

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

« ____ » _____ 2018 г.

ДИПЛОМНАЯ РАБОТА

На тему: Разработка мобильного приложения для ремонтного сервиса

Специальность: 5В060200 – «Информатика»

Выполнил: Махмутов К. И. Группа ИНФ-14-2

Научный руководитель: к. ф-м. н., доцент Калижанова А. У.

Консультанты:

по экономической части: к.э.н., доцент _____ А.И.Бекишева
« 05 » 06 _____ 2018 г.

по безопасности жизнедеятельности: к.т.н., доцент _____ А.А.Абикенова
« 07 » 06 _____ 2018 г.

по применению
вычислительной техники: ст. преп. _____ А.М. Рамазанова
« 06 » 06 _____ 2018 г.

Нормоконтролер: ст. преп. _____ Ш.Д.Толыбаев
« 08 » 06 _____ 2018 г.

Рецензент: PhD _____ О.Ж. Мамырбаев
« ____ » _____ 2018 г.

Алматы 2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В060200 – «Информатика»

ЗАДАНИЕ

на выполнение дипломной работы

Студенту Махмутову Камилжану Имаржановичу

Тема работы: Разработка мобильного приложения для ремонтного сервиса

Утверждена приказом по университету № 155 от «23» октября 2017 г.

Срок сдачи законченной работы «01» июня 2018 г.

Исходные данные к работе (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): среда разработки Delphi 10.1 Berlin, базы данных SQLite.

Перечень вопросов, подлежащих разработке в дипломной работе, или краткое содержание дипломной работы:

- а) Провести анализ рынка мобильных приложений;
- б) Изучить платформу Android;
- в) Использовать БД SQLite при разработке;
- г) Анализ основных аспектов мобильной разработки;

Перечень графического материала (с точным указанием обязательных чертежей): представлены 13 таблиц, 47 иллюстрации и 1 диаграмма.

Основная рекомендуемая литература:

- 1 Фаронов В.В. Delphi 6: Учебный курс. – СПб.: Питер, 2002.
- 2 Климова, Л. М. Delphi 7. Основы программирования. Решение типовых задач. Самоучитель / Л.М. Климова. - М.: КУДИЦ-Образ, 2017. - 480 с.
- 3 Б. Харди, Б. Филлипс, К. Стюарт, К. Марсикано, «Android. Программирование для профессионалов», 2016

4 Культин, Никита Основы программирования в Delphi XE / Никита Культин. - М.: "БХВ-Петербург", 2011. - 416 с.

5 Санников, Е. В. Курс практического программирования в Delphi. Объектно-ориентированное программирование / Е.В. Санников. - М.: Солон-Пресс, 2013. - 188 с.

Консультации по работе с указанием относящихся к ним разделов работы

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Бекишева А.И.	17.05-05.06.18	
Безопасности жизнедеятельности	Абикенова А.А.	14.03-04.06.18	
Программная часть	Рамазанова А.М.	05.05-1.06.18	
Нормоконтролер	Толыбаев Ш.Д.	05.05-08.06.18	

График
подготовки дипломной работы

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Анализ и исследование предметной области	01.11.2017-20.12.2017	Выполнено
Проектирование мобильного приложения	21.12.2017-20.02.2018	Выполнено
Программная реализация приложения	21.02.2018-30.04.2018	Выполнено

Дата выдачи задания «25» октября 2017 г.

Заведующий кафедрой  Т.С. Картбаев

Научный руководитель работы  А.У. Калижанова

Задание принял к исполнению студент  К. И. Махмутов

Аннотация

Цель дипломной работы разработка и создание приложения для платформы Android. Предназначением данного приложения является ведение базы данных для сервисного центра.

Целевой аудиторией программного продукта являются представители малого бизнеса, занимающиеся ремонтом мобильных телефонов. В приложении можно вести базу данных и добавлять запчасти и описывать их характеристики, такие как тип, цвет и цена. Также есть функция поиска по этой базе.

Программа была написана на языке программирования Delphi в среде разработки Delphi 10.1 с интеграцией фреймворка FireMonkey. В приложении использовались базы данных SQLite.

Annotation

The purpose of thesis work is development and creation of application for Android platform. The purpose of this application is accounting a database for the service-center.

The target audience of software product is small businesses that repair mobile phones. In the application you can maintain a database, add parts and describe their characteristics, such as type, color and price. There is also a search function for this database.

The program was written on the Delphi programming language in the Delphi 10.1 development environment with the integration of the FireMonkey framework. The application used SQLite databases.

Аңдатпа

Дипломдық жұмыстың мақсаты – Android платформасына арналған қосымшаны жасау. Осы қосымшаның мақсаты сервис орталығына арналған деректер базасын сақтау болып табылады.

Бағдарламалық жасақтама өнімінің мақсатты аудиториясы ұялы телефондарды жөндеу бойынша шағын кәсіпорындар болып табылады. Өтініште дерекқорды сақтауға және бөлшектерді қосуға және сипаттамаларын, мысалы, түрі, түсі мен бағасын сипаттай аласыз. Бұл дерекқор үшін іздеу функциясы бар.

Бағдарлама Delphi 10.1 әзірлеу ортасында Delphi бағдарламалау тілінде жазылған, ол FireMonkey құрылымын біріктірді. Қолданба SQLite дерекқорларын қолданды.

