установить многие пользователи. Усовершенствованный, удобный интерфейс от дизайнеров Google дополнил систему новыми возможностями. Важный минус ОС был исправлен – батарея стала выносливее и прибавила во времени до 30%. При заблокированном экране стал возможен просмотр уведомлений. Обновление можно считать одним из самых масштабных с момента запуска системы: во-первых, полный редизайн, во-вторых, огромный набор новых функций и приложений, в-третьих, значительное увеличение стабильности работы системы. И это только самые важные изменения, полный список можно продолжать очень и очень долго.

На последней Google I/O была представлена Android M с избытком новых функций, таких как мультиоконность, поддержка сканеров отпечатков пальцев, Android Pay и многое другое. Вместе с презентацией ОС также ожидается новый смартфон от Google и Huawei, а также реинкарнация Nexus.

Мобильные дополнения в основном приоритете делятся на группы по назначению по этой причине замену добавок можно разделить на аналогичные сорта: контент дополнения, которые более популярны среди пользователей. Эта группа обеспечивает аналогичные виды работы, как прослушивание музыки, просмотр различных фильмов, видео и фотографий,

и так после всего прочтения числовых книг и т. д.;  
торговые добавки-добавки, выполняющие узко сфокусированную деятельность, а также упрощение деловой практики-деятельность, мониторинг сделок и

выполнение вопросов управления;  
mobile entertainment более желанный мобильный сектор приложения, характеристики выбора огромны, создатели любого периода любят

Создать новый или улучшить выпущен;  
социальные связи. Так как же у них сегодня эта группа?  
время максимальной необходимости, большинство народов социальных сетей потребность в связи, торговле и отдыхе по этой причине в мобильных устройствах оборудование как вероятность также использует популярность.

Кроме того, существует систематизация дополнений по типу  
ликвидация:  
бесплатный;  
платный коммерческий;  
free + in-app purchase - бесплатные дополнения с опциями  
условное приобретение устройств, связанных с информационным приложением;

платная + in-app purchase-комбинация коммерческого дополнения (плата за скачать) и возможность приобретения условного оборудования, связанного с этим приложением;

пробная реклама с бесплатной дополнительной пилотной версией.

Анализ аукциона мобильных дополнений.ДЕСЯТЬ Чтобы собрать полную информацию и сделать анализ по значимости и заинтересованности к мобильным приложениям ниже будет описана и показана статистика по рынку мобильных устройств. и приложений.

Формирование рынка используемых мобильных приложения зависит от продаж и известности мобильных устройств, то есть смартфонов.

Телефон -это портативный мобильный телефон с расширенной функциональностью, он может использоваться как карманный персональный компьютер (PDA), который может звонить. Преимущество телефонов над простыми мобильными телефонами-наличие достаточно сложившейся

концепции оператора, открытой изучение программного обеспечения сторонних разработчиков сотрудничество концепция-простая крышка для мобильных телефонов.

Благодаря внедрению большего объема мобильных приложений можно увеличить характеристики и количество функций в мобильных устройствах.

Данные компании **Garther**, которая занимается аналитикой, консалтингом, специализируется на рынке информационных технологий (США). Компания **Garther** одна из лидирующих компаний по продаже смартфонов. Более 425 млн.шт.(Рисунок 1.1). Рост продажах с каждым годом растет,сравнивая прошлый год с нынешним рост был осуществлен на 0,68%( в количестве эта сумма равняется около 2,8 млн.шт.)

**Table 1**  
Worldwide Mobile Phone Sales to End Users by Vendor in 1Q13 (Thousands of Units)

Company	1Q13 Units	1Q13 Market Share (%)	1Q12 Units	1Q12 Market Share (%)
Samsung	100,657.7	23.6	89,284.6	21.1
Nokia	63,215.2	14.8	83,162.5	19.7
Apple	38,331.8	9.0	33,120.5	7.8
LG Electronics	15,615.8	3.7	14,720.4	3.5
ZTE	14,606.6	3.4	17,379.7	4.1
Huawei Technologies	11,114.8	2.6	10,796.1	2.6
TCL Communication	8,515.9	2.0	7,396.6	1.7
Sony Mobile Communications	7,955.5	1.9	7,898.4	1.9
Lenovo	7,778.9	1.8	5,820.6	1.4
Yulong	7,478.8	1.8	3,146.6	0.7
Others	150,550.6	35.4	150,229.4	35.5
<b>Total</b>	<b>425,821.6</b>	<b>100.0</b>	<b>422,955.4</b>	<b>100.0</b>

Source: Gartner (May 2013)

Рисунок 1.1- Изменения результатов продаж смартфонов за 4 квартал 2018 года и 1 квартал 2019 года.

Посмотрев на вторую таблицу (рисунок 2.2 – изменение характеристик торговли смартфонами за 1 квартал 2019 года и за 1 квартал 2018 года) результат, который отображен количеством продаж мобильных устройств в мире равен 210 млн.шт, Начало 2018 года показало, что продажи на начало года были ниже на 42,9%. Сравнения с показателями на 1 квартал 2019 год продажа осуществлялась более чем на 49,3%, что показывает нам востребованность в покупке смартфонов, активный рост продаж и увеличении продаж в количественном отношении.

**Table 2**  
Worldwide Smartphone Sales to End Users by Vendor in 1Q13 (Thousands of Units)

Company	1Q13 Units	1Q13 Market Share (%)	1Q12 Units	1Q12 Market Share (%)
Samsung	64,740.0	30.8	40,612.8	27.6
Apple	38,331.8	18.2	33,120.5	22.5
LG Electronics	10,080.4	4.8	4,961.4	3.4
Huawei Technologies	9,334.2	4.4	5,269.6	3.6
ZTE	7,883.3	3.8	4,518.9	3.1
Others	79,676.4	37.9	58,537.0	39.8
<b>Total</b>	<b>210,046.1</b>	<b>100.0</b>	<b>147,020.2</b>	<b>100.0</b>

Source: Gartner (May 2013)

Рисунок 1.2 – Изменения по продажам смартфонов за 1 квартал 2019 года.



Благодаря чему может вырасти спрос на мобильные приложения? На данный момент разрабатывают более функциональные смартфоны, которые работают быстро, который имеет удобный и понятный интерфейс. Качественные смартфоны требуют более функциональные приложения. Поэтому на данный момент разработка мобильных приложений можно считать актуальной и нужной разработкой.

Посмотрим исследования рынка мобильных приложений. В ежеквартальном рыночном отчете сервиса App Annie.com (компания, которая отслеживает изменений рейтинга и предоставляет широкую базу для разработчиков) доступна действующая информация по мировому рынку мобильных приложений.

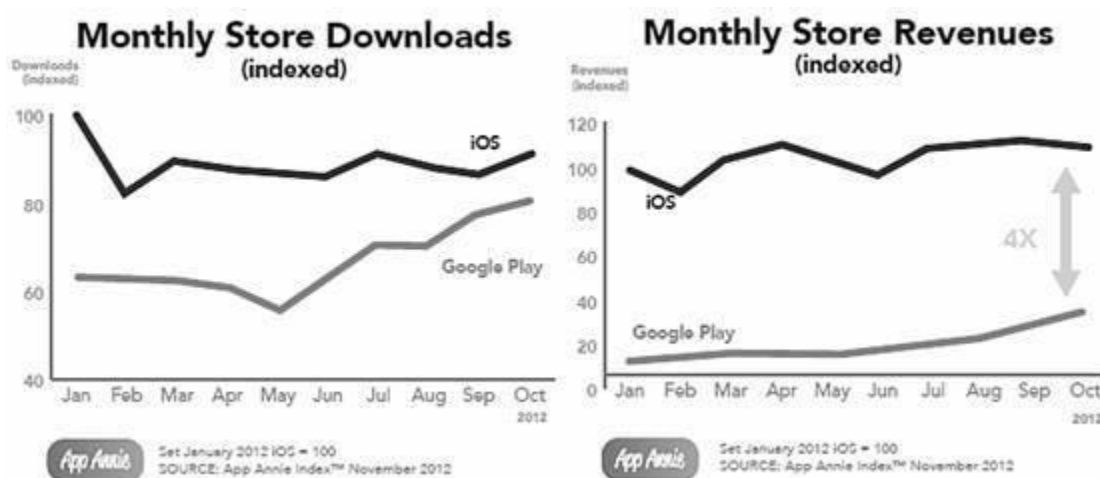


Рисунок 1.3 - Изменения количественного и денежного показателя скачиваний за январь - октябрь 2018 года, %.

Посмотрев на результаты по загрузкам за январь и октябрь 2018 года у нас на рынке имеются две крупные платформы, которые используют пользователи. Это Apple App Store и Google Play (Android). В диаграммах на рисунке 1.3 по показателям наблюдается небольшой отрыв Apple App Store от Google Play. Пользователя пользуются мобильными приложениями, рост скачиваний также увеличивается. Так как мобильные приложения на платформе Apple App Store в основном платные, денежной прибыли – это преимущество Apple App Store. Прибыль Apple Store больше прибыли Google Play в 2,6 раза за 1 квартал 2019 года. Квартальная прибыль iOS App Store составила на четверть больше предыдущей. Но прибыль Google Play тоже выросла на 90% в сравнении с прошлым годом.

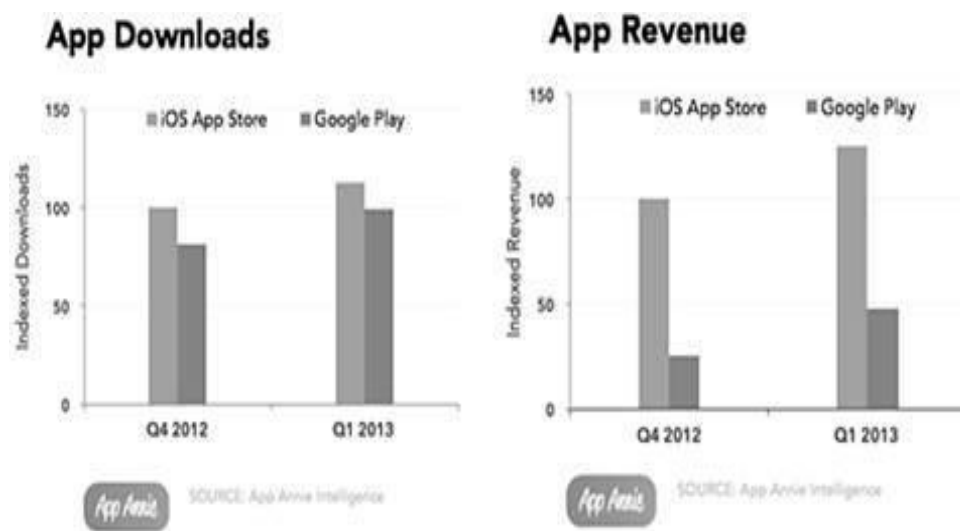


Рисунок 1.4 – Изменения количественного и денежного показателя загрузок за 2 последних квартала (4 кв.2018 -1 кв. 2019 гг.), %

Garther- компания, которая предоставляет анализ по загрузкам мобильных приложений в засимости со странами. Данный анализ за 2019 год приведен в таблице 1.1

Таблица 1.1 Apple App Store – страны лидирующие в скачивании мобильных приложений.

Ведущие страны по загрузкам в Apple App Store в 1 квартале 2019 года	
По количеству загрузок	По доходу
США	США
Китай	Япония
Великобритания	Великобритания
Япония	Китай
Франция	Австралия

По данной таблице можно сделать вывод, что у Китая замечен рост загрузок в iOS App Store. Лидируют по скачиванию мобильных приложений Япония и США, также и доход от App Store приходится в большей доли у США и Японии. Увеличению прибыли App Store США получается за счет скачивания игр, Китай также не отстает в данных показателе и за прошлый год он был в топ 10 стран. В течение прошлого квартала он поднялся на четвертую позицию с шестой, которую он занимал в четвертом квартале 2018.

Категория игр, благодаря данной категории идет основная прибыль в Китае. Данная категория принесла дохода больше, чем на 90%, самый лучший показатель всего App Store.

Google Play-тот рынок, который также не отстает в продаже и загрузках мобильных приложений. Данный анализ более интересен так как тут появляются такие страны как Южная Корея и Россия. Южная Корея – основатель компании Samsung, а как нам известно данные смартфоны разработки под платформу Android. Россия также используют смартфоны с платформой Андроид, основываясь на более низкие цены и доступность в покупке.

Таблица 1.2 Лидирующие страны по скачиванием в Google Play в 1 квартале 2019 года

Ведущие страны по загрузкам в Google Play в 1 квартале 2019 года	
По количеству загрузок	По доходу
США	Япония
Южная Корея	Южная Корея
Индия	США
Россия	Великобритания
Япония	Германия

Компания App Annie анализирую данный аспект утверждают, что лидирующие страны Google Play по скачиванием , а это США и Южная Корея гарантировали около 40% скачиваний за 1 квартал 2019 года, когда в App Store было 50%.

Также для полного анализа приведу изменения по более нужным со стороны пользователя категориям мобильных приложений за 2018 год. (рисунок 1.5).

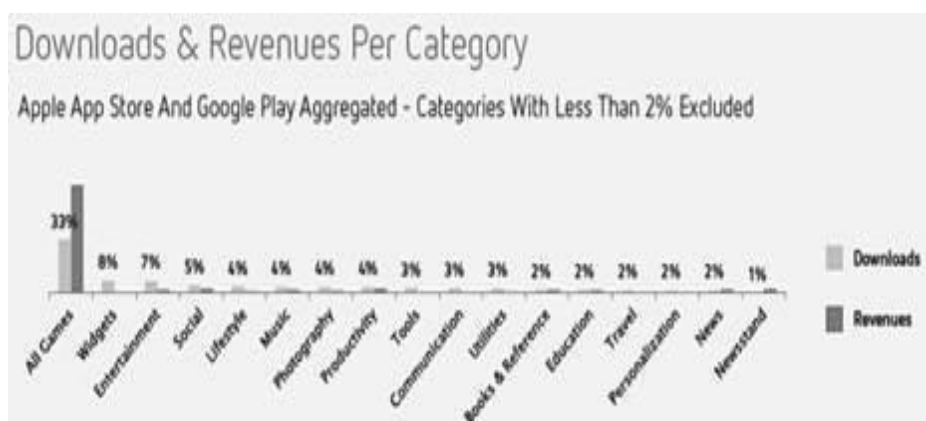


Рисунок 1.5 – Изменения более часто используемых категории в мобильных приложениях.

На Рисунке 1.5 отчетливо заметно, что лидирует категория Игр.

Таблица 1.3 Лидирующие категории по скачиванием в Apple Store в 1 квартале 2019 года

Ведущие категории по загрузкам в Apple Store в 1 квартале 2019 года	
По количеству загрузок	По доходу
Игры	Игры
Развлечения	Производительность
Фото и видео	Социальные сети
Утилиты	Образование
Стиль жизни	Развлечение

В Apple Store и Google Play спрос обнаружен на категорию игр. За счет скачивания игр рост по сравнению с 2018 на начало 2019 года составил 40% скачиваний и 70% прибыли, если анализировать только рынок Google Play, то категория игр увеличила и гарантировала около 80% прибыли.

Таблица 1.4 Лидирующие категории по скачиванием в Google Play в 1 квартале 2019 года

Ведущие категории по загрузкам в Apple Store в 1 квартале 2019 года	
По количеству загрузок	По доходу
Игры	Коммуникации
Инструменты	Производительность
Развлечения	Социальные сети
Коммуникации	Инструменты
Социальные сети	Производительность

«Скачивания мобильных приложений в таких магазинах как Apple App Store, Google Play, магазин Windows Phone и BlackBerry World увеличили свои показатели на 11% в начале 2018 года, сравнивая с показателями конца 2017года. Но показатели дохода увеличились только на 9%, что доказывает о популярности бесплатных мобильных приложений. Прибыль от платных мобильных приложений выросла на 9%. » - описывает ситуацию на рынке компания Canalys.

Прибыль в денежном эквиваленте равна 13,4 миллиарда, а выручка от продаж мобильных приложений равна 2,2 миллиарда долларов. Apple App Store лидирует и занимает 74% мирового рынка. Далее строчкой ниже оказывается Google Play. Windows Phone Store от Microsoft занимает третье место. BlackBerry делает упор на разработку своего интернет магазина.

### 1.3 Анализ существующих казахских и зарубежных разработок

Развитие IT-технологий и массовое распространение мобильных устройств позволили сформировать новый рынок — рынок мобильных приложений. Мобильные приложения являются одним из главных трендов последних лет в сфере IT.

С каждым днем количество мобильных приложений стремительно растет. Все программные продукты, касающиеся одной сферы и пытающиеся решить одни и те же проблемы, имеют множество сходств, но и множество различий.

По этой причине у одних программных продуктов может быть обширная аудитория заинтересованных пользователей, а другие же продукты, несмотря на схожие функции, могут так и остаться непризнанными.

При разработке мобильного приложения необходимо учитывать потребности пользователя. Поддержание интереса и создание максимальных удобств для пользователя - ставится в приоритете.

Был произведен поиск аналогов данного приложения. В ходе анализа был составлен топ 5 геолокационных приложений. Такой анализ позволит выявить сильные и слабые стороны приложения, какие функции пользуются популярностью и в процессе создания и развития могут быть добавлены.

OsmAnd.OsmAnd — это приложение для путешествий и навигации на базе автономных карт (рисунок 1.6), в котором можно найти местоположение в неопознанном помещении. Создать поиск в карте и проложить маршрут.

Некоторые из основных возможностей приложения:

- работа с автономными картами (векторными и растровыми) без Интернета (для этого нужно предварительно сохранить их);
  - доступность оффлайн карт для всех стран мира;
  - голосовая навигация в автомобильном, велосипедном, пешеходном режимах;
  - отображение дополнительной информации, такой как GPX треки или навигационные маршруты, объекты (POI), избранные местоположения, линии высот, остановки общественного транспорта;
  - наложение дополнительных карт с настраиваемой прозрачностью;
  - поиск адресов и объектов на карте (POI) без доступа в Интернет;
  - прокладка маршрутов на любые расстояния с использованием Интернет сервисов;
  - прокладка маршрутов на средние расстояния без доступа в Интернет;
  - регулярное обновление карт;
  - гибкое планирование маршрута;
  - бесплатный и удобный gps навигатор с самыми подробными картами.
- Ограничения бесплатной версии OsmAnd:

- ограниченное число загрузок карт;
- нет возможности просматривать описания достопримечательностей из Wikipedia без использования Интернет;
- ограниченное число работы приложения (демо 7 дней).