Содержание

	Введение	8
1	Аналитическая часть	9
1.1	Тренды рынка мобильных приложений	9
1.2	Анализ ОС Android	15
1.3	Основные аспекты разработки приложения	33
2	Проектная часть	50
2.1	Подготовка устройства для отладки	50
2.2	Установка Delphi	51
2.3	База данных для Android, SQLite	53
2.3	Разработка приложения	56
3	Экспериментальная часть	61
4	Технико-экономическое обоснование проекта	64
4.1	Трудоемкость разработки проекта	64
4.2	Расчет затрат на разработку	65
4.3	Определение возможной (договорной) цены проекта	73
4.4	Оценка социально-экономических результатов проектирования	74
5	Безопасность жизнедеятельности	76
5.1	Анализ условий труда	76
	Заключение	85
	Список литературы	86
	Приложение А	Ошибка! Закладка не определена.

Введение

Индустрия мобильных приложений обширна и с каждым днем она становится еще больше. Семь лет назад всем известная компания Apple, запустила площадку на которой можно было скачивать приложения и устанавливать их на свой смартфон. Называется она App Store. Чуть позже был представлен Android Market. Сейчас он известен под названием Google Play Market. В данный момент это площадка на которой каждый может разместить свое приложение. Для кого-то это отличный способ заработка, за счет этого компании Google и Apple дали огромный скачок для развития индустрии мобильных приложений. Они были пионерами и тогда и стали лидерами сейчас.

На сегодняшний день почти у каждого третьего человека есть мобильный телефон. Это неудивительно ведь учитывая темпы движения технологий очень важно быть в курсе событий и всегда оставаться на связи. За последние несколько лет, индустрия мобильных телефонов разрослась с невероятной скоростью. Старые устройства с огромным количеством ограничений заменяются на новые смартфоны с огромной поддержкой различных сервисов, начиная от Google maps, iCloud и т. д., заканчивая различными сторонними сервисами предлагаемые различными корпорациями, такими как Алиса от Яндекс, Vixby от Samsung и тому подобные.

Причин роста множество. Все они довольно просты и достаточно очевидны для понятия среднестатистического человека:

- абсолютно все смартфоны помещаются в карман, за счет этого мы все имеем прямой доступ к любому сервису, в необходимое нам время;

- рынок смартфонов рассчитан на потребителя любого класса. Ассортимент настолько колоссален, что позволить себе работоспособный телефон может каждый.

Целью моей дипломной работы является разработка приложения для сервис центра. Основная функция мобильного приложения, это ведение базы данных. Учет ведется по запчастям, которые есть в наличии на складе, а также по пришедшим клиентам, для составления клиентской базы, с целью дальнейшей работы с ними.

В данной работе я провел анализ рынка мобильных приложений и ОС Android в целом. Целью анализа было выявление трендов в разработке мобильных приложений для более гибкой разработки отвечающей запросам современного пользователя. Также было проведено исследование наиболее актуальных версий платформы Android, для того чтобы минимизировать возможные конфликты программного продукта с устройством. Системы учета актуальны всегда, особенно если они подтверждают ожидания клиента. В данный момент широкое применение мобильных приложений находится на пике, ведь их популярность растет с увеличением зон применения смартфонов.

1 Аналитическая часть

1.1 Тренды рынка мобильных приложений

На мировом рынке наблюдается огромный рост числа пользователей смартфонов. Вы не можете понять тенденции, которые следует учитывать при разработке мобильных приложений в 2018 году.

Все большее число пользователей смартфонов на мировом рынке привело к значительному увеличению числа приложений, которые потребители используют на своих телефонах. В магазине App Store есть около 2 миллионов таких приложений, в то время как в Google Play Market более 2,2 миллиона приложений. Ожидается, что это число увеличится в будущем.

Инфографика, выпущенная GO-Globe, также показывает, что общее использование мобильных приложений растет. Пользователи смартфонов понимают важную роль, которую играют приложения в оказании помощи клиентам и получении доступа к важной информации. Согласно этим статистическим данным, 52% времени, которое люди тратят на цифровой медиаконтент, находятся в мобильных приложениях. Этот аспект ясно указывает на то, что предприятия получают большой доход от использования мобильных приложений.

Согласно отчету comScore, число мобильных пользователей превысило пользователей настольных компьютеров в 2014 году. В прогнозе Gartner говорится, что более 268 миллиардов загрузок для мобильных устройств будут получать доход в размере 77 миллиардов долларов в 2017 году. По оценкам, глобальный доход от мобильных приложений составил 35 миллиардов долларов в 2014 году, 45 миллиардов долларов в 2015 году – 58 млрд. долл. США в 2016 году и предсказанные 77 млн. долл. США в 2017 году, как указано выше. Статистика показывает, что люди, которые попадают в возрастную группу от 18 до 24 лет, используют больше мобильных приложений, чем любая другая группа. За ними следует пристальное внимание от 25 до 34 лет, от 35 до 44 лет, от 45 до 54 лет, а затем более 55 лет, соответственно. Подразумевается, что компании, использующие мобильные приложения, будут нацелены на среднее поколение больше, чем на младшие и старшие поколения.

Согласно этим статистическим данным, женщины больше времени проводят в мобильных приложениях, чем мужчины. Кроме того, среднее время, которое люди тратят на приложения, увеличилось на 20% с 2014 по 2015 год. Статистика также показывает, что люди тратят 43% времени на мобильные игры, 26% на социальные сети, 10% на развлечения, 10% на коммунальные услуги, 2% на новости и производительность, 1% на здоровье и образ жизни и 5% на других. Неудивительно, что количество доступных игр для мобильных приложений находится на подъеме – 85% людей предпочитают собственные приложения для мобильных сайтов.

После анализа траты времени на приложения стало ясно, музыка составляет 79%, здоровье и фитнес – 51%, социальные сети – 49%, 28% – 22%, спорт – 16%, 15% – игры и новости на 14%. Процент мобильных приложений, которые были использованы только один раз, сократился до 20% в 2015 году после улучшения до 22% в 2014 году. Процент приложений, используемых более чем 11 раз, увеличился на 39% в 2014 году. Эти статистические данные теперь могут помочь нам определить тенденции развития мобильных приложений в 2018 году.

Приложения IoT (рус. Интернет вещей) играют важную роль в нашей повседневной жизни. Этот аспект будет по-прежнему играть важную роль в разработке мобильных приложений в 2018 году. Прибыль и доходы от IoT находятся на подъеме, и ожидается, что тенденция продолжится в 2018 году. Большинство предприятий экспериментируют с IoT в автомобилях, умных домах, образовании, безопасности и индустрии здравоохранения и имеют очень позитивные результаты. Следствием этой тенденции является то, что IoT станет отличным шагом в 2018 году.

В двух словах, Internet of Things – это концепция подключения любого устройства в Интернет и на другие подключенные устройства. IoT – это гигантская сеть связанных вещей и людей – все из которых собирают и обмениваются данными о том, как они используются, и об окружающей среде вокруг них.

Это включает в себя необычайное количество объектов всех форм и размеров от умных микроволновок, которые автоматически готовят вашу пищу в течение необходимого времени, к самоуправляемым автомобилям, чьи комплексные датчики обнаруживают объекты на своем пути, к пригодным для ношения фитнес-устройствам, которые измеряют ваш сердечный ритм и количество шагов, которые вы сделали в этот день, затем используют эту информацию, чтобы предложить планы упражнений, адаптированные к вам. Есть даже связанные футбольные мячи, которые могут отслеживать, как быстро их бросают, и записывать эту статистику через приложение в целях обучения в будущем.

Устройства и объекты со встроенными датчиками подключены к платформе Internet of Things, которая объединяет данные с разных устройств и применяет аналитику для обмена наиболее ценной информацией с приложениями, созданными для удовлетворения конкретных потребностей.

Эти мощные платформы IoT могут точно определить, какая информация полезна и что можно смело игнорировать. Эта информация может использоваться для обнаружения шаблонов, предоставления рекомендаций и выявления возможных проблем до их возникновения.

Например, если у меня есть бизнес по производству автомобилей, я, возможно, захочу узнать, какие дополнительные компоненты (например, кожаные сиденья или диски из сплава) являются самыми популярными.

Используя технологию Internet of Things, я могу:

- использовать датчики, чтобы определить, какие области в салоне являются самыми популярными, и где клиенты задерживаются дольше всего;
- перейти к имеющимся данным по продажам, чтобы определить, какие компоненты продаются быстрее всего;
- автоматически выравнивать данные о продажах с поставкой, чтобы популярные товары не заканчивались.

Информация, собранная подключенными устройствами, позволяет принимать разумные решения о том, что мне следует купить, основываясь на информации в реальном времени, что помогает экономить время и деньги.

Благодаря проницательности, предоставляемой передовой аналитикой, появляется возможность сделать процессы более эффективными. Умные устройства и системы позволяют автоматизировать определенные задачи, особенно когда они повторяются, обыденны, отнимают много времени или даже опасны.

Представьте, что вы просыпаетесь в 7 утра каждый день, чтобы идти на работу. Будильник делает работу, пробуждая вас в нужное время без вашего вмешательства. Ваш поезд метро отменен, и вам нужно ехать на работу за рулем. Единственная проблема заключается в том, что на вождение требуется больше времени, и вам нужно было встать в 6.45 утра, чтобы не опоздать. Вдобавок льет дождь, так что вам нужно ехать медленнее, чем обычно. Связанный с поддержкой IoT, будильник подстраивается на основе всех этих факторов, чтобы обеспечить своевременное пробуждение. Будильник знает, что ваш обычный поезд отменен, рассчитывает расстояние и время поездки для вашего альтернативного маршрута до работы, проверяет погоду и коэффициент медленной скорости передвижения из-за сильного дождя и рассчитывает, когда ему необходимо разбудить вас. Если это супер умный будильник, он способен синхронизировать свою работу с кофеваркой при помощи поддержки IoT, чтобы ваш утренний кофе был готов к пробуждению.

Дело в том, что у экспертов есть разногласия, когда дело доходит до определения точной даты появления мобильных приложений. Технически, именно BlackBerry впервые представила мобильные приложения. Им предлагается клиент электронной почты для пользователей смартфонов BlackBerry. 14 лет назад. Однако большинство экспертов согласны с тем, что история этих приложений началась с разработки iPhone приложений в 2008 году.

Сегодня Apple является одной из самых успешных технологических компаний. Одной из причин этого успеха является тот факт, что они также являются разработчиками мобильных приложений. Конечно, приложения, найденные в популярном магазине приложений, также выпускаются другими компаниями, потому что Apple позволяет сторонним разработчикам делиться своими приложениями.

Итак, после выхода первого iPhone, устройства, которое поставляется с некоторыми предустановленными приложениями, официально iPhone решили представить App Store в 2008 году. Интересно, что история разработки

приложений для Android началась с того же самого года, всего несколько месяцев спустя. Это был момент, когда Google представил Android Market, который теперь известен как Google Play Store. Первым смартфоном на платформе Android стал HTC Dream от HTC.