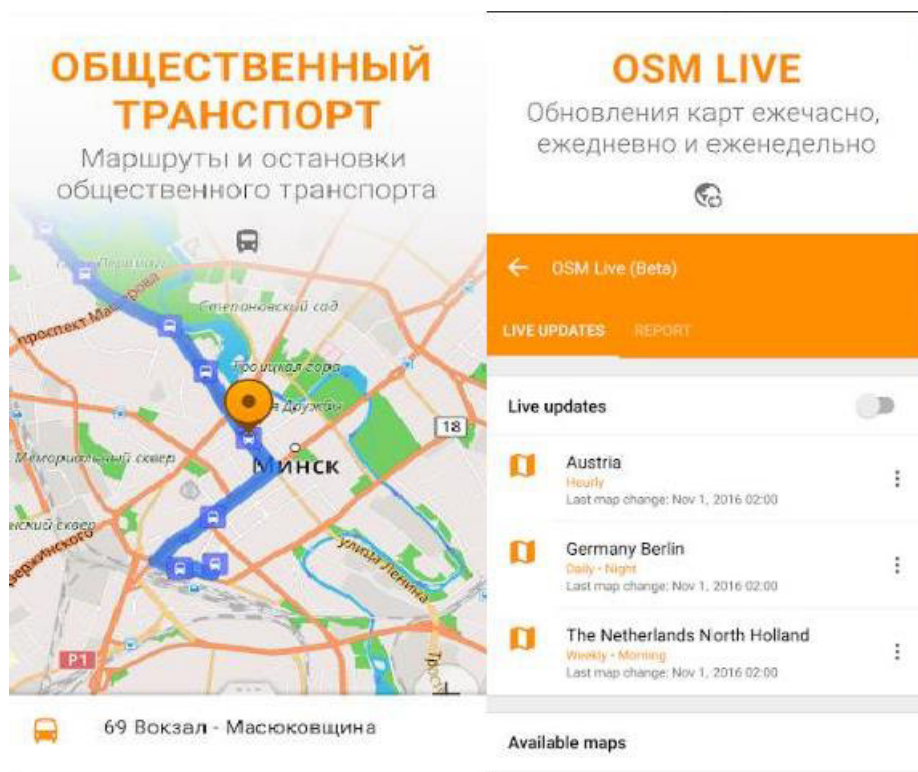
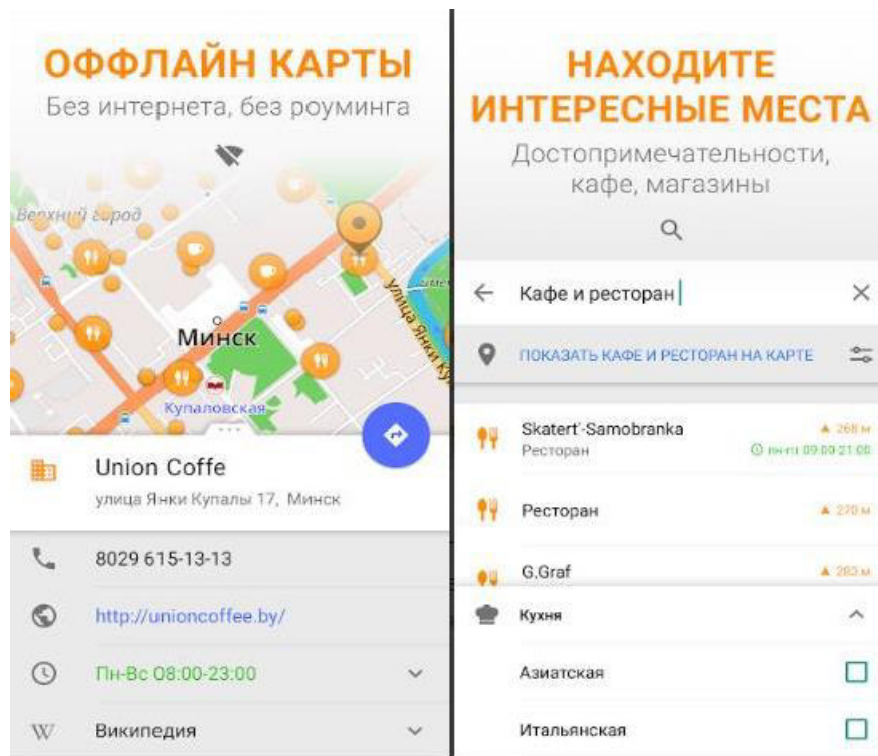


Рисунок 1.6 – Приложение OsmAnd

City Maps 2Go. City Maps 2Go – навигационная программа, которая базируется на системе OSM (Рисунок 1.7). Она предоставляет возможность загружать карты местности в память мобильного устройства и создавать маршруты без доступа к интернету.

Данное приложение превращает смартфон в геолокационный инструмент. Используются карты Google, максимально отображается местоположение пользователя. Для целевой аудитории, использующей данное приложение доступны такие функции как поиск ближайших остановок, помещений, магазинов и так далее.

Программа располагает обширной базой магазинов, кафе и других мест. Кроме того, для работы не нужен интернет. Приложение способно быстро определять местоположение, используя GPS. Помимо этого, приложение позволяет забронировать отель непосредственно из его интерфейса.

Из отрицательных моментов стоит выделить то, что функции автонавигации ограничены, в программе нет русского языка.

Навигатор подойдет больше пешим туристам, о чем можно понять из функционала приложения.

Приложение разработано только на платформе IOS.

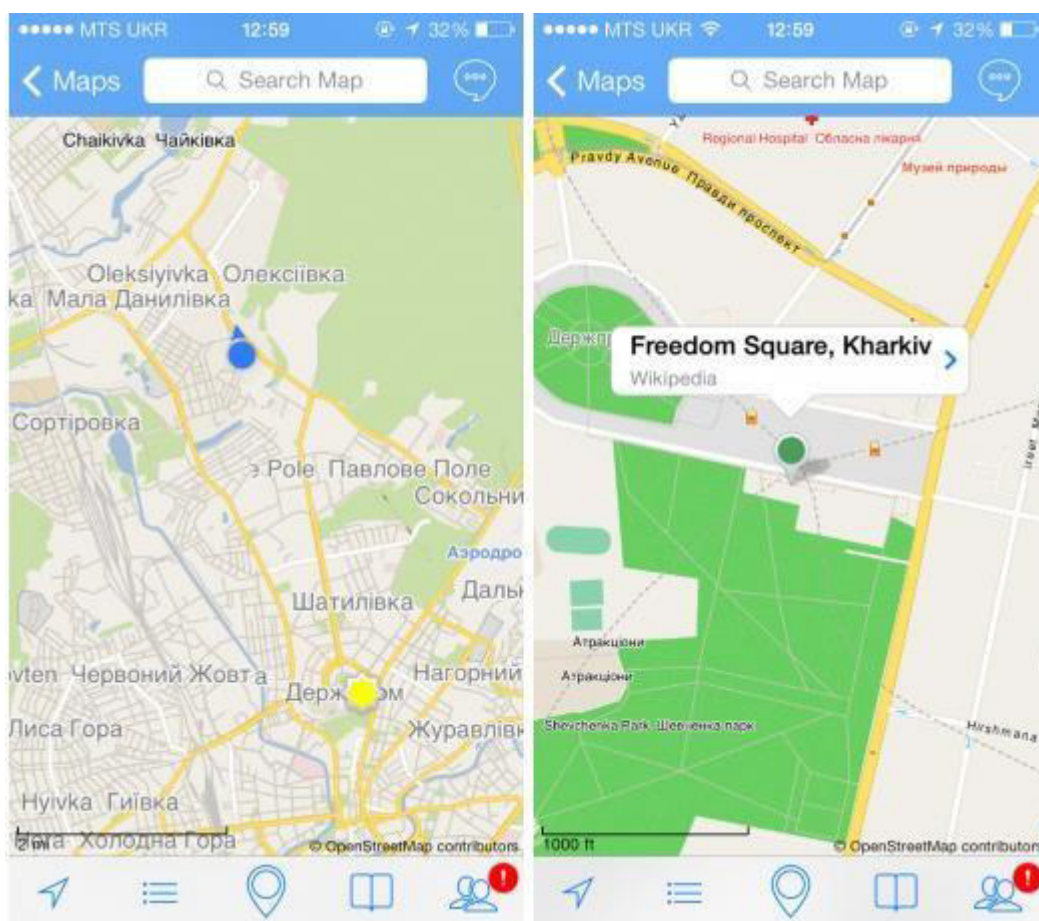


Рисунок 1.6 - City Maps 2Go

Triposo. В данном приложении пользователь может просмотреть рекомендацию о различных достопримечательностях, которые находятся в выбранном городе. То есть он не просто сообщает информацию, а предоставляет обширный отчет об культурных местах города. Приложение работает по данному принципу:

- сканирование веб-сайты с информацией, которая необходима и доступна;
- обработка информации, которая была получена;
- предоставление пользователям идеального маршрута.

Triposo является мировым приложением, которое обладает многими функциями. Пользователь скачивает информацию о выбранном городе или стране перед самой поездке, приложение работает оффлайн режиме. Так же в данном приложении содержится «Travel planner», куда сохраняются ваши чеки из мест, которые посетил пользователь.

Главный недостаток данного приложения в том, что в нем нет русского языка.

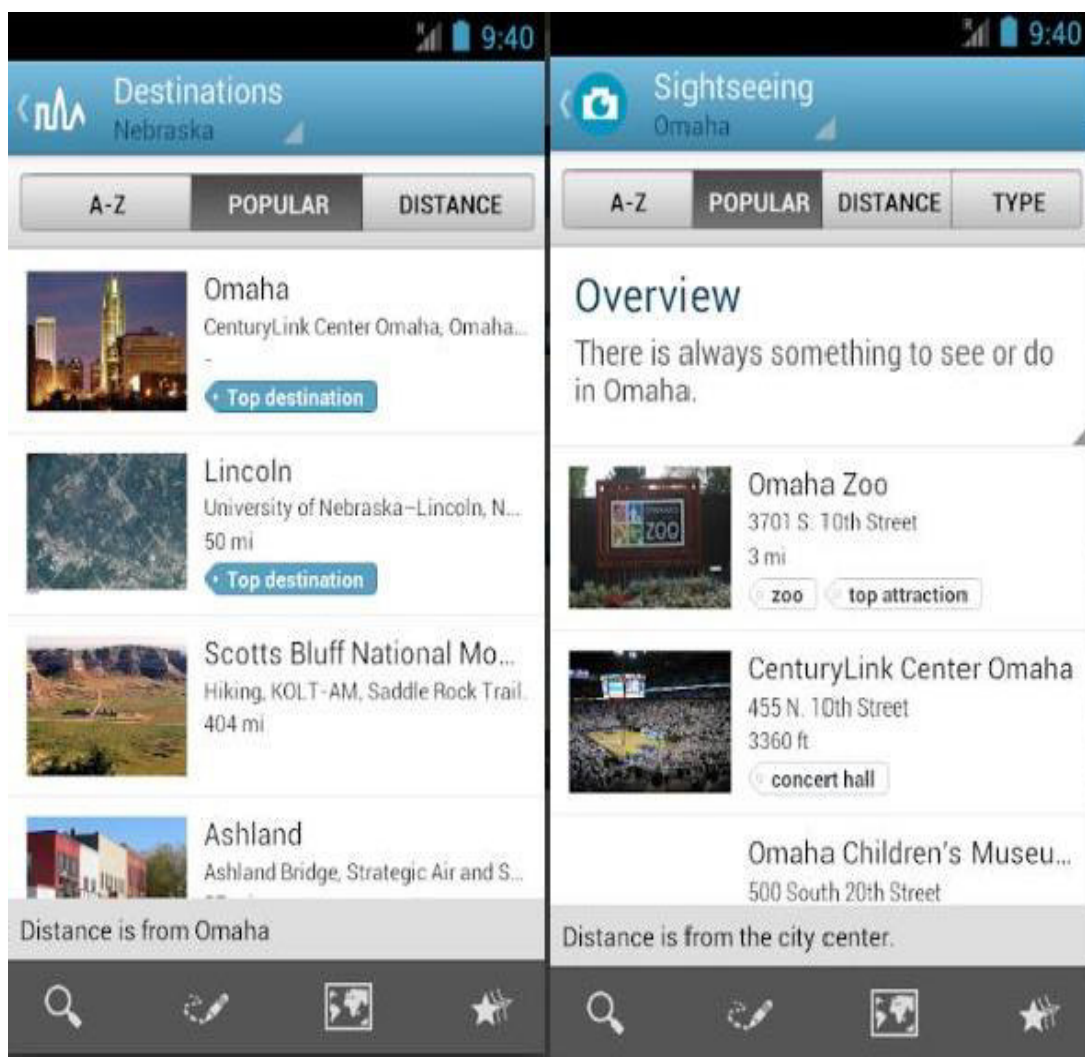


Рисунок 1.7.1 – Triposo



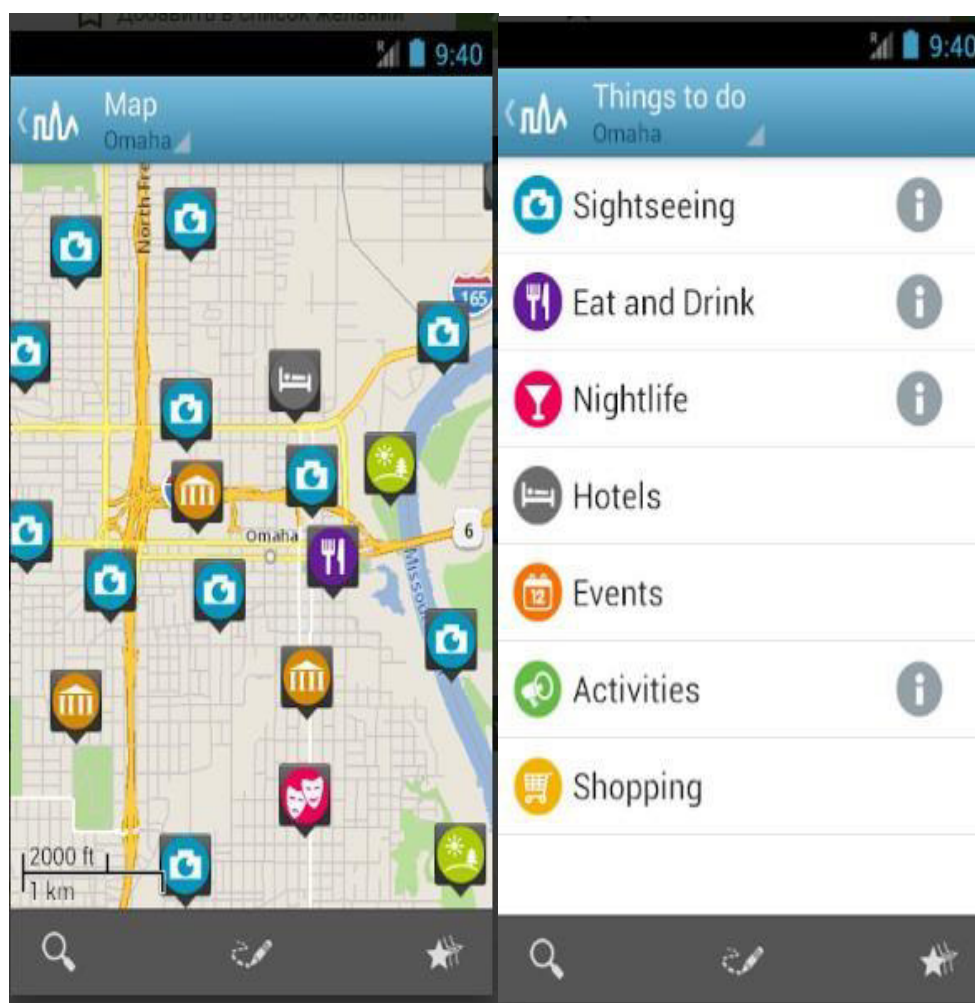


Рисунок 1.7.2 – Triposo 2

#### 1.4 Подведение итогов анализа и постановка задачи

В ходе проведения анализа геолокационных мобильных приложений, можно сказать, что данные приложения очень востребованы в наше время. Минусами уже существующих мобильных приложений являются ряд причин, такие как

- приложение только на английском языке;
- приложение платное;
- карты загружаются на телефон, что занимаем память мобильного устройства.

В ходе анализа транспортной обстановки в городе был сделан вывод, что автобусы, троллейбусы и метро – это основной транспорт алматинцев. Следовательно, данная тема актуальна на данный момент. В связи с этим следует отметить, что для удобства пользования общественным транспортом, жителям города необходимо иметь следующую информацию:

- информация по остановкам и маршрутам;
- поиск остановок и маршрутов;

- поиск объектов на карте;
- направление движения транспорта;
- возможность загрузки карт;
- время приездов и отъездов городского транспорта.

Таким образом, для предоставления данных сведений решением станет мобильное приложение «Умная остановка». Данное приложение будет содержать красивый и функциональный интерфейс, который обеспечит удобство и простоту использования в повседневной жизни обычного человека, пользующегося общественным транспортом.

## 2 Описание и обоснование выбора инструментов разработки

### 2.1 Анализ современных языков программирования

Развитие новых инструментов разработки, использование и создание новых технологий – это то, из чего состоит сфера IT. Условия и характеристика по которым нужно выбирать язык программирования – умения реализовать код и логика в выборе с учетом условия среды разработки и профилем выбора приложения. Так как находятся постоянные недостатки в имеющихся языках программирования и не соответствие одного языке в работе разных сфер разработки, создаются новые языки программирования, либо разработчики берутся за усовершенствования уже имеющегося языка.

Язык программирования – это некие символы и обозначения с помощью которых разработчик описывает алгоритм и структуру данных.

На данный момент каждый год появляется более одного языка программирования. Около 2,500 языков программирования используются на данный момент. В данном подпункте я опишу самый востребованные языки программирования.

Язык Dart, разработанный компанией Google, как один из языков веб-программирования.

Как и в JavaScript, в Dart использования синтаксиса и ключевых слов позаимствованы основой из языка. Ключевое отличие языка C от Dart в том, что, когда JavaScript опирается на прототипы, объекты в Dart образуются с помощью классов и интерфейсов, как в C++ или Java.

Еще язык программирования презентованный компанией Google является язык Go. Язык предусмотрен для веб-разработки, разработки приложений и информационных систем. Язык стабильный, гибкий, мобильно созданный для тестирования.