Интересно, что количество загрузок мобильных приложений в App Store достигло одного миллиарда после одного года. Это было, безусловно, подходящее время для создания компании по разработке мобильных приложений. Через несколько месяцев рынок разработки мобильных приложений Android достиг еще одного рубежа – одного миллиарда загрузок. Это был также период, когда различные производители представили свои первые фаблеты, такие как Apple iPad и Galaxy от Samsung.

Интересно отметить, что есть несколько других известных компаний, которые вложили средства в разработку мобильных приложений, а одна из них – Windows. Их пионерские шаги были сделаны в 1996 году, и даже несмотря на то, что Windows опередила конкуренцию, «классические» мобильные приложения, которые мы знаем, были представлены позже, чем приложения для IOS и Android. Разработка мобильных приложений для Windows продолжает оставаться одним из самых важных видов деятельности в этой отрасли.

Конечно, в этот период сами приложения менялись, улучшались и развивались. Например, в прошлом многие мобильные приложения использовали так называемую систему значков или текстовый дисплей, чтобы представить свои функции. С другой стороны, в современных приложениях есть HD-фотографии, видео и отличная мобильность. Кроме того, перелистывание заменило прокрутку во многих приложениях, делая пользовательский интерфейс более плавным, чем когда-либо. Последние приложения сосредоточены на сенсорных элементах управления, потому что это делает их использование намного проще.

Количество хакерских атак и угроз неизбежно поразило отрасль мобильных приложений, учитывая ее огромный рост. Ожидается более сильный рост встроенных функций безопасности приложений iPhone и Android. Разработчики приложений будут сосредоточены на включении сильного шифрования данных в ближайшие мобильные приложения. Безопасность мобильных приложений находится в центре притяжения для разработчиков мобильных приложений 2018 года.

Большинство мобильных устройств в настоящее время имеют возможности GPS. Будет продолжена эволюция LBS (Location Based Services) для предоставления пользователям информации в режиме реального времени в 2018 году. Ожидается, что интерес к приложениям, основанным на местоположении, будет процветать в 2018 году больше, чем когда-либо прежде. Некоторые из ключевых областей, в которых мы будем максимально использовать LBS, включают туризм и путешествия, навигацию, функции безопасности, розничные предложения, порталы для определения местоположения и картографирование внутри помещений. Ведущие

технологические гиганты совершили большие прыжки в области LBS. В этой области также могут процветать производители смартфонов, учитывая, что эта технология истощает срок службы батареи.

Недавно Google объявила о характеристиках продукта на базе ИИ, ребрендинге своей исследовательской группы, чтобы подчеркнуть ИИ, и демонстрации его потрясающе человеческого дуплексного агента искусственного интеллекта.

ИИ также играет определенную роль в последней версии Android OS (Android P, теперь в бета-версии) в виде новой функции «адаптивной батареи», которая учится на шаблонах использования, как увеличить время автономной работы конкретного устройства. Другая функция, основанная на ИИ, «адаптивная яркость», узнает, какая степень яркости экрана более комфортна глазу пользователя

«Темпы инноваций в области искусственного интеллекта захватывают дух», – сказал Джейсон Титус, возглавляющий группу разработчиков продуктов компании во время работы над проектом. «Весь новый набор возможностей изменит то, как мы делаем вещи. Мы находимся в точке перегиба. ИИ раньше был тем, что могли использовать только глубокие эксперты и докторанты. Теперь он становится доступным для всех».

Google увеличили поддержку языка программирования Kotlin. Команда Android от Google объявила о первоклассной поддержке Kotlin на прошлогоднем I/O и начала отправлять ее вместе с Java и C++ с официальной платформой Android IDE. С тех пор 35% профессиональных разработчиков теперь используют его для разработки на платформе Android.

Большинство людей рассматривают голосовых помощников как забавный и простой способ помочь им с цифровыми задачами. Самые известные помощники на современном рынке это Alexa, Siri, Google Assistant, и конечно же отечественная Алиса от компании Яндекс. Большинство разработчиков мобильных приложений стремятся понять, как работают помощники. В 2018 году мы увидим, что большинство из них работают над тем, чтобы помощники были похожи на людей в структуре распознавания речи и личностей, что приведет к более плавному взаимодействию.

Голосовая технология уже много лет работает с нами – от автоматических систем распознавания голоса, которые не понимают акценты, до простых диктофонов с голосовым текстом, которые производят неточную копию, – но недостатки этих систем не способствовали широкому распространению.

Тем не менее, достижения в области ИИ означают, что возможности голосового управления, наконец, становятся реальностью, согласно Speak Easy, исследование, проведенное Дж. Уолтером Томпсоном, Кантаром и Миндшаре.

В настоящее время наиболее популярное использование голоса среди сингапурцев относительно просто – задавать вопросы веселья (32%), выполнять онлайн-поиск (30%) и играть музыку (27%). Тем не менее,

функциональность, вероятно, будет быстро расти, как только подключенные устройства начнут появляться чаще.

«Естественные пользовательские интерфейсы, такие как голос и жест, представляют собой эволюцию технологий. По словам Марка Вайзера, отца вездесущих вычислений, мы вступаем в эпоху «спокойной технологии», где люди могут легко использовать технологии, не облагаясь налогом, чтобы изучить ее. Голос интуитивно понятен, и с улучшенной точностью и машинным обучением теперь можно обеспечить более плавный, умный опыт», – говорит Ида Сиу, руководитель отдела планирования, Singapore & SEA.

Также всегда были в тренде приложения позволяющие автоматизировать бизнес процессы. С появлением автономных устройств нужда в них только усилилась. Такие программы называются CRM (Customer Relationship Management).