Язык схож с C или C++, чем с Java или с C#. Go содержит следующее современные функции, а также Очистка памяти с ненужной информацией, отражение текущая деятельность и поддержка параллелизма. В Go style 15 отсутствуют классы, конструкторы и деструкторы. Тем не менее, есть структура взята из C-стиля, который, вероятно, является привязкой

пятнадцать функций с этим стилем Go вы можете создать программу на основе объектно-ориентированного программирования. Присутствие функция для того чтобы освободить память ненужной информации, упрощает обслуживание по сравнению с C++. Интернет-кодирование (для движка, что трудно, какова цель покупателя, сервера, информации о дворе и т. д. собственных индивидуальных стилей программирования. Инженеры создали последнюю речь ОРА. Этот язык формируется с учетом этого, чтобы объединить сокет пользователя, логику сервера и основу ввода-вывода информации. Переводчик без посторонней помощи решает, куда запустить проект и осуществляет созданный шифр.

Сфера исполнения Ора связывает личный интернет-компьютер и концепцию управления основой сведений, какие никак не имеют все шансы являться замещены независимыми кандидатурами. Ора поставляется безвозмездно и в этот период доступен с целью ШЕСТЬДЕСЯТ ЧЕТЫРЕ-х битных Linux и Mac OS X платформ, прочие ведь портки. С иной края в любом стиле имеются собственные недочеты, какие устанавливают основанную в данном стиле концепцию меньше не опасной.

Для постановления данной трудности эксперты обнаружили метод охраны такого рода, то что дает возможность уберечь и интернет-веб-сайты, и ПК юзеров, разработав концепцию позволяющую применять ряд стилей программирования присутствие формирования хвалой проекты.

Данная концепция дает возможность программерам создавать интернет-странички и дополнения, применяя непосредственно все без исключения способности C, PHP, HTML, CSS, JavaScript и иных стилей, и чураясь потребности применения определенных искусственного происхождения способов, какие считаются ключами возможной угрозы.

Данная концепция никак не только одна, образцами аналогичных проектов считаются – Scheme, ProteaJ, Spoofax и OJ. Они кроме того принимать решение перечисленные выше проблемы. В развитие проблемы, был сконструирован новейший язык программирования около наименованием Sketch. Отличительная черта стиля в этом, то что некто дает возможность присутствие написании проектов спускать определенные зоны программного кода, наполняя данные интервалы без помощи других присутствие данном оптимизируя вышедшие системы.

Основной мыслью реализованной в варианте компилятора стиля Sketch, считается способ отыскивания рационального постановления конкретной проблемы.

Известно, точно каждая логичная и точная цель способен являться разрешена некоторыми способами, число каковых, определенных вариантах, способен ст-ре-завести к нескончаемости WolframLanguage считается стилем, опирающимся в большую основу познаний, в каковой заключены все безисключения стандартные вычисляемые процедуры.

Рассмотрев инновационные стили программирования, определенные с каковых вводятся в программирования, определенные ещё доводятся возможно совершить заключения о этом, то что в наше время период, обладая основу с 2-ух с супругой тыс. стилей программирования отсутствует буква 1-го многоцелевого. До тех пор пока подобная цель никак не нужно, по этой причине в данной стадии возможно отметить концепцию Wuyvern, концепция, позволяющую применять ряд различных стилей программирования присутствие формирования одной проекты. Возможно допустить, то что несмотря на то б единственный с осмотренных стилей сменит в перспективе известных в

наше время период стилей программирования

## 2.2 Выбор языка программирования Java

Java был построен с философией "написать один раз, запустить в любом месте" (WORA). Java-код (чистый Java-код и библиотеки), который вы пишете на одной платформе (операционная система), будет работать на других платформах без изменений. Для запуска Java используется абстрактная машина, называемая Java Virtual Machine (JVM). JVM выполняет байт-код Java. Затем процессор выполняет JVM. Поскольку все JVMs работают одинаково, один и тот же код работает и на других операционных системах, делая Java независимой от платформы.

Объектно-ориентированный язык

Существуют различные стили программирования. Объектно-ориентированный подход является одним из популярных стилей программирования. В объектно-ориентированном программировании сложная задача разделяется на меньшие множества путем создания объектов. Это делает ваш код многократно, имеет преимущества дизайна и упрощает обслуживание кода.

Многие языки программирования, включая Java, Python и C++ в объектно-ориентированных возможностях. Если вы серьезно относитесь к программированию, вы обязательно должны изучить объектно-ориентированный стиль программирования.

Более ранние версии Java были подвергнуты критике за медлительность. Но сейчас, без исключения, все совершенно по-другому. Самые последние обычные машины гораздо быстрее. И микропроцессор, реализующий JVM, также становясь без исключения самыми сильными. Теперь, Java считается одним из самых быстрых языков программирования. Хороший оптимальный Java-шифр таким образом, потому что быстрый, как стили самой низкой степени, такие как C/ C++, и намного быстрее, чем Python, PHP и т. Д .

Java 8 считается безопасным

Платформа Java предоставляет множество функций для обеспечения

Безопасность надстройки Java. В значительной степени определяется из 8 функций,

какие процессы Java:

- обеспечивает безопасную платформу с целью изучения и начала

Приложение механическое управление памятью, уменьшает помеху памяти

и уязвимости- обеспечивает безопасную связь, защищая целостность и конфиденциальность передаваемых данных

Большая Стандартная Библиотека

Одной из причин широкого использования Java является наличие огромной стандартной библиотеки. Среда Java имеет сотни классов и методов

под различными пакетами, чтобы помочь разработчикам программного обеспечения, как мы. Например,

Java.lang-для расширенных функций строк, массивов и т. д.

Java.util-для структур данных, регулярных выражений, функций даты и времени и т. д.

java.io - для ввода-вывода файлов, обработки исключений и т. д.

Приложения Java

Терминология Java, которую вы должны знать, прежде чем начать

Исследуйте последний стиль программирования способен быть сложной задачей.

Вы услышите большое количество новейших определений, которые имеют все шансы быть подавляющим для новичка. Таким образом, я решил уточнить ряд определений, что вы можете услышать в обществе стиля программирования Java.

Обратите внимание, что это не весь список из 9 глоссариев Java. Он список определений, которые вы должны знать, 9пр основе изучения программирование на языке Java.на основе объектно-ориентированного программирования. Присутствие

функция очистки памятиОбратите внимание, что это не полный список глоссария Java. Это список терминов, которые вы должны знать перед началом изучения программирования на Java.

Согласно Java-Java - это набор технологий (речевое программирование и компьютерная платформа) с целью формирования и запуска программного обеспечения обеспечение. Но Java часто используется для обозначения языка 8 8

Программирование Java с целью простоты.

Язык программирования Java-сильный, многофункциональный, независимый от платформа, объектно-ориентированное речевое программирование.

Java 8-Java 8 считается окончательным основным выпуском для Java. Наш учебник Java будет содержать все ключевые функции Java 8 без исключения.

Java ee приложения и Java на SE-Java ee приложения и Java SE8 стенды для Java

Платформа Enterprise Edition, Micro Edition и 8 standard edition в соответствии с этим.

Java EE предназначен для надстроек, 8 из которых функционируют на серверах.

Java ME предназначен для устройств с узкими ресурсами, таких как встроенный прибор. И Java SE-8-это 8 базовая сфера Java, используемая для создание типовых проектов.Если вы новичок в программировании на java, мы рекомендуем вам начать с J2SE.

Виртуальной Java - машины JVM (виртуальная машина Java) является абстрактной машиной, которая позволяет вашему компьютеру для запуска программы Java.

JRE-JRE (среда выполнения Java) содержит JVM, поддерживающие библиотеки и другие компоненты для запуска программы Java. Однако, он не содержит компилятора и отладчика.

JDK-JDK (Java Development Kit) содержит JRE и такие инструменты, как компиляторы и отладчики для разработки приложений Java.

Одно из самых основных преимуществ Java, что данный продукт востребован в течении двадцати лет. На данный момент язык Java занимает 5 место в сравнениях с другими актуальными языками. Java, JavaScript, HTML, CSS и SQL после данных информационных технологий идет Java.

Java—это язык программирования общего назначения, который следует парадигме ООП и подход «Написать один раз и использовать везде» . Java используется для настольных, сетевых, мобильных и корпоративных приложений. Подробная информация:

- загрузки Java;
- документация Java;
- сообщество Oracle Java;
- javaWorld—популярный портал о Java;
- javaRanch—форум для Java-разработчиков.

Java—это не только язык программирования, но и экосистема инструментов, охватывающая почти все, что может понадобиться при программировании на Java. В нее входят:

– java Development Kit (JDK)—комплект разработчика Java. С помощью JDK и стандартного блокнота можно писать и запускать/компилировать код на Java;

– java Runtime Environment (JRE)—исполняющая система Java. Механизм распространения программного обеспечения, состоит из автономной виртуальной машины Java, стандартной библиотеки Java (Java Class Library) и инструментов настройки;

– integrated Development Environment (IDE)—интегрированная среда разработки. Инструменты, которые помогают запускать, редактировать и компилировать код. Самые популярные из них—IntelliJ IDEA, «Eclipse» и «NetBeans».

Java используется во многих средах разработки. Используется для веб-разработки, технологиями, где обрабатывается большой объем информации, таких как Hadoop и Apache Storm. Java язык можно использовать в любых стилистических проектах. До того как появились многофункциональные мобильные устройства(смартфоны) Java использовался и для разработки и программирования в телефонах. Первые мобильные игры были написаны именно на этом языке программирования.

На рисунке показана диаграмма активности использования языков программирования. При разработанных новых языках программирования Java по-прежнему остается на лидирующих позициях всей IT индустрии. (Рисунок 2.1).

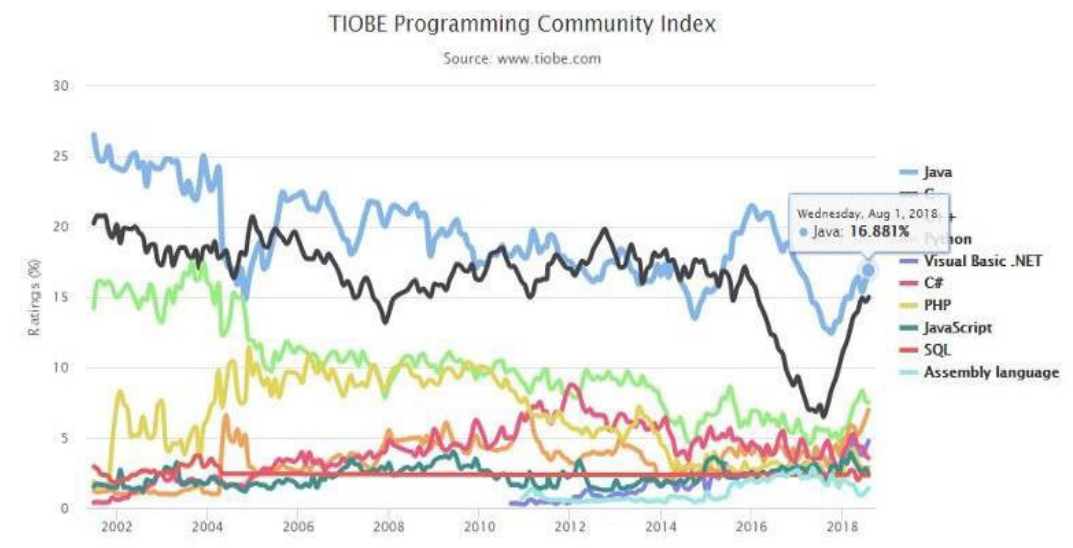


Рисунок 2.1 – диаграмма сравнения использования языков программирования

#### Преимущества Java :

**Гибкость.** Java доказала, что C—процедурный, управляемый вручную и зависящий от платформы код—это не предел совершенства . Благодаря Java, все больше людей начали применять объектно-ориентированное программирование, которое сейчас используется повсеместно.

**Апплеты.** Еще до появления JavaScript, в Java добавили апплеты—небольшие веб-программы, которые предоставляют интерактивные элементы для визуализации и обучения. Они не используются ни для чего, кроме простой анимации, однако апплеты привлекли внимание многих программистов и подтолкнули их к разработке HTML5, Flash и JavaScript.

**Разработка через тестирование.** Java TDD—уже давно не экспериментальная практика, а стандартный способ разработки программного обеспечения. Введение JUnit в 2000 году считается одним из самых больших достижений Java.

#### Плюсы программирования на Java

Java—уже не единственный официально поддерживаемый язык для разработки на Android. Java далеко не единственный выбор в веб-программировании. Тем не менее, Java идет в ногу со временем. Давайте рассмотрим, какие преимущества предлагает Java.

**Объектно-ориентированное программирование.** Java включает в себя объектно-ориентированное программирование (ООП)—концепцию, в которой вы не только определяете тип данных и его структуру, но и набор



функций, применяемых к нему. Таким образом, структура данных становится объектом, которым можно управлять для создания отношений между различными объектами (Рисунок 2.2).

При другом подходе—процедурном программировании—нужно следовать четким инструкциям, использовать переменные и функции. При ООП можно группировать эти переменные и функции посредством контекста, маркировать их и ссылаться на функции в контексте каждого конкретного объекта.

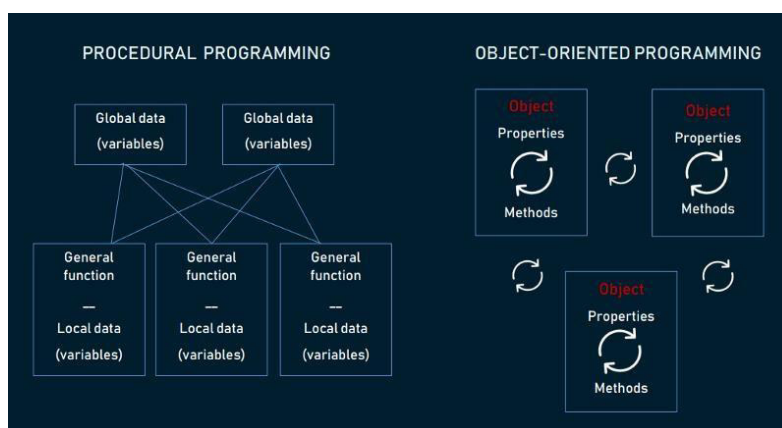


Рисунок 2.2 - Сравнение процедурного и объектно-ориентированного программирования

Основываясь на информации победивших больше Java является единственным из самых практичных и не опасных стилей программирования. Эта речь считается многопоточной, создателям Java не нужен ручной способ муравей. автоматический шифр записи с целью управления памятью благодаря механическому управлению памятью (АММ), кому-то принадлежит интегрированная система для общего применения информации и проектов некоторыми ПК, что-то, что повышает эффективность и результативность работы. Данная речь подходит для изучения мобильных дополнений к посадочной площадке робота.

Исследования окружающей среды об этом мобильное приложение существует, специально подобранных для Android-студия.

Исследовательская среда. Android Studio-новейший и полностью интегрированный объем исследовательских дополнений, не так давно опубликованных Google с целью операторской концепции Android. Это производство призвано обеспечить создателей новейшими устройствами с целью формирования приложений

Когда вы создаете последний план в Android Studio, план представлен абсолютно всеми файлами, содержащимися в каталоге SDK. Это преобразование в концепцию управления Gradle дает ход исследования еще большую эластичность.

В Android Studio можно следить за разнообразием визуальных преобразований, которые я изменяю в этот период в приложении. Приложение можно проверить в то же время в разных устройствах рядом с Android платы, с различными вариантами и разрешением экрана. В добавках в целом, последняя область исследований обладает целью Google Cloud Messaging, который дает вам возможность отправлять информацию с сервера на устройство Android из-за скопления. Это хороший способ отправить push-сообщение на ваш применение.. Главные преимущества Android Studio:

- надежная и простая среда разработки;
- просто контролировать эффективность дополнения в разных видах устройство;
- помощники и стандарты для целей общих программных компонентов Андроид;
- полнофункциональный поток с большим количеством дополнительных инструменты для ускорения изучения добавок. Использование библиотек в данном мобильном приложении:
- appcompat-v7;
- constraint-layout;
- junit;
- runner;
- espresso.

Библиотека Appcompat. На данный момент в Android Studio создаются проекты на основе библиотеки AppCompat. Благодаря данной библиотеке используются такие возможности как ActionBar, материальный дизайн, Toolbar и т.д.

Прежде всего необходимо добавить зависимость в модуле проекта в файле build.gradle.

```
dependencies {  
    compile 'com.android.support:appcompat-v7:23.0.0'  
}
```

Библиотека содержит в себе зависимость от библиотеки support-v4.

Рисунок 2.2.1 – Код из файла build.gradle

В данном программном продукте используется последняя версия SDK:

```
android {  
    targetSdkVersion 28.0.3  
}
```

Рисунок 2.2.2 – Версия SDK

Диалоговые окна AlertDialog поддерживает материальный дизайн.

В линейке Android 4.x использовалась тема Holo. Для перехода на материальный дизайн: `android:Theme.Holo.Light` на `Theme.AppCompat`.

Библиотека `constraint-layout`. Новейшая модель `ConstraintLayout` возникла в Android Studio 2.2 и доступен для устройств с версией Android 2.3. Это может быть найдите в области макеты. Два Если этот элемент выбран в порядке оформления, то в этом случае в панели ряд кнопок легко доступны для вас -2 точки параметров просмотра показывают ограничения-вводит установленные ограничения в режимы предварительного просмотра и компоновки. В 1 вариантов режим просмотра может быть полезен, а не в других. Наличие огромного количества ограничения этот ключ вводит очень большой объем данных, 2 показать Поля, исчезают невыбранные виды.

Turn On Autoconnect-наличие введения порядка ограничения Autoconnect будет автоматически настраиваться на наличие видов буксира область предварительного просмотра. Семинар стремится угадать, что ограничения должны обладать элементом и формировать их в соответствии с потребностями. Два поля по умолчанию-это нормальная роль для отступления. Можно установить отдельно для любого элемента. Предпочтительный элемент, установите роль, затем снова отдайте предпочтение другому элементу и установите другая роль. Очистить все ограничения-устраняет все ограничения из макета без исключения.

Вывод ограничения-автоматически2 порывыБиблиотека `junit`. Данная библиотека создана для модульного тестирования программного обеспечения на языке Java. Библиотека во многом упрощает и автоматизирует процесс написания тестов.

Библиотека `runner`. Данная библиотека также создана для создания модульных тестов.

Библиотека `espresso`. Правила деятельности библиотеки: эспресо имеет несколько ключевых классов для тестирования.

Эспрессо-основная группа. Включает постоянные способы, такие как как нажать целые клавиши (13 назад, домой), позвонить / скрыть клавиатуру, откройте список, перейдите к компоненту.

Просмотр совпадений-позволяет найти элемент на экране иерархию.

Просмотр действий - позволяет взаимодействовать с компонентом (нажмите, длинное щелчок, двойной щелчок, Размах, прокрутка и т. д.).

Просмотр утверждений-позволяет принять позицию элемента.

Эспрессо-это фреймворк для целей тестирования. Были использованы ранее

однако в качестве единой библиотеки, включая выпуск версии 2.0, Espresso является начальной частью

репозиторий поддержка Android репозиторий, что позволило быстро и удобно добавьте через исследования в план `wasinski`. Функции в приборах и эмулятор. Писать тесты на Espresso очень удобно, так как много функционала она берёт на себя. Например, не нужно делать фокус на кнопке,

проверять доступна ли она на экране и т.д., достаточно обратиться к кнопке по её идентификатору или text/description и Espresso уже всё сделает за вас, если кнопка не видна или не доступна на экране то тест автоматом уже не пройдёт.

Espresso использует синхронизацию с основным потоком Main Thread.

Основные компоненты Android приложения.

Дополнения для Android написаны в стиле Java программирования.

Инструменты Android SDK (комплект для разработки программного обеспечения-development kit программное обеспечение) составьте 12 написанных вами шифров — и все требуют комп. информация и данные ресурсов-в файле APK 7-пакет прошивки Android, о чем говорит сам, архив документов с расширением .7 apk . В вAPK файле есть все без исключения, что необходимо для целей 7 Android-дополнений, а также она дает возможность определить дополнение в каждом приборе около управления. Система Android-дополнение, определенное в функциях инструмента в собственной " песочница "(разделенная программным сектором):

операционная концепция Android включает в себя многопользовательская концепция Linux, как и любое приложение является одним пользователем;

по умолчанию концепция устанавливает уникальное дополнение к любому

ID пользователя Linux (этот личный номер используется только система и незнакомое дополнение); концепция определяет возможности для все файлы кроме того, с этим, чтобы разрешить доступ к ним был действителен только пользователь с персональным номером, выделенным этому дополнению;

на любой скорости есть личный, условный автомобиль (VM), так что же такое шифр добавок отдельно от других добавок; по умолчанию любая надстройка выполняется самостоятельно

Процессы в Linux. Android позволяет процедуру, если вам нужно реализовать тот или иной-или элемент дополнения, а затем завершает процедуру, если кто-то больше не необходимо или если концепция необходима, чтобы избавиться от воспоминаний с целью других приложения. Таким образом, концепция Android будет выполнять правила предоставления

минимальная ярмарка. В этом случае любая надстройка имеет доступ по умолчанию

только к этим компонентам, что ему нужно для цели деятельности, и письмо к чему другой. В результате этого, только не-опасная сфера создана, внутри

— какое дополнение не имеет доступа к запрещенным областям концепции. Это не что иное, как добавки имеет свои собственные личные типы предоставление информации другие дополнения и допуск к абсолютно всем предложениям: два дополнения можно установить уникальный идентификатор пользователь Linux. В этом случае 7 каждый со

своим курортом smozhet kfile другие дополнения.любой ход имеет свой виртуальный автомобиль (VM), так что выполняется заполнение шифра изолированы от других дополнений; по умолчанию любая надстройка выполняется самостоятельно . Процессы в Linux. Android позволяет процедуру, если вам нужно реализовать тот или иной элемент дополнения, а затем завершает процедуру, если кто-то больше не необходимо или если концепция необходима, чтобы избавиться от воспоминаний с целью других применение.таким образом, концепция Android будет выполнять правила предоставления минимальный ярмарка. В этом случае любая надстройка имеет доступ по умолчанию только до этих компонентов, что ему нужно для цели деятельности, и письмо к чему другой. В результате этого, только не опасная сфера создана, внутри какое дополнение не имеет доступа к запрещенным областей концепции. Тем не менее, приложение имеет свои собственные типы данных другие дополнения и доступ ко всем услугам:два дополнения можно задать уникальный идентификатор пользователь Linux. В этом случае, любой из них сможет прибегнуть к файлы других дополнений. Для того чтобы сохранить ресурсы, концепция может также быть сделайте это для того, чтобы дополнение с тем же персональным номером пользователя были выполнены в 1 курсе Linux и использовали 1 VM (приложения также должен быть подмахнул один сертификат); приложение может сделать запрос на допуск данных устройство, например контактам пользователя, SMS-оповещение, подключена карта памяти (SD-карта), камера, Bluetooth и Т. д. Все разрешение на добавление должно быть дано при наличии его установки. Это ключевые данные, каким образом Android add есть такое понятие. В других частях этого важного документа раскрываются следующая проблема: основные элементы, устанавливающие надстройку; файл манифеста, в котором объявляются элементы и функции устройства, необходимые для дополнения; ресурсы, которые доступны отдельно с надстройкой кода и позволяет адаптировать услуги к устройствам разная конфигурация. Компоненты приложения. Элементы дополнения являются базовыми, скаковая лошадь состоит приложение с целью Android. Любой элемент подразумевает отдельное место, с помощью которого понятие способно входить в дополнение. Не все элементы считаются точками входа для целей пользователя, а некоторые из них зависимы товарищ с товарищем. Наличие этого любого элемента является самостоятельная скелетная вещь и имеет особое значение —

каждый из них включает в себя уникальный компонент текстуры, который устанавливает надстройку на полную.

Компоненты могут быть отнесены к 1 из 4 типов.

Компоненты любого рода предоставляются для целей конкретной миссии, они имеют

собственный фактический оборот, который устанавливает способ формирования и

прекращение жизни элемента. Девять Операции. Операция (Activity) представляет собой один экран с пользовательским интерфейсом. Например, в приложении для работы с электронной почтой одна операция может служить для отображения списка новых сообщений, другая – для составления сообщения и третья операция – для чтения сообщений. Несмотря на то что операции совместно формируют связное взаимодействие пользователя с приложением по работе с электронной почтой, каждая из них не зависит от других операций. Любые из этих операций могут быть запущены другим приложением (если это позволяет приложение по работе с электронной почтой). Например, приложение для камеры может запустить операцию в приложении по работе с электронной почтой, которая составляет новое сообщение, чтобы пользователь мог отослать фотографию

Операция относится к подклассу класса Activity. Подробные сведения об этом можно найти в руководстве для разработчиков в статье Операции.

Службы. Служба (Service) представляет собой компонент, который работает в фоновом режиме и выполняет длительные операции, связанные с работой удаленных процессов. Служба не имеет пользовательского интерфейса. Например, она может воспроизводить музыку в фоновом режиме, пока пользователь работает в другом приложении, или же она может получать данные по сети, не блокируя взаимодействие пользователя с операцией. Служба может быть запущена другим компонентом, который затем будет взаимодействовать с ней, – например операцией.

Служба относится к подклассу класса Service. Подробные сведения об этом можно найти в руководстве для разработчиков в статье Службы.

Поставщики контента. Поставщик контента (Content provider) управляет общим набором данных приложения. Данные можно хранить в файловой системе, базе данных SQLite, в Интернете или любом другом постоянном месте хранения, к которому у вашего приложения имеется доступ. Посредством поставщика контента другие приложения могут запрашивать или даже изменять данные (если поставщик контента позволяет делать это). Например, в системе Android есть поставщик контента, который управляет информацией контактов пользователя. Любое приложение, получившее соответствующие разрешения, может запросить часть этого поставщика контента (например ContactsContract.Data), для чтения и записи сведений об определенном человеке.

Поставщики контента также используются для чтения и записи данных, доступ к которым внешним компонентам приложение не предоставляет. Например, в образце приложения Note Padc помощью поставщика контента выполняется сохранение заметок.

Поставщик контента относится к подклассу класса `ContentProvider`. Он должен реализовывать стандартный набор API-интерфейсов, с помощью которых другие приложения будут выполнять транзакции. Подробные сведения можно найти в руководстве для разработчиков в статье Поставщики контента.

Приемники широковещательных сообщений. Приемник широковещательных сообщений (`Broadcast receiver`) представляет собой компонент, который реагирует на объявления распространяемые по всей системе. Многие из этих объявлений рассылает система — например объявление о том, что экран выключился, аккумулятор разряжен или был сделан фотоснимок. Объявления также могут рассылаться приложениями, — например, чтобы сообщить другим приложениям о том, что какие-то данные были загружены на устройство и теперь готовы для использования. Несмотря на то, что приемники широковещательных сообщений не имеют пользовательского интерфейса, они могут создавать уведомления в строке состояния, чтобы предупредить пользователя о событии "рассылка объявления". Однако чаще всего они являются просто "шлюзом" для других компонентов и предназначены для выполнения минимального объема работы. Например, они могут инициировать выполнение службой определенных действий при возникновении события.

Приемник широковещательных сообщений относится к подклассу класса `BroadcastReceiver`, а каждое такое сообщение предоставляется как объект `Intent`. Подробные сведения изложены в руководстве, посвященном классу `BroadcastReceiver`.

Когда система запускает компонент, она запускает процесс для этого приложения (если он еще не был запущен) и создает экземпляры классов, которые требуются этому компоненту. Например, если ваше приложение запустит операцию фотографирования в приложении для камеры, эта операция будет выполняться в процессе, который относится к этому стороннему приложению, а не в процессе вашего приложения. Поэтому, в отличие от приложений для большинства других систем, в приложениях для Android отсутствует единая точка входа (например, в них нет функции `main()`).

Поскольку система выполняет каждое приложение в отдельном процессе с такими правами доступа к файлам, которые ограничивают доступ в другие приложения, ваше приложение не может напрямую вызвать компонент из другого приложения. Это может сделать сама система Android. Поэтому, чтобы вызвать компонент в другом приложении, необходимо сообщить системе о своем намерении (`Intent`) запустить определенный компонент. После этого система активирует для вас этот компонент.

Активность компонентов. Компоненты трех из четырех возможных типов — операции, службы и приемники широковещательных сообщений — активируются асинхронным сообщением, которое называется `Intent` (намерение). Объекты `Intent` связывают друг с другом отдельные компоненты во время выполнения, будь то это компоненты вашего или стороннего

приложения (эти объекты Intent можно представить себе в виде мессенджеров, которые посылают другим компонентам запрос на выполнение действий).

Объект Intent создается с помощью объекта Intent, который описывает запрос на активацию либо конкретного компонента, либо компонента конкретного типа — соответственно, намерение Intent может быть явным или неявным.

Для операций и служб Объект Intent определяет действие, которое требуется выполнить (например, просмотреть (view) или отправить (send) что-то), а также может указывать URI (Uniform Resource Identifier — унифицированный идентификатор ресурса) данных, с которыми это действие нужно выполнить (помимо прочих сведений, которые нужно знать запускаемому компоненту). Например, объект Intent может передавать запрос на выполнение операции "показать изображение" или "открыть веб-страницу". В некоторых ситуациях операцию можно запустить, чтобы получить результат. В этом случае операция возвращает результат также в виде объекта Intent (например, можно отправить сообщение Intent, чтобы дать пользователю возможность выбрать контакт и вернуть его вам — в ответном сообщении Intent будет содержаться URI, указывающий на выбранный контакт).

Для приемников широковещательных сообщений Intent просто определяет передаваемое объявление (например, широковещательное сообщение о низком уровне заряда аккумулятора содержит только строку "аккумулятор разряжен").

Компоненты четвертого типа — поставщики контента — сообщениями Intent не активируются. Они активируются по запросу от ContentResolver. Процедура определения контента (content resolver) обрабатывает все прямые транзакции с поставщиком контента, с тем чтобы этого не пришлось делать компоненту, который выполняет транзакции с поставщиком. Вместо этого он вызывает методы для объекта ContentResolver. Это формирует слой, абстрагирующий (в целях безопасности) поставщика контента от компонента, запрашивающего информацию.

Для активации компонентов каждого типа имеются отдельные методы:

- можно запустить операцию (или определить для нее какое-то новое действие), передав объект Intent методу start Activity()start или ActivityForResult() или (если требуется, чтобы операция вернула результат).

- можно запустить службу (либо выдать работающей службе новые инструкции), передав объект Intent методу startService(). Либо можно установить привязку к службе, передав объект Intent методу bindService().

- можно инициировать рассылку сообщений, передав объект Intent таким методам, как sendBroadcast(),sendOrderedBroadcast() и sendStickyBroadcast().

- можно выполнить запрос к поставщику контента, вызвав метод query()для объектаContentResolver.



Подробные сведения об использовании объектов Intent приведены в документе Объекты Intent и фильтры объектов Intent. Более подробная информация об активации определенных компонентов также приведена в следующих документах: Операции, Службы, BroadcastReceiver и поставщики контента.

Файл манифеста. Для запуска компонента приложения системе Android необходимо знать, что компонент существует. Для этого она читает файл AndroidManifest.xml приложения (файл манифеста). В этом файле, который должен находиться в корневой папке приложения, должны быть объявлены все компоненты приложения.

Помимо объявления компонентов приложения, манифест служит и для других целей, среди которых:

- указание всех полномочий пользователя, которые требуются приложению, например разрешения на доступ в Интернет или на чтение контактов пользователя;

- объявление минимального уровня API, требуемого приложению, с учетом того, какие API-интерфейсы оно использует;

- объявление аппаратных и программных функций, которые нужны приложению или используются им, например камеры, службы Bluetooth или сенсорного экрана;

- указание библиотек API, с которыми необходимо связать приложение (отличные от API-интерфейсов платформы Android), например библиотеки Google Maps и многое другое.

## 3 Программная реализация

### 3.1 Проектирование информационной системы

Начнем с блок-схемы, которая включает в себя графическое представление определения, анализа и метода решения задачи, в котором используются символы для отображения данных, потока, оборудования. В блоках дается описание соответствующего действия описании и изображении алгоритмов данного программного продукта.

На рисунке 3.1.1 представлена блок-схема, отображающая полностью работу данного программного продукта, представляющая циклы, выборы данных, работу приложения от начала до конца.

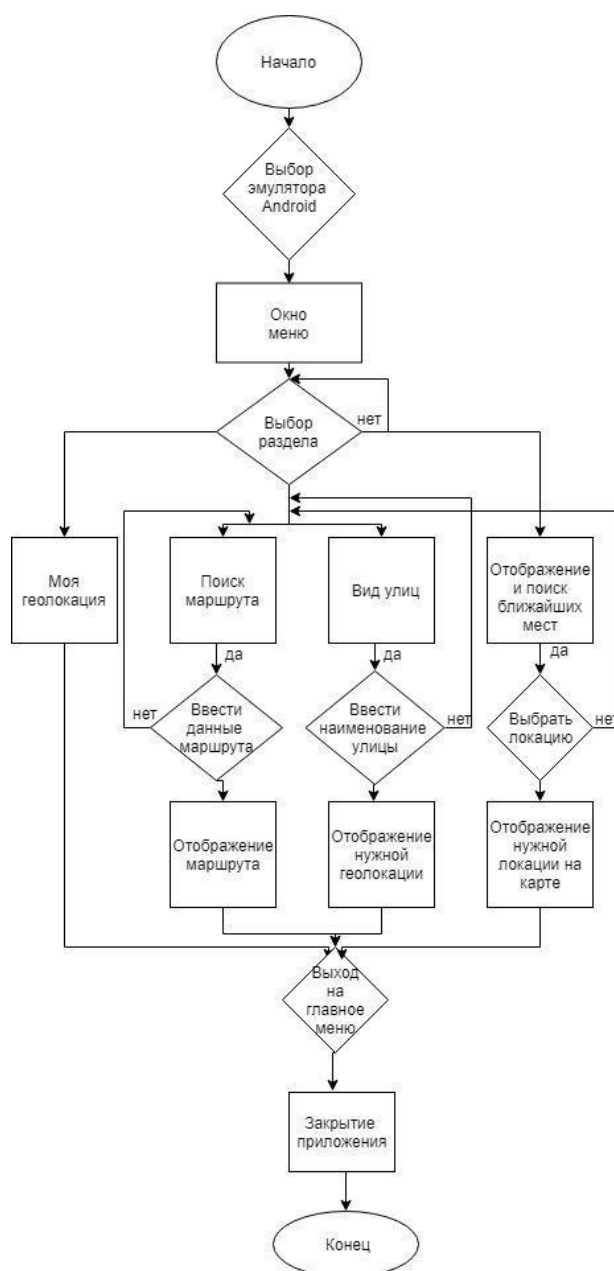


Рисунок 3.1.1 - Блок-схема мобильного приложения

На рисунке 3.1.2 представлена диаграмма представления требования пользователя. На данной диаграмме поставлены задачи пользователя и разработчика, имеется два исхода событий положительный и отрицательный и после, исходя от выбора событий предоставляется вывод.

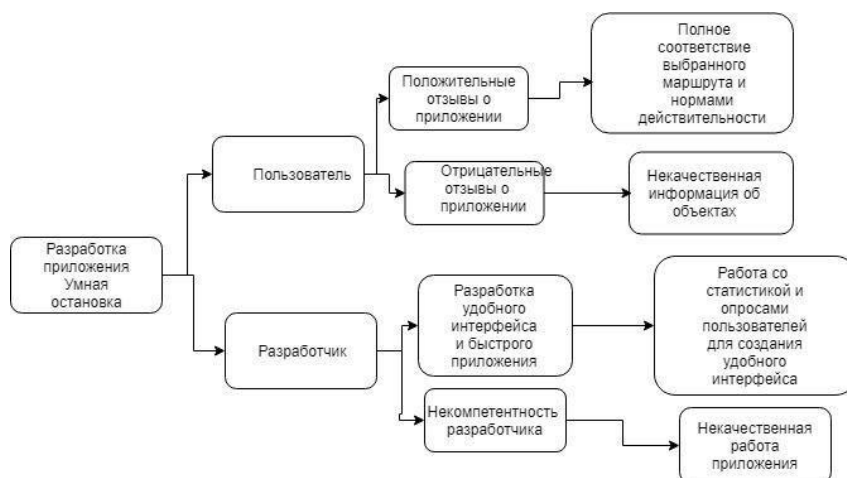


Рисунок 3.1.2 - Диаграмма представления требования пользователя

Далее представлена диаграммы компоненты, которая показывает, как выглядит модель системы на физическом уровне. На диаграмме изображены компоненты программного обеспечения и связи между ними.