Mobile CRM (Customer Relationship Management) обеспечивает полный опыт работы с CRM на смартфонах, планшетах и других устройствах с поддержкой Интернета. Это позволяет группам продаж, маркетинга и обслуживания клиентов получать доступ к ключевой информации и управлять ею в реальном времени, где бы они ни находились.

Массовая миграция пользователей Интернета с настольных компьютеров на мобильные устройства означает, что люди теперь ожидают, что информация в реальном времени будет доступна из любого места.

Фактически, доля глобального интернет-трафика, поступающего с мобильных устройств, выросла с 0,7% в 2009 году до 52,99% в 2017 году, и число продолжает расти. CRM должен соответствовать этому росту. Мобильный CRM позволяет пользователям получать важные бизнес данные независимо от того, где они находятся. В некотором смысле, они позволяют им вести свой бизнес на ходу.

Сотрудники офиса больше не «привязаны» к своим столам; они работают на ходу. Члены отдела продаж путешествуют между офисами, и их местонахождение меняется постоянно. Полевым агентам необходим доступ к расписанию и рабочим ордерам, даже если они не находятся в офисе.

Даже вакансии, которые всего пять лет назад требовали присутствия сотрудника в офисе, становятся более продуктивными, когда им предоставляется возможность получить доступ к своей работе удаленно. Исследования показали, что компании ежегодно получают 240 часов работы от сотрудников, использующих мобильные устройства для работы.

Это связано с тем, что мобильные устройства позволяют пользователям работать независимо от их местонахождения. Даже если они не находятся в одном здании или даже в той же стране, отделы всей вашей организации получают доступ к тем же данным, независимо от проблем совместимости, связанных с платформами, в создании более совместной рабочей среды.

С помощью мобильного программного обеспечения CRM мобильные устройства, такие как планшеты и смартфоны, могут предоставлять все классические функции системы, размещенной на ноутбуке или настольном

компьютере, но с дополнительными преимуществами сквозного доступа через облачные сервисы. По сути, пользователи должны иметь возможность вести свой бизнес со своего телефона.

1.2 Анализ ОС Android

На базе Android работает около 2 миллиардов устройств. Это целая четверть всего земного шара. Эта платформа разработана компанией Google. Предназначена она для мобильных устройств, но на ней работают также и умные часы и всевозможные аудиоустройства. Создана она на основе Linux, язык создавался на языке C.

Секрет популярности этой системы прост, так как она разработана на основе Linux, она имеет открытый исходный код, это значит что любой желающий может настраивать его под себя для максимального комфорта при использовании. Каждый может изменить интерфейс своего устройства менять виджеты, и добавлять крутые фишки.

С технической стороны телефон Android работает под управлением операционной системы Android, ключевых приложений и промежуточного программного обеспечения. Это Linux на базе ядра и написан на языке C. Android запускает приложения, написанные в основном на языке Java®. Телефоны Android считаются революционными, все дело в том что они имеют открытый исходный код в стоковой версии. Все разработчики, будь то профессионалы или любители, поощряются к написанию программ и приложений для операционной системы и вносят свой вклад в развивающийся проект, отправляя их в Google Play Store , торговую площадку на которой разработчики размещают свои работы. Затем пользователи могут загружать и использовать приложения на своих телефонах, обращаясь к библиотеке приложений через свои телефоны.

Телефон на базе андроид может одновременно запускать несколько приложений в фоновом режиме, что упрощает многозадачность. Это делает функциональность телефона более гибкой, чем некоторые конкурирующие платформы, хотя степень этой возможности зависит от аппаратного обеспечения отдельного телефона. В некоторых телефонах используются многоядерные процессоры, которые более полезны для «опытных пользователей», которым нужны их телефоны для запуска нескольких операций одновременно.

Некоторые сопоставимые устройства, использующие другие операционные системы, регулируются строгими собственными правилами. Разработчикам, заинтересованным в написании приложений для iPhone, например, не разрешается создавать программы, которые очень похожи на приложения, принадлежащие Apple, такие как iTunes. Все приложения должны быть одобрены корпорацией, прежде чем им разрешат появляться на площадке App Store . Это не относится к приложениям для телефонов Android, которые разрабатываются под эгидой Open Handset Alliance, консорциума,

который поощряет использование и продвижение программного обеспечения с открытым исходным кодом для мобильных телефонов.

Еще одним важным отличием платформы является то, что она не привязана к конкретному производителю оборудования. Любой производитель, заинтересованный в создании собственной версии телефона на основе Android, может использовать операционную систему. В результате многие конкурирующие компании сотовой связи теперь используют эту платформу для поддержки своих мобильных телефонов и выхода на рынок своих устройств.

Несмотря на относительно открытую структуру операционной системы, производители многих устройств, которые используют платформу от Google, часто ограничивают доступ пользователей к определенным ресурсам и функциям в структуре программного обеспечения. Как правило, производители делают это, чтобы обеспечить постоянное и стабильное использование всеми пользователями и, возможно, поддерживать определенный уровень контроля над использованием своих продуктов.

На самом деле эту проблему очень легко решить, надо всего лишь поставить Root. «Rooting» – это сленговый термин для использования программных хаков, чтобы обойти эти ограничения в Android. Некоторые пользователи предпочитают делать это либо для расширения возможностей телефона, либо просто для хобби и для экспериментов. Процесс варьируется, но обычно включает загрузку специального файла из Интернета и установку его на телефоне с внешней карты памяти. Подробные инструкции по подключению определенного телефона Android широко доступны в Интернете, хотя пользователи должны знать, что есть риск для этого. Во многих случаях Root аннулирует гарантию телефона, а также существует вероятность того, что устройство может оказаться бесполезным, если что-то пойдет не так.