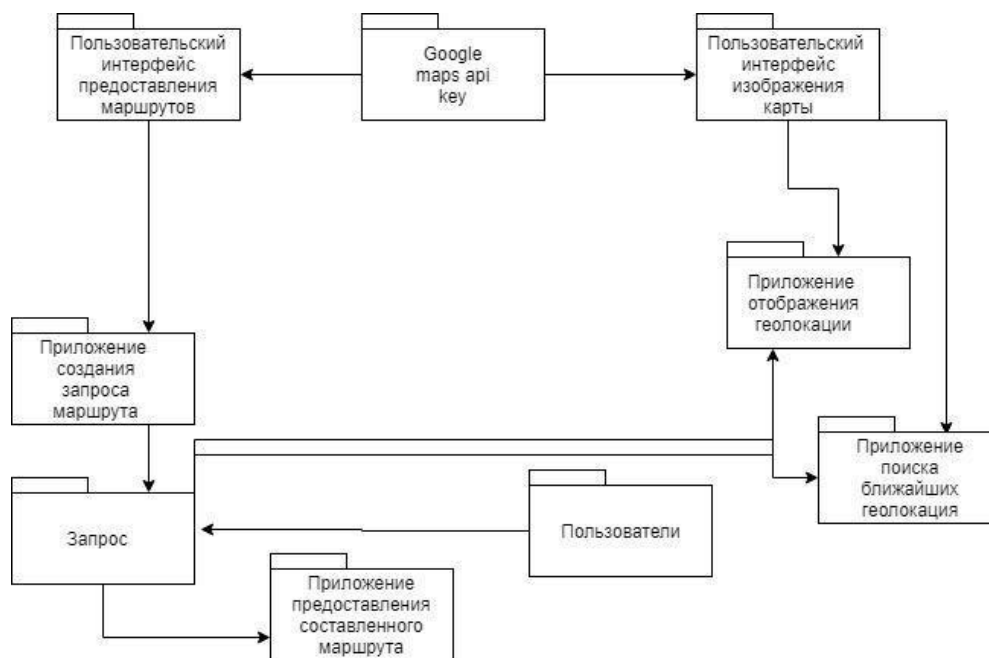


Рисунок 3.1.3 – Диаграмма компонентов

Данное мобильное приложение должно предоставлять следующие функциональные возможности для пользователя:

- вход в приложение(отображение основного меню;

- отправление запроса и получение информации о ближайших нужных местах;
- отправление запроса и получение информации о своем местоположении;
- отправления запроса и получении информации о нахождении маршрута от точки А до точки Б;
- отправления запроса и нахождении на карте необходимой улицы.

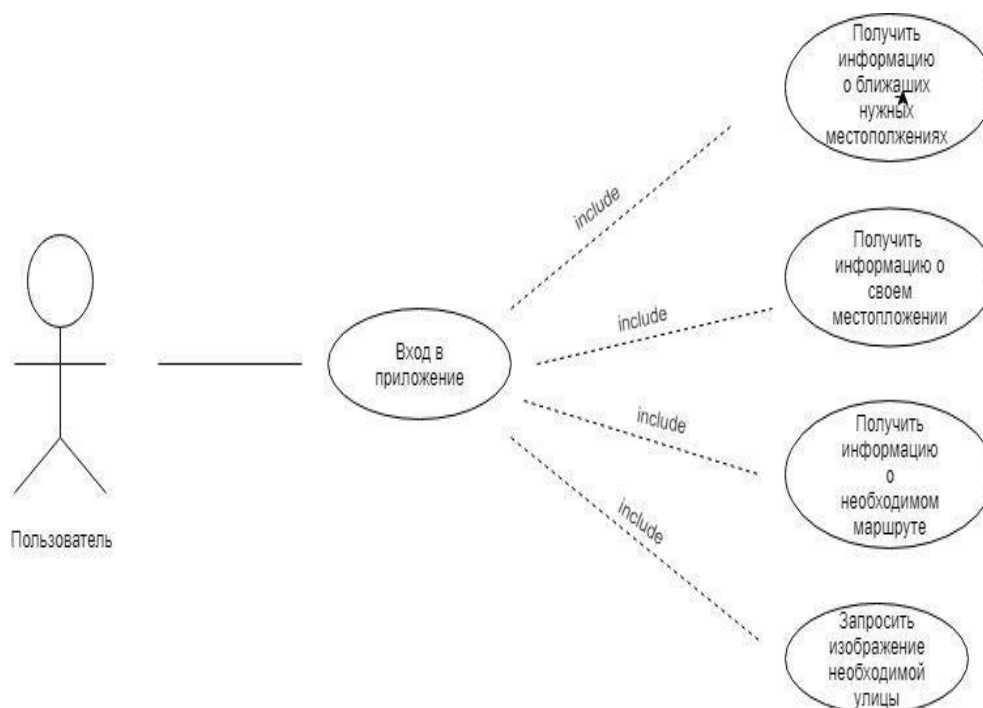


Рисунок 3.1.4 – Диаграмма прецедентов для пользователя

Диаграмма прецедентов для разработчика включает в себе также технологию определения функциональных требований к системе.

- система наделяет разработчика следующими возможностями;
- изменение данных( под значением данных имеется в виду;
- геолокационные, изменение интерфейса, изменение маркеров в проложении маршрута, изменение координат, названия улиц, помещений и т.д.);
- добавление данных;
- удаление данных;
- просмотр данных.

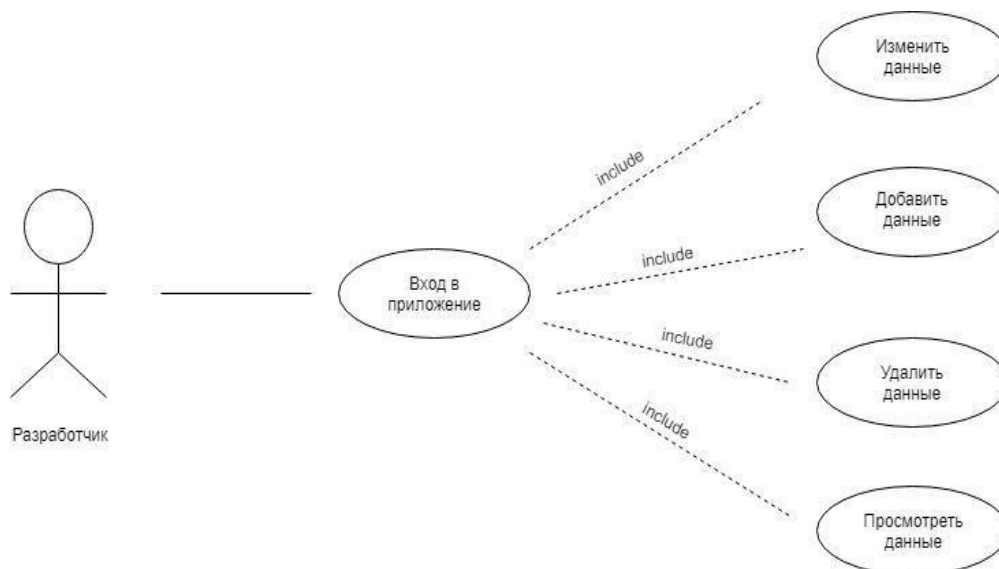


Рисунок 3.1..5 – Диаграмма прецедентов для разработчика

### 3.2 Аспекты реализации

Системные требования. Рекомендуемые системные требования:

- операционная система: Windows 7/8/10-64-bit;
- процессор (CPU): Intel Core i5 (Desktop);
- оперативная память (RAM): 2 ГБ( или больше);
- свободное место на жестком диске: ~35 МВ.

Минимальные системные требования:

- операционная система: Windows 7/8/10-32-bit;
- процессор (CPU): Intel Core i3 (Desktop);
- оперативная память (RAM): 1ГБ( или больше);
- свободное место на жестком диске: ~35 МВ.

Одним из самых важных аспектов для создания данного приложения является подгрузка карты Google в мое мобильно приложение. Для этого нужно получить Google Maps API Key.

Для получения данного ключа я создала учетную запись Google, перешла на платформу Google Maps Platform -> Get Started -> Maps -> Continue. Создала проект, дождалась приглашения, согласилась с условиями, предоставляемыми мне от Google, выбрала страну. После выбора моего аккаунта включила набор сервисов API, далее API был сгенерирован. (рисунок 3.2.1)

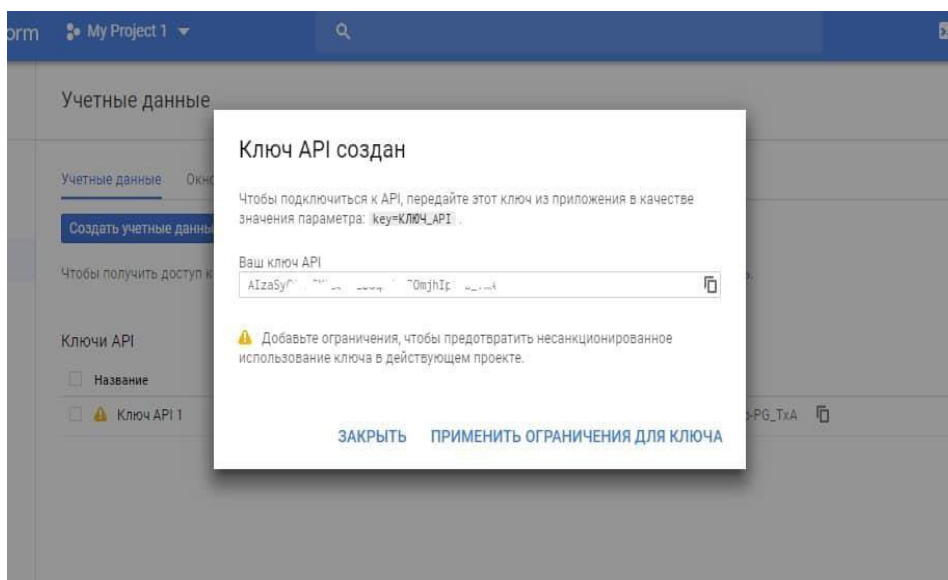


Рисунок 3.2.1 - Присвоение ключа API

Также один из важных пунктов при создании приложения это прорисовка маршрута между двумя маркерами на карте.

Есть две позиции, определенные объектами LatLng, с помощью службы маршрутов добавляю маркеры в начале и в конце маршрута. Так как мне необходимо получить направление между двумя маркерами, обязательно должна быть подгружена исходная карта.

Для отображения маршрута необходимо установить свойство map, после этого при изменении точек на месте, приложение будет маршрутизировано, маршрут будет отображаться (рисунок 3.2.2).

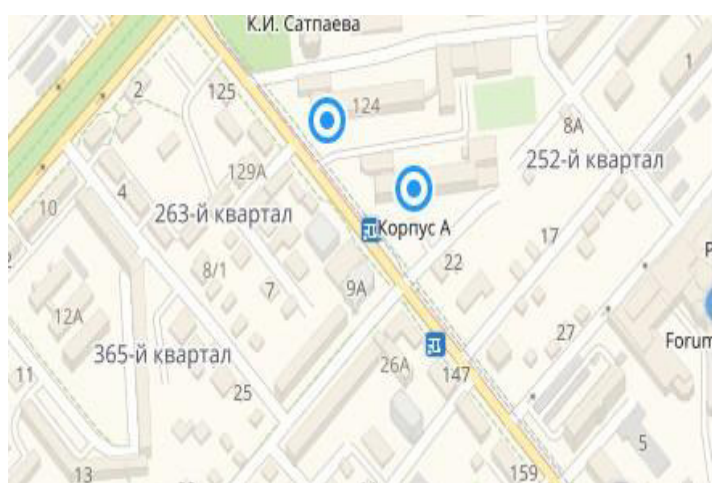


Рисунок 3.2.2- Пример отображения маршрутов.

GPS – основа работы мобильного приложения «Умная остановка». Система Глобального Позиционирования (GPS или Global Positioning System) является спутниковой и работает под управлением Министерства Обороны

США. Система является глобальной, всепогодной и обеспечивает возможность получения точных координат и времени 24 часа в сутки. Основы системы GPS можно разбить на пять основных подпунктов:

- спутниковая трелатирация – базовая часть системы;
- спутниковая дальнометрия –показатель дистанции от устройства до спутника;
- точная временная привязка – функция для обоснования надобности синхронизации часов в устройстве и на спутнике.
- расположение спутников – точное расположения спутника в космосе;
- коррекция ошибок – анализ неудачных попыток в связи с неправильной информации в соях космоса.



Рисунок 3.2.3 - Спутниковая трilaterация

Местоположения предоставленные до сотых по данным широты и долготы, а именно предоставления точного геолокационного местоположения могут быть рассчитаны с помощью спутниковой трilaterацией. Спутники предоставляют четкие координаты. Например, если нам известно диапазон размещения одного спутниками после можно описать сферу вокруг данного спутника, зная диапазон размещения второго спутника пересечением двух сфер будет являться третий спутник.



Рисунок 3.2.4 - Третий спутник определяет две точки на окружности.

При пересечении двух сфер необходимо выбрать точку. Но при наличии быстроты изменения местоположения точки, ее можно не регистрировать. Благодаря наличию координат трех точек можно вычислить локацию третьей точки.

Спутниковая дальнометрия. Расстояние до спутников определяется по измерениям времени прохождения радиосигнала от космического аппарата до приёмника умноженным на скорость света. В тот момент, когда сигнал будет потерян и локация не будет отображена можно вычислить время сигнала.

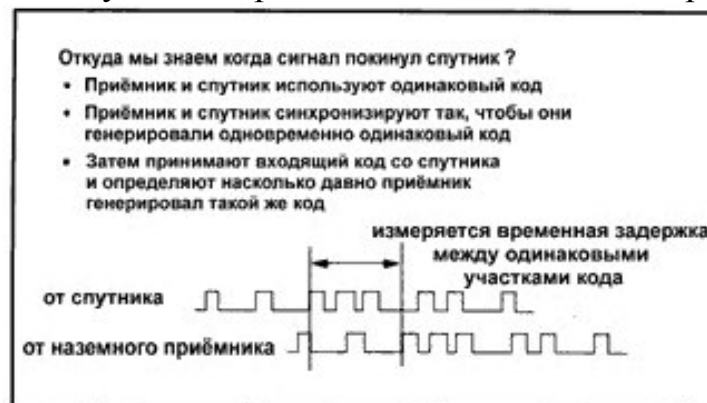


Рисунок 3.2.5 – Сигнал спутника

Для этого на спутнике и в приёмнике одновременно генерируется одинаковый Псевдослучайный

Любой спутник GPS транслирует 2 радиосигнала: в частоте  $L1=1575.42$  МГц и  $L2=1227.6$  МГц. Знак  $L1$  обладает 2 дальномерными программными кодами с псевдослучайным гулом (PRN), P-шифр и C/A шифр. -Точный! либо P-шифр способен являться зашифрованным с целью боевых полетов. -Грубый! либо C/A шифр никак не зашифрован. Знак  $L2$  преобразуется только лишь с P-кодом. Большая часть гражданских юзеров применяют C/A шифр в отсутствие возможности



GPS концепциями. Определенные приёмники Trimble тригонометрического класса функционируют с P-кодом.

Приёмник проводит проверку вступающий знак с попутчика и устанавливает если некто производил такого рода ведь шифр. Приобретенная отличия, поднятая в темп освещение (~300000 км/с) предоставляет желанное дистанция. Использование программный код дает возможность приёмнику установить скоротечную приостановку в каждом периоде. Помимо этого, попутчики имеют все шансы испускать знак в одной и этой ведь частоте, таким образом равно как любой попутчик определяется согласно собственному Псевдослучайному коду (PRN либо PseudoRandom Number code). Конкретная кратковременная связь (рисунок 2.11)

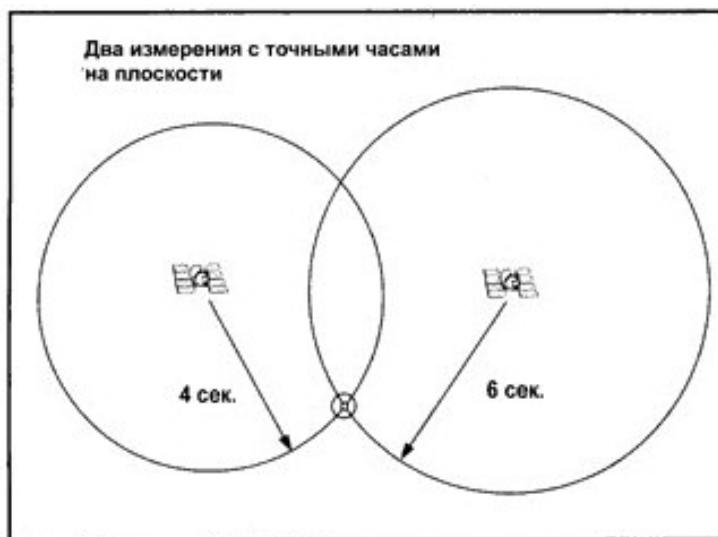


Рисунок 3.2.6 - Точная временная привязка

Результаты полностью зависят от достоверности того, как идут часы. Синхронизация устройства и спутника должна осуществляться в одно время. Спутники владеют часами у которых достоверность соответствует до наносекунд. Но атомные часы очень дорогие в обслуживании и установке, поэтому за точность времени используется четвертый спутник.

Чтобы не допускать ошибок в решении отображения геолокации, используя измерения, показанные на рисунке 3.2.7, так решается проблема если время на устройстве и спутнике различаются. 3 спутника вычисляют местоположения устройства (рисунок 3.2.7)

Эти измерения можно использовать для устранения ошибок, которые возникают если часы на спутнике и в приёмнике не синхронизированы (рисунок 3.2.7). Для наглядности, иллюстрации приведённые ниже рассматривают ситуацию на плоскости, так как только три спутника

необходимо для вычисления местоположения объекта.

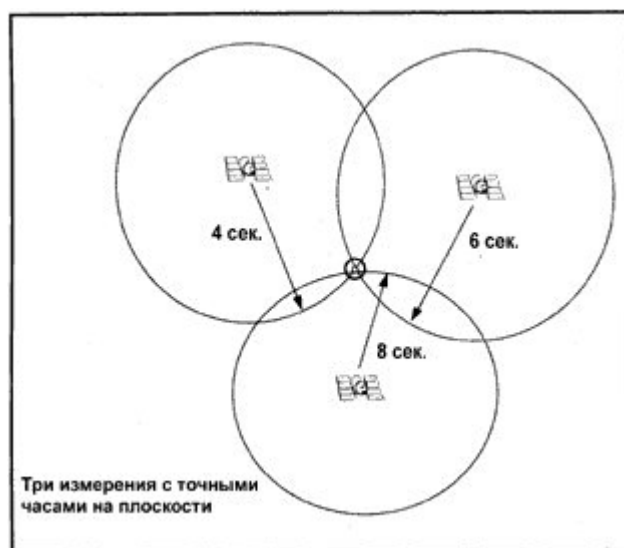


Рисунок 3.2.7 – Измерения часами на плоскости

В случае когда время устройства и спутника синхронизированы точные геолационные данные могут быть получены используя диапазоны 2-ух спутников.