Android предлагает пакет разработки программного обеспечения (SDK), который поможет разработчикам создавать функциональные приложения. Хотя некоторые части операционной системы Android защищены в соответствии с лицензией Apache для защиты от приложений, которые могут поставить под угрозу базовые функции телефонов, большая часть из них выпущена в соответствии с лицензионным соглашением, которое предлагает разработчикам свободно вносить изменения. SDK включает в себя обширные ресурсы и инструменты для облегчения такого рода развития. Он регулярно обновляется вместе с самой операционной системой, и приоритет отдается поддержке работы как с более новыми, так и с более старыми версиями.

Android SDK (Software Development Kit) – пакет инструментов помогающих в проектировании и разработке программных продуктов под платформу Android. Android SDK включает в себя следующее:

- необходимые библиотеки ;
- отладчик;
- эмулятор;

- соответствующая документация для интерфейсов прикладных; программ Android (API);
- пример исходного кода;
- учебники для ОС Android.

Каждый раз, когда Google выпускает новую версию Android, также выпускается соответствующий SDK. Чтобы иметь возможность писать программы с последними функциями, разработчики должны загрузить и установить SDK для конкретного телефона. Среды разработки, совместимые с SDK, включают операционные системы, такие как Windows (XP или новее), Linux (любой недавний дистрибутив Linux) и Mac OS X (10.4.9 или новее). Компоненты Android SDK можно загрузить отдельно. Сторонние дополнения также доступны для загрузки. Хотя SDK можно использовать для написания мобильных приложений в командной строке, наиболее распространенным методом является использование интегрированной среды разработки (IDE). Официальной средой является Android Studio. Однако другие IDE, такие как NetBeans или IntelliJ, также будут работать. Большинство этих IDE предоставляют графический интерфейс, позволяющий разработчикам быстрее выполнять задачи разработки. Поскольку приложения для Android написаны в коде Java, у пользователя должен быть установлен Java Development Kit (JDK).

Android – самая используемая операционная система на планете. Фактически, у платформы есть своя экосистема. У каждой из версий Android есть как плюсы, так и свои недостатки. Поэтому у каждой версии есть свои поклонники.

Очень большая проблема Android это количество устаревших устройств работающих под этой ОС.

Google не виноват в конкретной проблеме устаревшей установки Android, производители устройств, но это не мешает ей быть вредным для ОС. Числа, выпущенные Statista, показывают, что 25% устройств Android, которые в настоящее время используются, работают с устаревшей версией ОС с конца 2014 года.

Как вы можете видеть на изображении (рисунок 1.1), IOS имеет только 7% пользователей в версиях операционной системы до 2016 года. Google имеет 17%, устройств с установленной ОС старше 2014 года, что является поистине гигантским количеством очень устаревших телефонов и планшетов.

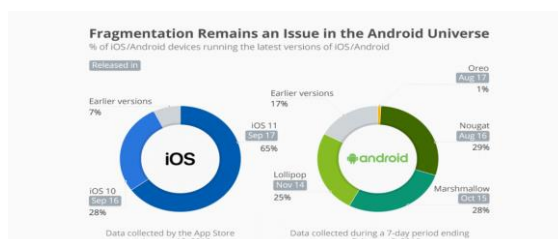


Рисунок 1.1 – Диаграмма использования обновлений

Самый большой процент устройств (29%) приходится на Nougat, который был выпущен в августе 2016 года и Marshmallow (28%), который был выпущен в октябре 2015 года. Это 57% всех устройств Android, на которых установлена ОС старше года. Только 1% пользователей Android находятся на Oreo, последней версии Google, в то время как 65% пользователей Apple находятся на последней версии, IOS 11. Эти две версии их соответствующих операционных систем были выпущены в течение месяца друг другом летом 2017 года.

Для многих пользователей обновления приходящие на телефоны с Android не особо важны, как IOS от Apple. Теперь пользователи могут купить последний Android-телефон, увидеть одно крупное обновление ОС и ничего больше. Такие компании, как Samsung, Sony и LG, постоянно выпускают обновления для системы безопасности – это здорово – но ваш новый Samsung Galaxy S9 может получить обновление для Android P (в этом году) и никогда не увидит Android Q. Пользователи Apple могут быть уверены, что они получат как минимум два года обновлений, хотя компания никогда не заявляет, как долго она намерена поддерживать устройства.

IOS 11 поддерживается на таких устройствах, как iPad Air и iPhone 5S, с 2013 года, это пятилетний пробег, что неплохо. Так что, пока Apple не идеальна, на экране ограничения на сторонние экраны, пользователи действительно получают современный телефон за приличное количество времени. Многие могут утверждать, что Apple имеет гораздо меньшее количество устройств для управления. Конечно, это правда, но на самом деле это не оправдывает медлительность многих производителей устройств Android при поддержке их собственных устройств. Samsung, Sony, LG и другие имеют небольшое количество телефонов, которые они выпускают каждый год, так неужели так сложно поддерживать их с обновлениями Android?

Я подозреваю, что правда заключается в том, что с двухгодичным циклом обновления означает, что производители телефонов Android не видят необходимости вкладывать деньги в цикл обновления программного обеспечения. И, более того, многие могут (ошибочно) думать, что это замедлит продажи новых устройств. Я это прекрасно понимаю, но Apple вроде показывает, что желание купить новое устройство, не привязано к обновлениям ОС.

Конечно, можно утверждать, что есть хорошие новости для Android. Старые устройства остаются в использовании в течение очень долгого времени. Это правда, и, конечно же, актуальная ОС это не единственная проблема, когда дело доходит до безопасности. Однако эта проблема, в общем, усложняет работу разработчиков и почти наверняка будет иметь некоторые неотъемлемые проблемы безопасности. Например, разработчикам необходимо будет постоянно выпускать обновления, особенно для обеспечения безопасности, во множество разных версий. Это, вероятно, трудоемкий и дорогостоящий процесс.

Мировая доля рынка Android составляет 37,93%, а для Windows – 37,91%. Это означает, что Android является господствующей операционной системой, пусть и с минимальным отрывом [3]. Как минимум это свидетельствует о стабильной динамике роста и в долгосрочной перспективе Android способен сместить Windows еще ниже. Всего за 5 лет популярность Windows упала в три раза (рисунок 1.2).



Рисунок 1.2 – Доля рынка операционных систем

Доля использования Интернета, рассчитанная StatCounter, объединяет различные аппаратные платформы, включая ноутбуки, настольные компьютеры, планшеты и смартфоны. Генеральный директор StatCounter Aodhan Cullen назвал победу Android важной вехой в истории технологий, которая также знаменует собой конец всемирного лидерства Microsoft, которое они удерживали с 1980-х годов (рисунок 1.3).

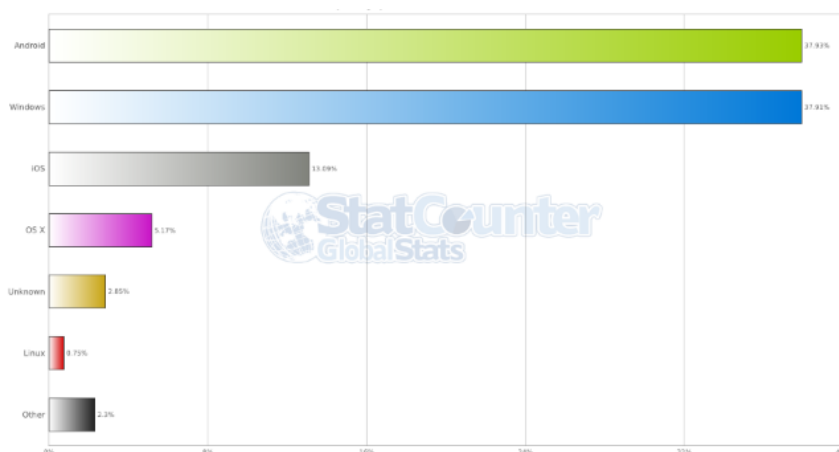


Рисунок 1.3 – Использование ОС

Таблица 1.1 – Доля рынка версий Android за январь 2018 года

Название	Версия	Процент использования
Marshmallow	6.0	28.6%↓
Nougat	7.0, 7.1	26.3%↑
Lollipop	5.0, 5.1	25.1%↓
KitKat	4.4	12.8%↓
Jelly Bean	4.1.x, 4.2.x, 4.3.x	5.6%↓
Oreo	8.0, 8.1	0.7%↑
Ice Cream Sandwich	4.0.3, 4.0.4	0.5%
Gingerbread	2.3.3 to 2.3.7	0.4%↓

Nougat стал самой популярной версией операционной системы Android. Marshmallow, который раньше занимал первое место, быстро теряет свой рынок и упал до 25,5%. С другой стороны, Oreo, похоже, догоняет темпы с почти 5% приростом с момента его запуска. Однако оставшиеся дистрибутивы продолжают терять свою долю в этом месяце. В таблице 1.2 приведены свежайшие данные за май 2018 года:

Таблица 1.2 – Доля рынка версий Android за май 2018 года

Название	Версия	Процент использования
Nougat	7.0, 7.1	31.1%↑
Marshmallow	6.0	25.5%↓
Lollipop	5.0, 5.1	22.4%↓
KitKat	4.4	10.3%↓
Oreo	8.0, 8.1	5.7%↑
Jelly Bean	4.1.x, 4.2.x, 4.3.x	4.3%↓
Ice Cream Sandwich	4.0.3, 4.0.4	0.4%↓
Gingerbread	2.3.3 to 2.3.7	0.3%↓

Начиная с KitKat и заканчивая Nougat, каждая новая версия Android, казалось, ударила по меньшему количеству устройств и делала это медленнее. Однако с этим значительным увеличением числа Oreo после всего лишь одного месяца, может ли быть, что Google, наконец, зажгла огонь, когда дело доходит до того, что OEM выпускает более своевременные обновления для устройств? Тем не менее, мы также знаем, что люди держатся на своих телефонах в течение более длительных периодов времени, а это означает, что количество активных устройств, работающих под управлением старых версий Android, остается высоким. По мере того, как цена на флагманские телефоны продолжает расти, это может замедлить скорость принятия Android

обновлений. На графике приведено изменение использования версий (рисунок 1.5).