Рисунок 3.2.7 описывает ситуацию когда спутник и устройство синхронизировано, время отображается одинаковое, также известны диапазоны всех 3-ех спутников. Если это так, то пересечение будет выглядеть подобно рисунку 3.2.7.

Но если ошибка в синхронизации будет составлять хотя бы одну секунду, то ситуация будет происходить как на рисунке 3.2.8.

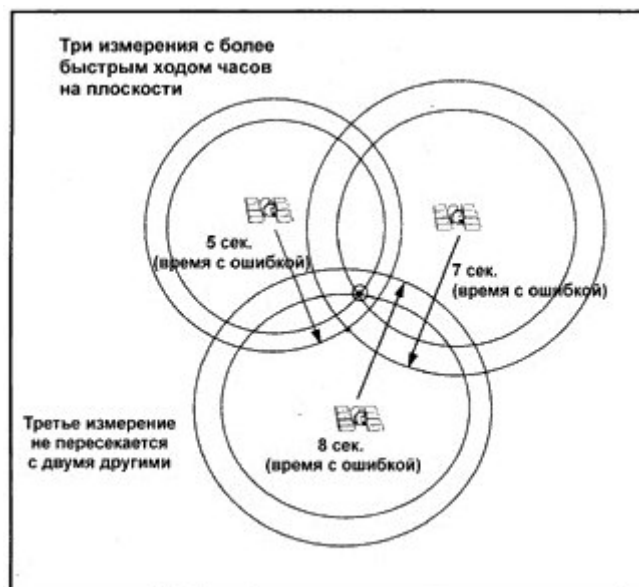


Рисунок 3.2.8 – часы на приемнике, время с ошибкой

Когда устройству с GPS предоставлены несколько диапазонов, где пересечение происходит в 1 точке, то устройство использует метод последовательных итераций до того момента, пока три диапазона не окажутся в одной точке.

Когда GPS приёмник получает серию измерений которые не пересекаются в одной точке, то компьютер в приёмнике начинает вычитать (или добавлять) время методом последовательных итераций до тех пор, пока не сведёт все измерения к одной точке. Уже после данного рассчитывается исправление и производится надлежащее уравнивание.

При надобности 3-ей сферы необходимо использовать 4-ый спутник, чтобы не допустить несоответствия в предоставлении часового пояса в устройстве. Если пользователь находится в поле, для отображения его геолокации будет задействовано 4 спутника. На рисунке 3.2.9. отображены спутники и показаны пересечения в трех измерениях.

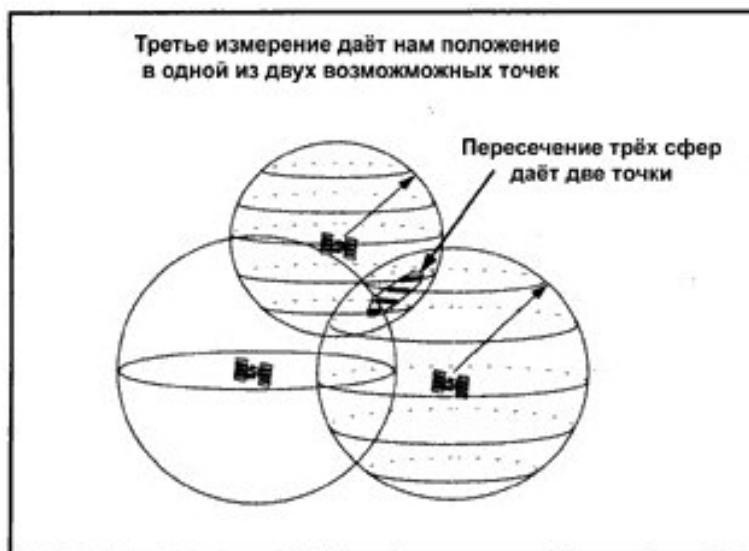


Рисунок 3.2.9 - Расположение спутников

Система NAVSTAR имеет 24 рабочих спутника с орбитальным периодом в 12 часов на высоте примерно 20200 км от поверхности Земли. В шести различных плоскостях имеющих наклон к экватору в  $55^\circ$ , расположено по 4 спутника. Указанная высота необходима для обеспечения стабильности орбитального движения спутников и уменьшения фактора влияния сопротивления атмосферы.

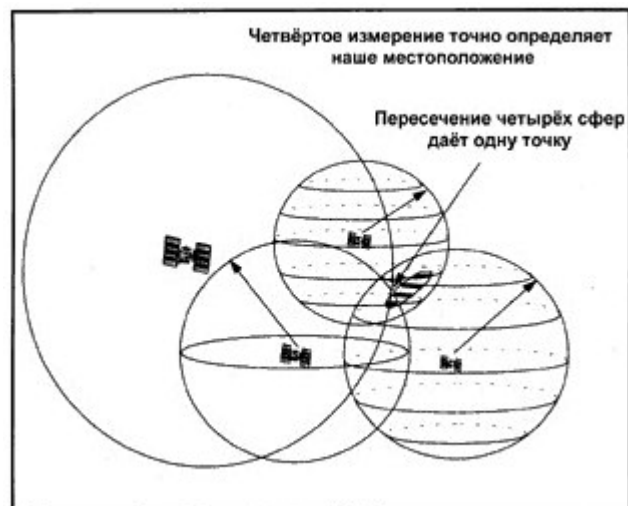


Рисунок 3.2.10 - Уточнённые элементы.

### 3.3 Разработка и описание интерфейса

Главное меню мобильного приложения (рисунок 3.3.1) демонстрирует выбор одного из функций, т.е. локацию, автобусы( маршруты от точки А до точки Б), остановки, места поблизости.

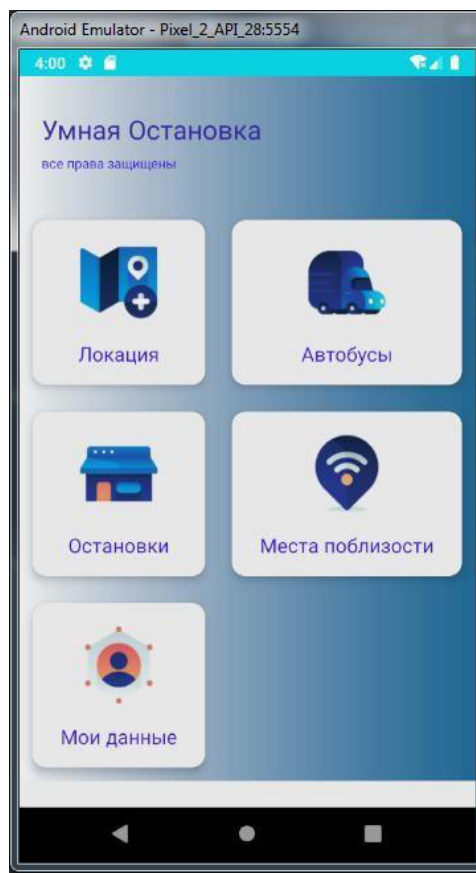


Рисунок 3.3.1 – Главное меню

После входа в приложение и попадания на главную страницу имеются вкладки, после которых можно перейти в один из пунктов, разработанных в данном мобильном приложении. Рисунок 3.3.2 демонстрирует обозначения основных функций. После выбора одной из предоставляющих кнопок, выходит запрос и отображения геолокации либо маршрута, в зависимости от выбора кнопки.

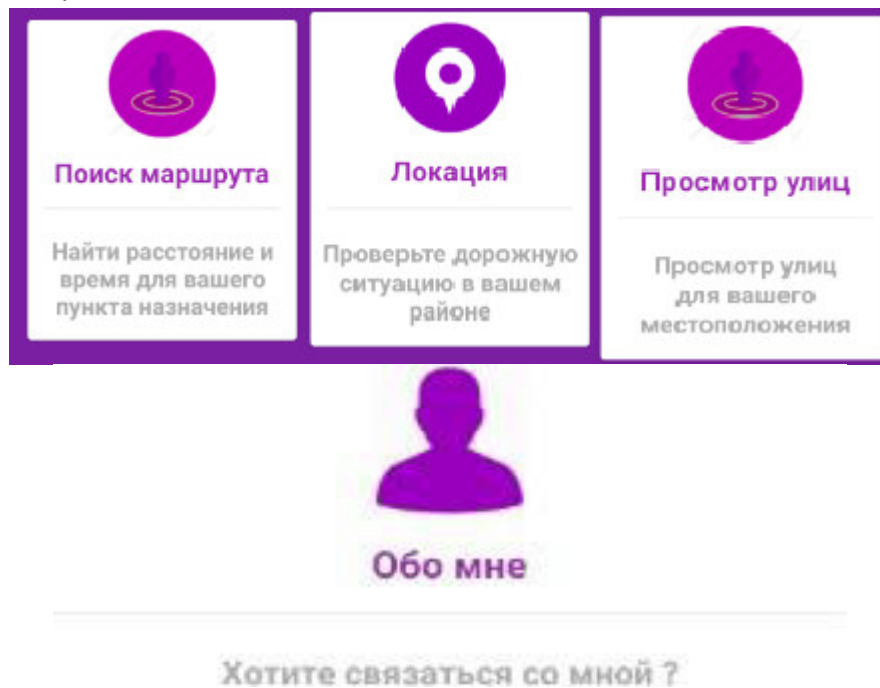


Рисунок 3.3.2 – Основные функции приложения

При нажатии на кнопку места по близости отображается окно с местами (рисунок 3.3.3), которые также можно выбрать, после чего в зависимости от выбора помещения все отображается на карте.

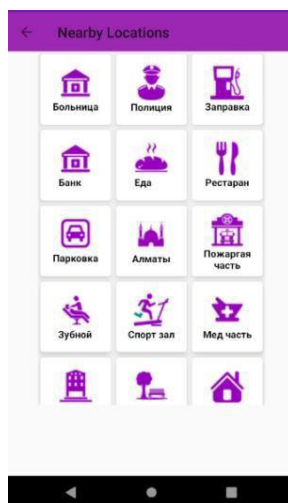


Рисунок 3.3.3 – Окно ближайших геолокационных мест

При выборе информации о разработчике данного приложения выходит окно (рисунок 3.3.4) с данными: фотография, имя, сотовый телефон, почта, социальные сети, место проживания.

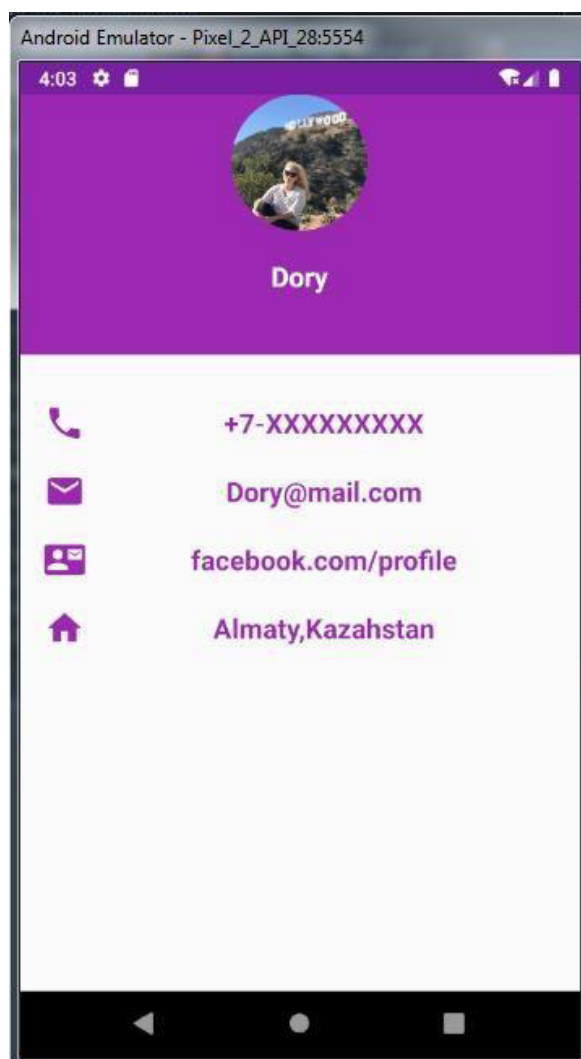


Рисунок 3.3.4 – Окно о разработчике

#### 4 Экономическая часть. Техничко-экономическое обоснование

Цель данного дипломного проекта заключается в разработке программного обеспечения, позволяющего максимально быстро проложить маршрут от точки А до точки Б с обратной связью с пользователем. На данный момент существует множество различных программ позволяющих отображать геолокацию человека и отображение маршрута человека.

Программа будет сочетать в себе оптимальное решение, которое позволит сделать программу менее дорогой и легальной. Данное программное обеспечение может использоваться компаниями, которые регулярно взаимодействуют с автобусными парками и соответствующими государственными органами. В разработке программного обеспечения будет участвовать 1 специалист - программист-разработчик. В обязанности программиста-разработчика входит разработка технического обоснования, разработка программного обеспечения, его тестирование и сопровождение.

Техничко-экономическое обоснование содержит следующие пункты:

- определение трудоемкости разработки программного обеспечения;
- расчет затрат на разработку ПО;
- определение ценности готового продукта;
- оценка результатов работы программного обеспечения.

##### 4.1 Определение сложности разработки ПО

Для того, чтобы точно определить сложность разработки программного обеспечения, необходимо произвести деление всей задачи на более простые подзадачи. Это позволит эффективно следить за прогрессом разработки программного обеспечения. Такой подход, с моей точки зрения, считается более эффективным и позволяет результативно и быстро обрабатывать подзадачи. Модель распределения трудоемкости разработки ПО и стадии разработки представлены в таблице 4.1.

Таблица 4.1 – Этапы разработки ПО

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 1	Постановка задач	10
Этап 2	Разработка и утверждение ТЗ на разработку ПО	20
Этап 3	Поиск и изучение подобных программ	15
Этап 4	Поиск и изучение соответствующей литературы	15
Этап 5	Оформление теоретической части дипломной работы	15

*Продолжение таблицы 4.1*

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 6	Разработка практической части дипломного проекта	25
Этап 7	Реализация проекта	160
Этап 8	Внедрение	15
Итого: трудоемкость выполнения дипломного проекта		275

Продолжительность рабочего дня равна 8 часам. В результате для реализации программного обеспечения необходимо 34 рабочих дней, т.е. 275 часов мы делим на 8-ми часовой рабочий день ( $275/8=34$  дня)

#### **4.2 Расчет затрат на разработку ПО**

Определение затрат необходимых для разработки программного обеспечения производится на основе имеющейся сметы, которая включает следующие элементы:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

Материальные затраты делятся на основные и вспомогательные затраты на материалы, энергию и другие затраты необходимые для разработки ПО. Расчет материальных затрат происходит по форме, предоставленной в таблице 4.2.

Таблица 4.2 – Затраты на материальные ресурсы

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Бумага для офиса	International Paper	Упаковка	2	1 000	2 000
Тетрадь (96 листов)	Маяк Канц	Штук	2	190	380
Блокнот	КТС-ПРО	Штук	2	400	800
Ручки	Parker Jotter	Штук	2	90	180
Компьютерная мышь	TECH	Штук	1	3 000	3 000
Итого:					7 360



Для разработки программного обеспечения будет использоваться ноутбук Lenovo, характеристики данного ноутбука полностью соответствуют выполнению данной дипломной работы. Так как ноутбук содержит установленную операционную систему Windows x64 и программное обеспечение необходимое для разработки ПО, нет нужды производить дополнительные затраты на новую ОС и ПО.

Общую сумму, необходимую на материальные средства ( ) можно рассчитать по следующей формуле:

$$Z_m = \sum P_i * C_i, \quad (4.1)$$

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

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

i - вид материального ресурса;

n - количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 4.3.

Таблица 4.3 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Lenovo Ideapad 330 81FK00CXRK	Штук	1	220 000	220 000
Принтер	HP LASERJET pro M102A	Штук	1	46 990	46 990
Модем	TP-Link Acher c50	Штук	1	13 200	13 200
ОС	Windows 10	Штук	1	-	-
Итого:					280 190

$$Z_m = 7\,360 + 280\,190 = 287\,550 \text{ (тг)}$$

Для реализации программного обеспечения необходимы материалы на сумму 287 550 тенге.

Ноутбук, принтер

Модем

ОС

### 4.3 Расчет затрат на электроэнергию

Так как при разработке программного обеспечения не обойтись без потребления электроэнергии, имеет смысл произвести расчет затрат на электроэнергию.

Согласно таблице 4.1 для разработке программного обеспечения необходимо порядка 225 часов, теперь необходимо рассчитать стоимость электроэнергии, которая будет потрачена в течении 225 часов. Для принтера расчет будет проводиться для периода в 24 часа, так как нет необходимости постоянно использовать принтер.

$$\mathcal{E} = \mathcal{E}_{\text{эл. эн. обор.}} + \mathcal{E}_{\text{доп. нужды.}} \quad (4.2)$$

где – затраты на электроэнергию оборудования;

– затраты электроэнергии на дополнительные нужды.

Расчет электроэнергии, которая необходима для оборудования определяется по следующей формуле:

$$\mathcal{E}_{\text{эл. эн. обор.}} = \sum W * K_{\text{исц}} * S * T \quad (4.3)$$

где W – потребляемая мощность, Вт;

– коэффициент использования ( = 0,7..0,9);

T – время работы;

S – тариф (1кВт/ч = 18,32 тг).

Итоги по расчетам стоимости затрачиваемой электроэнергии представлены в таблице 4.4.

Таблица 4.4 – Затраты на электроэнергию

Наименование прибора	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена на ЭЭ тг/кВтч	Сумма, тг.
Ноутбук	0,6	0,7	275	18,32	2115,96
Модем	0,08	0,9	275	18,32	362,74
Принтер	0,5	0,9	24	18,32	197,86
Кондиционер	0,8	0,9	180	18,32	2374,27
Освещение	0,3	0,7	275	18,32	1057,98
Итого:					6108,8

$$\mathcal{E}_{\text{эл. эн. обор.}} = 6108 \text{ (тенге)}$$

На дополнительные потребности расходы подсчитываются на основе повышенного показателя в объеме 5% от расходов на электроэнергию:

$$З_{\text{доп. нужды}} = 5\% * З_{\text{эл. эн. обор.}} \quad (4.4)$$

Определим затраты на дополнительные потребности согласно формуле (4.4):

$$З_{\text{доп. нужды}} = 0.05 * 6108,8 = 305,44 \text{ (тенге)}$$

Исходя из всех расчетов, полные расходы на электроэнергию составляют:

$$\mathcal{E} = 305,44 + 6108,8 = 6414,24 \text{ (тенге)}$$

#### 4.4 Расчет затрат на оплату труда

Для разработки программного обеспечения, как указывалось ранее, необходимо два работника:

– руководитель проекта – управление рабочим временем, корректировка рабочих процессов, координация, изучение предметной области;

– разработчик – разработка ПО, тестирование и сопровождение.

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$З_{\text{тр}} = \sum ЧС_i * T_i \quad (4.5)$$

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

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

$n$  - количество работников, занятых разработкой ПП.

Во время реализации проекта рабочее время участников не равномерно, поэтому имеет смысл установить часовую ставку каждого работника и общий объем заработной платы.

Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (4.6)$$

где - месячная заработная плата  $i$ -го работника, тг;

- месячный фонд рабочего времени  $i$ -го работника, час.

Месячная заработная плата разработчика равняется 150 000 тенге. Рассчитаем часовую ставку разработчика согласно формуле (4.6):

$$ЧС_{\text{разработчик}} = \frac{150\,000}{22 \cdot 8} = 852,3 \text{ тг/ч}$$

Часовая ставка разработчика составляет 852,3 (тг/ч), трудоемкость разработки равняется 275 часам. Согласно формуле (4.5) можно рассчитать сумму расходов на заработную плату работников:

$$З_{\text{тр}} = 852,3 \cdot 275 = 191\,767,5$$

Расчеты затрат по оплате труда показаны в таблице (4.5).

Таблица 4.5. – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Разработчик	Программист	225	852,3	191 767,5
Итого:				191 767,5

#### 4.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонда оплаты труда. Социальный налог можно рассчитать по следующей формуле:

$$С_{\text{н}} = (\text{ФОТ} - \text{ПО}) \cdot 0,095 \quad (4.7)$$

где ПО - отчисления в пенсионный фонд, они составляют 10% от ФОТ.

$$\begin{aligned} \text{ПО} &= 191\,767,5 \cdot 0,1 = 19\,176,75 \text{ тенге} \\ С_{\text{н}} &= (191\,767,5 - 19\,176,75) \cdot 0,095 = 16\,396,1212 \text{ тенге} \end{aligned}$$

Результаты расчетов представлены в таблице (4.6):

Таблица 4.6 – Начисление социального налога

Категория работника	Количество человек	Заработная плата, тг	Пенсионные отчисления, тг	Социальный налог, тг

Продолжение таблицы 4.6

Категория работника	Количество человек	Заработная плата, тг	Пенсионные отчисления, тг	Социальный налог, тг
Разработчик	1	191 767,5	19 177	16 396,1212
Итого:				16396,1212

#### 4.6 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$A_T = \frac{C_{об} * H_a}{100} \quad ( )$$

где, – стоимость оборудования;

– норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$A_T = \frac{220\,000 * 25}{100} = 55\,000 \text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$A_T = \frac{55\,000 * 34}{365} = 5\,123,29 \text{ тенге}$$

Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (4.7).

Таблица 4.7 – Амортизация ОФ

Наименование оборудования, н·ПО	Стоимость оборудования, н·ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	220 000	25	55 000	5 123,29

Продолжение таблицы 4.7

Наименование оборудования и ПО	Стоимость оборудования и ПО, ₸	Годовая норма амортизации, %	Сумма амортизации за год, ₸	Сумма амортизации за время разработки, ₸
Принтер	46 990	25	11 747,5	4 377,15
Модем	13 200	20	2 640	164,5
Итого:			69 387,5	9 664,94

Смета расходов на разработку ПО.

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице (4.8). Также предусматриваем прочие расходы (расходы на интернет) в сумме 10,000 тенге.

Таблица 4.8 – Смета затрат на разработку ПО

Статьи затрат	Сумма, ₸	Сумма в %
Затраты на оборудование	280 190	55%
Затраты на оплату труда	191 767,5	38%
Социальные налоги	16 396,1212	3%
Затраты на электроэнергию	5 512	1%
Амортизация основных фондов	5 123,29	1%
Прочие расходы	10 000	2%
Итого по смете:	508 988,911	100%

### Диаграмма затрат

- затраты на оборудование(55%)
- Затраты на оплату труда(38%)
- Социальные налоги (3%)
- Затраты на электроэнергию(1%)
- Амортизация основных фондов(1%)
- Прочие расходы(2%)

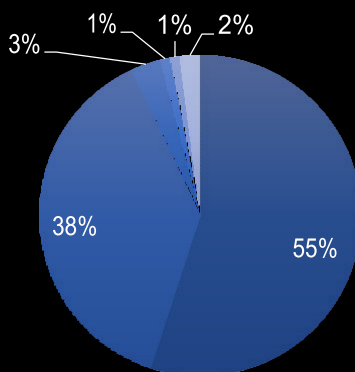


Рисунок 4.1 - диаграмма затрат

#### 4.7 Определение возможной (договорной) цены ПО

Стоимость программного обеспечения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость для программного обеспечения можно рассчитать по следующей формуле:

$$Ц_{д} = Z_{\text{нир}} \left( 1 + \frac{P}{100} \right), \quad ( )$$

где – затраты на разработку программного обеспечения, тг;

P – средний уровень рентабельности ПО, (%). Данный параметр принят равным 25%.

$$\begin{aligned} Ц_{д} &= 508\,988,911 \left( 1 + \frac{25}{100} \right) = 508\,988,911 + (508\,988,911 * 0,25) = \\ &= 508\,988,911 + 127\,247,228 = 636\,236,139 \text{ тенге} \end{aligned}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле:

$$Ц_{р} = Ц_{д} + Ц_{д} * \text{НДС}, \quad ( )$$

$$Ц_{р} = 636\,236,139 + 636\,236,139 * 0,12 = 712\,584,476 \text{ тенге}$$

Данную цену можно округлить до 713 000 тенге.

Стоимость реализации с учетом НДС равна 712 584,476 тенге

Прибыль П равна  $508\,988,911 * 0,25 = 127\,247,228$  тенге

Себестоимость мобильного приложения С равна 508 988,911 тенге

## 5 Производственная безопасность

В данной дипломной работе «Мобильное приложение на платформе Android «Умная остановка»» будут рассмотрены условия труда для одного разработчика в комнате размерами 4\*3\*2.5, с одним окном 1\*2.25, одна лампа. В данном помещении условия освещения и шума соответствуют нормам и требованиям согласно ГОСТ 12.1.003, потому что ноутбук Lenovo G50-70, лампы накаливания в количестве 1 штуки не превышают нормативного давления и нормам освещенности. Поэтому рассмотрим условия вентиляции.

Ниже предоставлены таблица с исходными данными и рисунок (5.1) описание помещения.

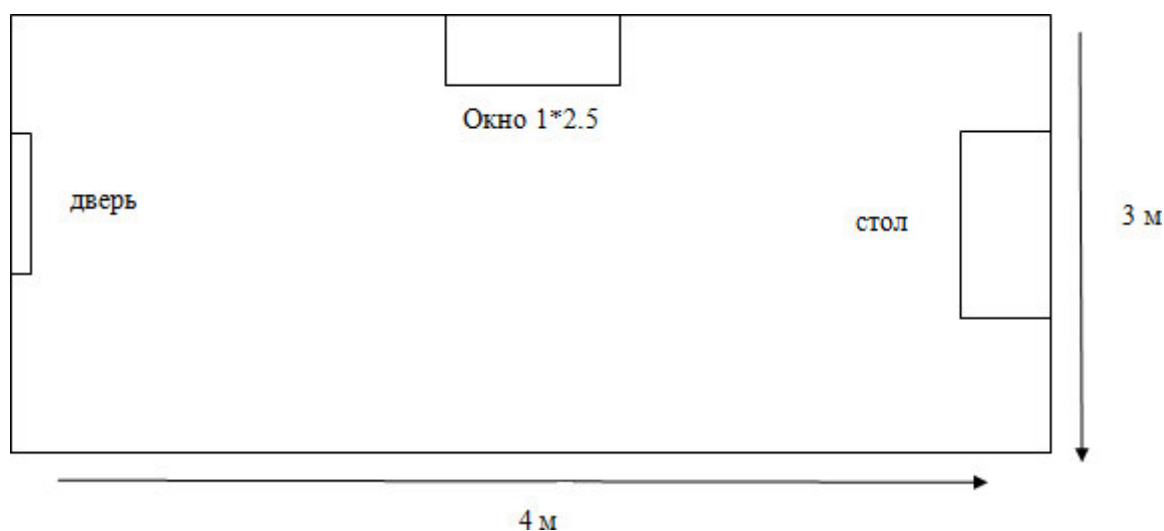


Рисунок 5.1 - Помещение для разработки

Ниже предоставлена таблица с исходными данными. Все что имеется в помещении, количество человек. Условия труда данного помещения.

Таблица 5.1 - Исходные данные

Город	Алматы	
Параметры помещения (Д x Ш x В), м	4 x 3 x 2,5	
Данные по оборудованию	кол-во, шт	1
	мощ. $P_{об}$ , кВт/ч	0,65
	КПД, $\eta$	0,75
Данные по источникам света	мощ. $N_{ос.уст.}$ , Вт/м <sup>2</sup>	60
	вид ист. св.	лампы накаливания
Число сотрудников, из них	Мужчины	0
	Женщины	1
Окна	кол-во	1



Продолжение таблицы 5.1

Окна	кол-во	1
	площадь 1 окна, м <sup>2</sup>	2,25
	Расположение	ЮВ/ЮЗ
	Вид	Шторы, пластиковый подоконник, загрязнение незначительное
Расчетное время суток, ч.	14-15	
Температура помещения, °С	летом	24
	Зимой	-5
Вид положения работы	Сидя	

Расчет тепловых нагрузок в помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние).¶

### 5.1 Наружные тепловые нагрузки

Данные нагрузки представлены следующими составляющими:

- теплопоступления или теплопотери в результате разности температур снаружи и внутри здания через стены, потолки, полы, окна и двери.
- разность температур снаружи здания и внутри него летом является положительной, в результате чего имеет место приток тепла снаружи во внутрь помещения; и наоборот – зимой эта разность отрицательна и направление потока тепла меняется;
- теплопоступления от солнечного излучения через застекленные площади; данная нагрузка проявляется в форме ощущаемого тепла;
- теплопоступления от инфильтрации.

В зависимости от времени года и времени суток наружные тепловые нагрузки могут быть положительными.

Теплопоступления и теплопотери в результате разности температур определяются по формуле:

$$Q_{огр}^n = 0, \text{ кВт.}$$

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

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

$$V_{пом} = 4 \cdot 3 \cdot 2,5 = 30 \text{ м}^3;$$

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

$$t_{\text{Нрасч}} = -12 \text{ }^{\circ}\text{C}$$

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

$$t_{\text{Врасч}} = 18 \text{ }^{\circ}\text{C}$$

$$Q_{\text{озр}}^3 = 30 * 0,42 * (-12 - 18) = 12,6 * (-30) = -378 \text{ Вт}$$

Теплопоступление от солнечного излучения через остекление определяется по формуле:

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

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

$F_0^I, F_0^{II}$  – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м<sup>2</sup>;

$\beta_{\text{с.з.}}$  – коэффициент теплопропускания. По таблице :

$$\beta_{\text{с.з.}} = 0,15$$

Для периода тени, когда лучи солнца не проникают через окна (рассеянная радиация)  $F_0^I=0$ ;  $F_0^{II}=0$ :

$$Q_p = q^{II} F_0 \cdot \beta_{\text{с.з.}} = q_{\text{вр}} \cdot K_1^T \cdot K_2 \cdot \beta_{\text{с.з.}} \cdot n \cdot S_0, \text{ Вт}$$

$q_{\text{вп}}; q_{\text{вр}}$  – тепловые потоки от рассеянной радиации, Вт/м<sup>2</sup>. По таблице для широты в 44° СШ после полудня в 14-15 ч. при расположении ЮВ:

$$q_{\text{вр}} = 63 \text{ Вт/м}^2;$$

$F_0 = nS_0 = 1 * 2,25 = 2,25 \text{ м}^2$  – площадь светового проема ( $n$  – число окон;  $S_0$  – площадь 1 окна);

$K_1$  – коэффициент затемнения остекления переплетами ( $K_1^T$  – для проемов в тени). По таблице 6 [1]:

$$K_1^T = 1,28;$$

$K_2$  – коэффициент загрязнения остекления. По таблице :

$$K_2 = 0,95.$$

Тогда:

$$Q_p = 63 * 1,28 * 0,95 * 0,15 * 1 * 2,25 = 25,86 \text{ Вт}$$

По таблице для широты в 44°СШ после полудня в 14-15 ч. при расположении ЮЗ:

$$q_{\text{вп}} = 101 \text{ Вт/м}^2;$$

$$F_o = nS_o = 1 \cdot 2,25 = 2,25 \text{ м}^2$$

$$Q_p = 101 \cdot 1,28 \cdot 0,95 \cdot 0,15 \cdot 9 = 165,8 \text{ Вт}$$

Тогда:

Тогда общее теплоступление солнечного излучения с обеих сторон равно:

$$Q_p = 25,86 + 165,8 = 191,66 \text{ Вт}$$

### 5.3 Расчет теплового баланса в помещении

На основании выполненных расчетов составим баланс теплоступлений в помещении:

$$Q_{\text{изб}} = Q_p + Q^a + Q_{\text{осв}} + Q_{\text{об}} + Q_{\text{орг}} + Q_{\text{опр}}$$

$$\text{Лето: } Q_{\text{изб}}^{\text{л}} = 191,66 + 56,95 + 662,4 + 490 + 195 - 0 = 1596,01 \text{ Дж}$$

$$\text{Зима: } Q_{\text{изб}}^{\text{з}} = 191,66 + 97,75 + 662,4 + 490 + 195 - 378 = 1258,81 \text{ Дж}$$

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

$$Q_{\text{н}} = \frac{Q_{\text{изб.лет}} \cdot 860}{V_{\text{пом}}} = \frac{1596,01 \cdot 860}{128} = 10,72 \text{ ккал/м}^3.$$

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

при  $Q_{\text{н}} < 20 \text{ ккал/м}^3$ ,  $\Delta t = 6 \text{ }^\circ\text{C}$ .

Определение количества воздуха, необходимое для поступления в помещение:

$$L = \frac{Q_{\text{изб}} \cdot 860}{C \cdot \Delta t \cdot \gamma} = \frac{1596,01 \cdot 860}{0,24 \cdot 8 \cdot 1,206} = 592,769 \text{ м}^3/\text{час}$$

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

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

Определение кратности воздухообмена:

$$n = \frac{L}{V_{\text{пом}}} = \frac{592,769}{128} = 4,63 \text{ час}^{-1}.$$

### 5.4 Выбор кондиционера и расположение его в помещении

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

Таблица 3.1 – Основные технические характеристики кондиционера фирмы UNIFLAIР модель SUA 0351.

Размеры, мм	Мощность, кВт			масса, кг	расход воздуха, м <sup>3</sup> /ч мин: 1940; макс: 3020
	по холоду	компрессора	электронагревателя		
1740x850x450	13,4	3,7	3,3	185	

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 (см. рисунок 5.4.2), а забирается через лицевую панель.

В данном помещении на основании выполненных расчетов баланс теплоступлений в помещении равен 1596 Дж, этот баланс мы можем заменить кондиционером. Ниже предоставлен рисунок расположения кондиционера в комнате для оптимальной вентиляционной системы (см. рисунок 5.4.1).



Рисунок 5.4.1 - помещение для разработки с кондиционером

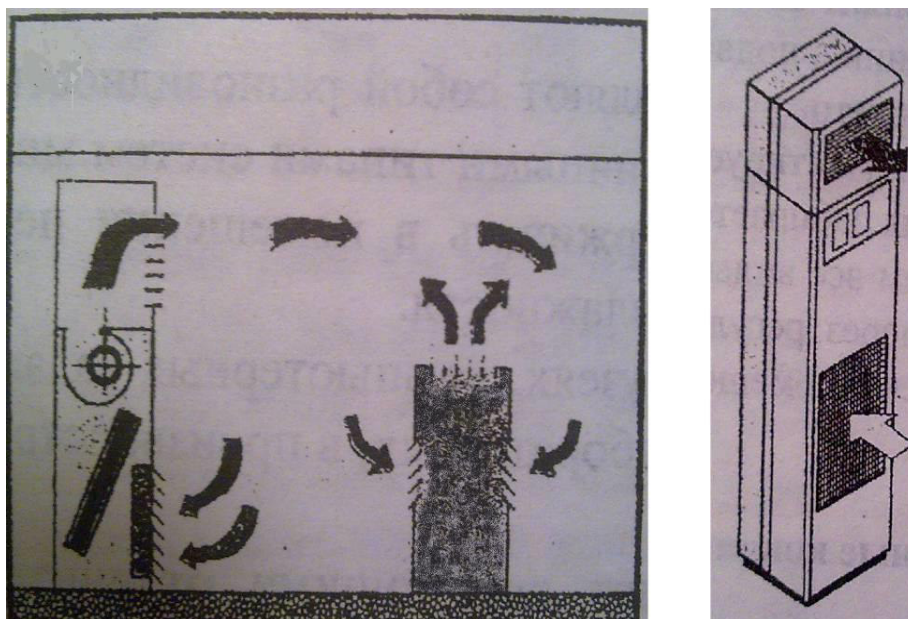


Рисунок 5.4.2 – Схема подачи воздуха (верхняя)

## Заключение

В настоящее время основным условием стабильного функционирования на рынке любого предприятия является совершенствование методов управления. Один из инструментов совершенствования систем управления предприятием – внедрение информационных систем, которое позволяет повысить эффективность хозяйственной и финансовой деятельности компании благодаря снижению рисков за счет оперативного получения необходимой информации, своевременного принятия решений и т.д.

Для этого были реализованы следующие задачи:

- дана характеристика транспортной деятельности в Казахстане и зарубежом;
- проведен анализ сравнения;
- на основе проведенного анализа разработана система;
- дана оценка экономической эффективности внедрения информационной системы;
- описаны условия безопасности жизнедеятельности.

## Список литературы

- 1 Леонтьев Ю. Общественный транспорт в России. Создание приложений. Руководство пользователя. – М.: Компьютера, 2001.
- 2 Медникс З., Дорнин Л. Программирование под Android. Издательство Питер, 2012.
- 3 Амелин К. С., Граничин О. Н., Кияев В. И., Корякко А. В.. Введение в разработку приложений для мобильных платформ. Издательство ВВМ, 2011.
- 4 Мельникова О.М.: Смартфоны на Android. Издательство Эксмо, 2013.
- 5 Каймин В.А. Информатика: Учебное пособие: Изд. 2-е. Издательство РИОР, 2007.
- 6 Варакин М.В. Разработка мобильных приложений под Android. УЦ «Специалист» при МГТУ им. Н. Э. Баумана, 2012.
- 7 Пайлон Д. UML 2 для программистов. Издательство Питер, 2012.
- 8 Котляров В.П. Основы тестирования программного обеспечения.
- 9 Издательство Бином, 2009. Druker P. Innovation and Entrepreneurship. Harper Collins, 1985.
- 10 Лаптев Г.Д., Ладоненко М.А. и др. Компетентностный подход и роль дизайн-мышления в обучении инновационному предпринимательству.— М.: ТЕИС, 2010.
- 11 <http://appdeveloper.intel.com/en-us/article/program-overview>
- 12 Эрве Лебре. Стартапы. Чему мы ещё можем научиться у Кремниевой долины.— М.: ООО «Корпоративные издания», 2010, 216 с.
- 13 Основы компьютерного моделирования// Программа компьютерного моделирования BPwin (AllFusion Process Modeler). URL: <http://bourabai.kz/cm/bpwin.htm> (дата обращения: 22.02.2019)
- 14 Студенческая библиотека онлайн// Обеспечение работы веб-сайта. URL: [https://studbooks.net/2215545/informatika/obespechenie\\_raboty\\_sayta](https://studbooks.net/2215545/informatika/obespechenie_raboty_sayta) (дата обращения 25.02.2019)
- 15 Ресурс для IT-специалистов// Visual Studio Code – редактор кода для Linux, OSX и Windows. URL: <https://habr.com/ru/company/microsoft/blog/262523/> (дата обращения: 01.03.2019)
- 16 Ткаченко В. Обучение в Интернет// Структура программного обеспечения ПК. URL: <https://www.lessons-tva.info/edu/e-inf1/e-inf1-3-1.html> (дата обращения: 03.03.2019)
- 17 Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат – Алматы: АИЭС, 2009. – 20 с.
- 18 Естественное и искусственное освещение. Государственные нормативы в области архитектуры, градостроительства и строительства. СНиП РК 2.04.-05.2002.

- 19 Косьяненко М.А. Способы разработки мобильных приложений одновременно для os android и ios / Инноватика-2018: сб. материалов XIV 18 Международной школы-конференции студентов, аспирантов и молодых ученых (26-27 апреля 2018 года) / под ред. А.Н. Солдатова, С.Л. Минькова. - Томск, СТТ, 2018 (в печ.)
- 20 Гриффитс Д., Гриффитс Д. Head First. Программирование для Android [Текст]. – СПб.: Питер, 2016. – 704 с.: ил. – (Серия «Head First O'Reilly»).
- 21 SQLite vs MySQL vs PostgreSQL: сравнение систем управления базами данных [Электронный ресурс]. – URL: <http://devacademy.ru/posts/sqlite-vs-mysql-vs-postgresql/> (дата обращения: 21.02.2018).
- 22 Android Developers. CameraDevice [Электронный ресурс]. – <https://developer.android.com/reference/android/hardware/camera2/CameraDevice.html>. (дата обращения: 27.04.2018)
- 23 Android Developers. Connecting to the Network [Электронный ресурс]. – <https://developer.android.com/training/basics/network-ops/connecting.html> (дата обращения: 07.04.2018).
- 24 Android Developers. LocationListener [Электронный ресурс]. – <https://developer.android.com/reference/android/location/LocationListener.html>. (дата обращения: 07.03.2018).
- 25 Android Developers. Motion Sensors [Электронный ресурс]. – ([https://developer.android.com/guide/topics/sensors/sensors\\_motion.html#sensors-motion-linear](https://developer.android.com/guide/topics/sensors/sensors_motion.html#sensors-motion-linear)). (дата обращения: 15.05.2018).
- 26 Обзор среды программирования Android Studio [Электронный ресурс]. – URL: <https://www.intuit.ru/studies/courses/12643/1191/lecture/21980?page=3> (дата обращения: 9.02.2018).
- 27 Медникс З., Дорнин Л. Программирование под Android. Издательство Питер, 2012.
- 28 <https://informburo.kz/tags/obshchestvenniy-transport.html> - Общественный транспорт в Казахстане
- 29 Варакин М.В. Разработка мобильных приложений под Android.
- 30 Ресурс для IT-специалистов// Visual Studio Code – редактор кода для Linux, OSX и Windows. URL: <https://habr.com/ru/company/microsoft/blog/262523/> (дата обращения: 01.03.2019)



## Приложение А (обязательное)

### Техническое задание

Цель работы: Мобильное геолокационное приложение на платформе  
Андроид

Общие требования:

Наименование программного продукта (ПП): Мобильное приложение

Цель разработки ПО:

Целью разработки ПО является создание мобильного приложения, которое будет отображать геолокацию пользователя, показывать маршруты от точки А до точки Б, показывать нужные гео-позиции по близости от места нахождения.

Идеология программного обеспечения:

Идеологией программного обеспечения является объективное отображение геолокационных данных на карте, отображении маршрутов и автобусов города Алматы.

Используемая технология создания ПО:

Программирование в Android studio на языке программирования java .

Выбор модели ПО:

Структурная модель. Позволяет оформлять отдельные блоки программы (повторяющиеся и неповторяющиеся) в процедуры и функции, которые затем могут использоваться в других частях программы. Изменения в коде функций и процедур не влекут за собой изменение других частей кода программы.

Общая блок-схема программного продукта (рисунок А)

Блок-схемы, которая включает в себя графическое представление определения, анализа и метода решения задачи, в котором используются символы для отображения данных, потока, оборудования. В блоках дается описание соответствующего действия описании и изображении алгоритмов данного программного продукта.

На рисунке А представлена блок-схема, отображающая полностью работу данного программного продукта, представляющая циклы, выборы данных, работу приложения от начала до конца.

Общая блок-схема программного продукта (рисунок А)

Блок-схемы, которая включает в себя графическое представление определения, анализа и метода решения задачи, в котором используются символы для отображения данных, потока, оборудования. В блоках дается описание соответствующего действия описании и изображении алгоритмов данного программного продукта

Продолжение приложения А

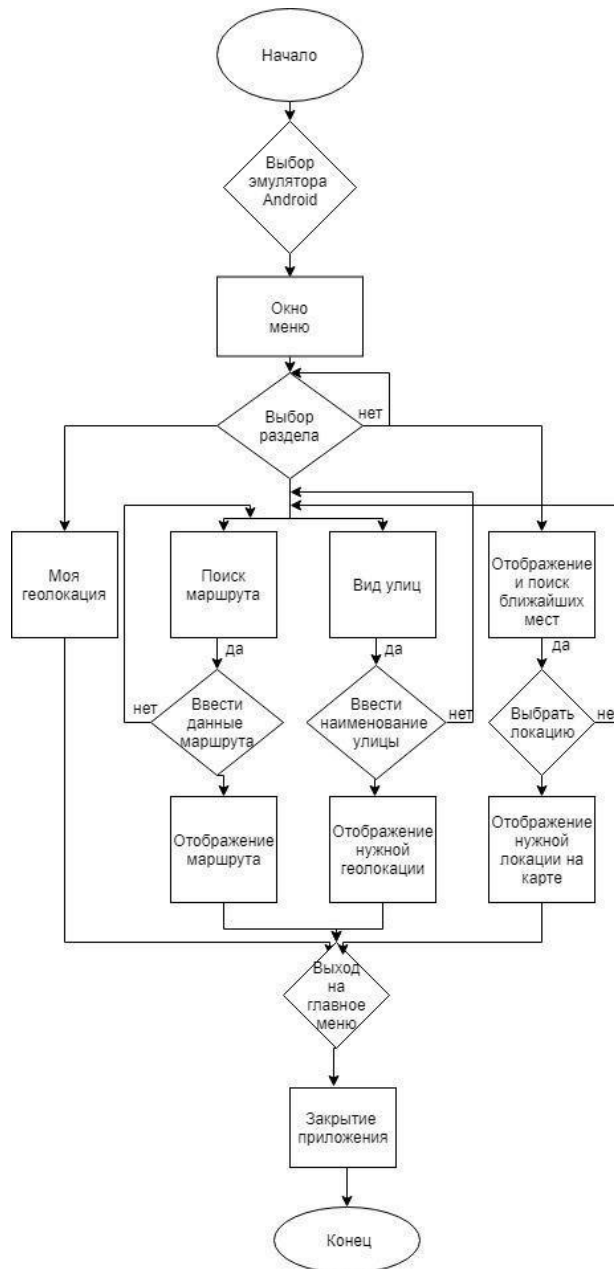


Рисунок А – Блок схема программного продукта

Технические требования:

- основной диапазон разрешения мониторов, на которых будет просматриваться ПП: 1280x960 пикселей;
- минимальное разрешение монитора, в котором будет просматриваться ПП: 800x600 пикселей;
- операционные системы, на которых работает программное обеспечение: семейство поколений Windows, начиная с Windows 7;
- минимальные требования ПК: частота процессора 1.0 ГГц, 128 Мб ОЗУ, 32 Мб видео памяти, CD-ROM;

## *Продолжение приложения А*

- рекомендуемая конфигурация компьютера: частота процессора 2.0 ГГц, 256 Мб ОЗУ, 64 Мб видео памяти, CD-ROM;

- обеспечение защищенности программного продукта: в основе защиты ПП лежат методы шифрования данных в файлах путем создания двоичных файлов вопросов (в формате bin).

- надежность программного обеспечения: надежность программы должны находиться на высоком уровне. Ни при каких обстоятельствах ПП не должен вызывать сбои. Работа программы не вызовет конфликт с другими программными продуктами, системами и платформами Windows.

- тестирование и отладка программного продукта: тестирование и отладка программы будут проводиться в параллельной разработке пробной версии. Изначально данный ПП будет работать в виде пробной версии (в течении 7 дней).

- тестирование на вредоносный код: после завершения создания ПП необходимо будет протестировать его на наличие вредоносного кода с помощью известных программ Kaspersky, NOD32 и т.д. При наличии вредоносного кода в программе необходимо будет обезвредить его. Наличие вредоносного кода не будет умышлено, так как создание данной программы не подразумевает создание вредоносных функций (уничтожение данных, уничтожение реестра и т.д.).

Специфические требования:

- адаптивность ПП: программа будет отвечать данному требованию. Адаптироваться она будет по результатам ответов на вопросы, выдавая непосредственно следующие вопросы;

- интеллектуальное развитие ПП: в данном ПП не будет необходимости интеллектуального развития программы;

- системность ПП: ПП не будет системным;

- полнофункциональность ПП: ПП будет выполнять все возложенные на него функции: надёжность, легкость применения, мобильность, завершённость, независимость от устройств и т.д.;

- целостность ПП: данный ПП должен быть целостным и не требовать у пользователей установки дополнительных модулей;

- функциональность: данная программа будет выполнять все функции внедренные в неё;

- легкость применения: данное ПО должно быть легким для применения любому пользователю. Содержать понятливые документации по применению и работы с данной программой. ПП не должна вызывать сбоев при работе;

- сопровождаемость: реализация данного пункта предусматривает два критерия качества – изучаемость и модифицируемость. Изучаемость должна быть обеспечена грамотным интерфейсом и ясной справочной системы.

*Продолжение приложения А*

Модифицируемость не предусматривается в рамках разработки данной программы;

– завершенность: разрабатываемый ПП должен быть отвечать всем возложенным на него требованиям;

– точность: в данной программе все вычисления должны быть выполнены с минимальными погрешностями;

– эффективность по памяти: разрабатываемый ПП должен будет справляться со всеми возложенными на него задачами при определенных ограничениях на используемую память;

– эффективность по устройствам: работа ПП не должна перегружать устройства вычислительной машин;

– структурированность: в данном ПП должна быть без ошибок налажена связь между всеми компонентами программы в единое целое;

– модульность: изменение одного из дискретных компонент программы вызовет минимальное воздействие на другие компоненты, так как в разрабатываемом ПП будет четко спроектирована связь и взаимодействие между всеми компонентами.

## Приложение Б (обязательное)

### Листинг программы

Activity location

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:ads="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimaryDark">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:gravity="center"
        android:textColor="#ffffff"
        android:textStyle="bold"
        android:textSize="20sp"
        android:fontFamily="monospace"
        android:text="Please enable GPS."
        android:layout_alignParentEnd="true"
    />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="monospace"
        android:layout_below="@+id/textView"
        android:layout_marginTop="62dp"
        android:textColor="#FFFFFF"
        android:gravity="center_horizontal"
        android:text="To enjoy all that App has to offer, turn on your GPS and give access to your
location"
        android:layout_alignParentStart="true"
    />
    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#ffffff"
        android:text="Turn on GPS"
        android:onClick="turn_on_gps"
        android:textSize="18sp"
        android:textAllCaps="false"
        android:layout_alignParentBottom="true"
```

## Продолжение приложения Б

```
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"/>
<com.google.android.gms.ads.AdView
    xmlns:ads="http://schemas.android.com/apk/res-auto"
    android:id="@+id/adView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    ads:adSize="SMART_BANNER"
    ads:adUnitId="@string/banner_home_footer"
    android:layout_above="@+id/button3"
    android:layout_alignParentStart="true">
</com.google.android.gms.ads.AdView>
<ImageView
    android:id="@+id/imageView6"
    android:layout_width="175dp"
    android:layout_height="175dp"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="73dp"
    ads:srcCompat="@drawable/icon" />
</RelativeLayout>Activity about <?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.haroonfazal.myapps.gpsnavigation.AboutActivity">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="5"
    android:orientation="vertical"
    android:layout_below="@+id/myToolbar">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1.5"
    android:background="@color/colorPrimary"
    android:orientation="vertical">
<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/imageView5"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_gravity="center"
    android:src="@drawable/dory" />
<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

## Продолжение приложения Б

```
        android:layout_marginTop="20dp"
    android:gravity="center"
        android:text="Dory"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:textStyle="bold"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="vertical"
    android:padding="15dp"
    android:layout_marginTop="20dp"
    android:layout_weight="3.5">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="36dp"
            android:layout_height="30dp"
            android:src="@drawable/ic_phone_black_24dp" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:textStyle="bold"
            android:textSize="20sp"
            android:textColor="@color/colorPrimary"
            android:text="+7-XXXXXXXXXX"
            android:gravity="center" />
    </LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal">
    <ImageView
        android:layout_width="36dp"
        android:layout_height="30dp"
        android:src="@drawable/ic_email_black_24dp" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:textStyle="bold"
        android:textSize="20sp"
        android:textColor="@color/colorPrimary"
        android:text="Dory@mail.com"
        android:gravity="center" />
</LinearLayout>
```

## Продолжение приложения Б

```
</LinearLayout>
<LinearLayout
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal" android:layout_width="36dp"
    android:layout_height="30dp"
    android:src="@drawable/ic_contact_mail_black_24dp" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:textStyle="bold"
    android:textSize="20sp"
    android:textColor="@color/colorPrimary"
    android:text="facebook.com/profile"
    android:gravity="center" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:orientation="horizontal">
    <ImageView
        android:layout_width="36dp"
        android:layout_height="30dp"
        android:src="@drawable/ic_home_black_24dp" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:textStyle="bold"
        android:textSize="20sp"
        android:text="Almaty,Kazahstan"
        android:textColor="@color/colorPrimary"
        android:gravity="center" />
</LinearLayout>
</LinearLayout>
</LinearLayout>
</LinearLayout>
<com.google.android.gms.ads.AdView
    android:id="@+id/adView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:adSize="SMART_BANNER"
    android:layout_alignParentBottom="true"
    app:adUnitId="@string/banner_home_footer">

</com.google.android.gms.ads.AdView>
</RelativeLayout>
    Google maps IP<resources>
<!--
```



## Продолжение приложения Б

*TODO: Before you run your application, you need a Google Maps API key.*

To get one, follow this link, follow the directions and press "Create" at the end:

[https://console.developers.google.com/flows/enableapi?apiid=maps\\_android\\_backend&keyType=CLIENT\\_SIDE\\_ANDROID&r=CE:7B:7E:DA:12:83:95:1B:8F:4B:85:12:03:62:31:DB:C7:F2:28:1B%3Bcom.haroonfazal.myapps.gpsnavigation](https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=CE:7B:7E:DA:12:83:95:1B:8F:4B:85:12:03:62:31:DB:C7:F2:28:1B%3Bcom.haroonfazal.myapps.gpsnavigation)

You can also add your credentials to an existing key, using these values:

Package name:

CE:7B:7E:DA:12:83:95:1B:8F:4B:85:12:03:62:31:DB:C7:F2:28:1B

SHA-1 certificate fingerprint:

CE:7B:7E:DA:12:83:95:1B:8F:4B:85:12:03:62:31:DB:C7:F2:28:1B

Alternatively, follow the directions here:

<https://developers.google.com/maps/documentation/android/start#get-key>

Once you have your key (it starts with "AIza"), replace the "google\_maps\_key" string in this file.

-->

```
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">
</string>
```

```
</resources>
```

My GPS// Top-level build file where you can add configuration options common to all sub-projects/modules.

```
buildscript {
    repositories {
        jcenter()
        google()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.4.1'
```

```
        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
```

```
    }
}
allprojects {
    repositories {

        jcenter()
        maven {
            url "https://maven.google.com"
        }
        maven
        {
            url "https://jitpack.io"
        }
        google()
    }
}
```

## Продолжение приложения Б

```
task clean(type: Delete) {
delete rootProject.buildDir}
nav_header_my_navigation
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="@dimen/nav_header_height"
android:background="@drawable/side_nav_bar"
android:gravity="bottom"
android:orientation="vertical"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:theme="@style/ThemeOverlay.AppCompat.Dark">
<ImageView
android:id="@+id/imageView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:paddingTop="@dimen/nav_header_vertical_spacing"
android:src="@drawable/circleloc"/>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:paddingTop="@dimen/nav_header_vertical_spacing"
android:textStyle="bold"
android:textSize="22sp"
android:text="Dory разработчик"
android:textAppearance="@style/TextAppearance.AppCompat.Body1" />
<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Dory@gmail.com" />
</LinearLayout>
```