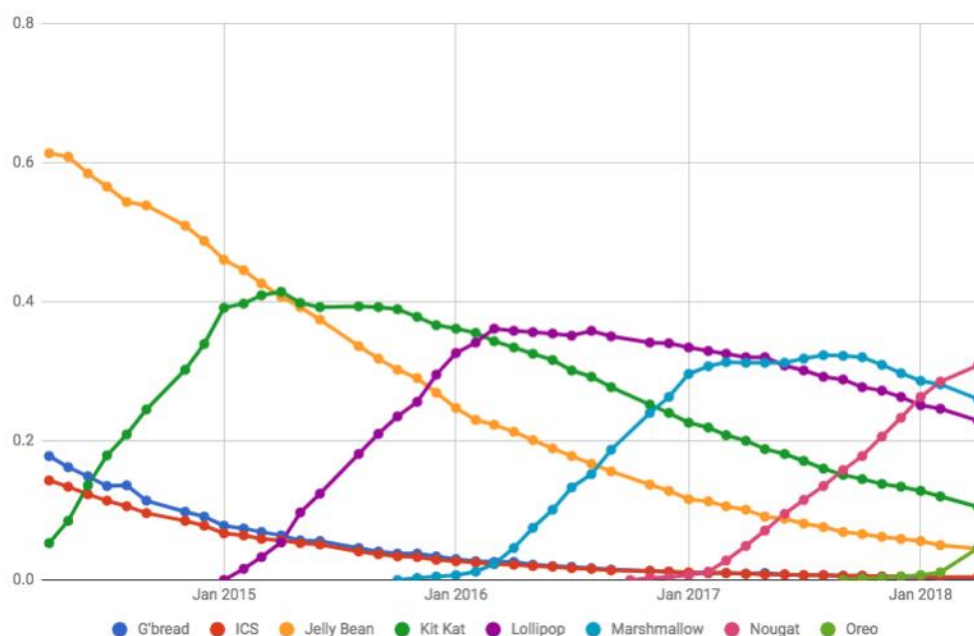


Рисунок 1.5 – Динамика использования версий

Увеличение скорости выпуска основных версий Android может быть одной из главных причин для формы графика выше. Jelly Bean отсутствовал около 16 месяцев до того, как появился KitKat. KitKat застрял чуть больше года, а затем Lollipop, который проскользнул за 11 месяцев до того, как Marshmallow попал на сцену. Затем Marshmallow был выпущен только за десять с половиной месяцев до появления Nougat в середине августа. Oreo является единственной версией программного обеспечения, которая способствовала росту тенденции, выпуская почти через год, на следующий день после Нуги.

Совсем недавно на конференции Google I/O компания представила обновление операционной системы Android. После прошлогоднего выпуска Oreo, 2018 год является годом Android P. Мы все еще находимся на некотором отдалении, прежде чем Google выпустит новое программное обеспечение для всех пользователей, но даже в его нынешней форме P выглядит многообещающе. Эта комбинация тапов и свайпов немного запутанна на момент бета тестирования, но ожидается, что Google развернет более уточненную версию этого в более поздних версиях Developer Preview или в окончательной сборке. Вы можете использовать традиционные три кнопки, если не привыкли к жестам, но довольно очевидно, что это будущее которое Google хочет для Android.

Android P не так сильно изменился в отношении визуальной части, как мы видели с переходом от KitKat до Lollipop, но по сравнению с Oreo, есть некоторые элементы, которые заметно отличаются.

На первый взгляд, такие вещи, как красочные значки в настройках, круговые значки быстрых настроек и закругленные углы для всего меню выглядят просто отлично. Но с другой стороны требуется немного времени, чтобы привыкнуть к этим элементам.

Что-то еще, что вы заметите в Android P, – это то, насколько он жив. Между новыми жестами и обновленной анимацией Android движется так, как я никогда раньше не видел. Oreo был гладким и маслянистым, но Android P летает под кончиками пальцев таким образом, что его можно воспринимать только лично.

Google много говорил о том, чтобы помочь людям с их «цифровым благополучием» на конференции I/O в этом году, и многие из этих функций были интегрированы прямо в Android P. Пока этого нет в Developer Preview 2, в более поздней версии Android P представят новую систему под названием Android Dashboard. Android Dashboard предложит вам быстрый взгляд на то, как вы используете свой телефон, включая статистику о том, какие приложения вы используете больше всего, сколько раз вы включили экран, сколько уведомлений вы получили и сколько времени вы потратили на каждое приложение. Также присутствует функция под названием App Timers, которая ограничит использование определенного приложения после того, как вы потратите на него слишком много времени. Дополнительная фишка это инструменты для легкого включения режима «Не беспокоить» и переключения экрана на монохромные цвета, чтобы уменьшить интерес к телефону и наконец-то приступить ко сну.

Используя программные хитрости, компания старается выжать максимум из вашей батареи. Кажется, Google всегда пытается найти способы максимально увеличить время автономной работы телефона, и с Android P эти усилия присутствуют в новом режиме адаптивной батареи. Подобно тому, как Adaptive Brightness автоматически настраивает уровень яркости вашего дисплея на основе вашей среды и использования, Adaptive Battery изучит, как вы используете свой телефон и ограничивает использование ЦП для приложений, которые вы редко используете. Google отмечает, что адаптивная батарея может снизить загрузку процессора на целых 30%, и благодаря использованию Machine Learning программа будет становиться умнее, чем больше вы используете свой телефон.

Изучение данной ветви требовалось для более точной разработки приложения. Точное знание использования версий помогает создавать программы для более обширной аудитории. Например мы решили разработать приложение под последний дистрибутив Oreo. В таком случае мы теряем более 99% потенциальных пользователей. Такая стратегия может привести к тотальному краху нашего проекта. Вообще разработка это очень тонкий и ресурсоемкий процесс, в основном, в интеллектуальном плане.

Android от Google и Apple IOS – это операционные системы, используемые в основном на мобильных устройствах, таких как смартфоны и планшеты. Android, основанный на базе Linux с частично с открытым

исходным кодом, более похож на ПК, чем на IOS, поскольку его интерфейс и основные функции, как правило, более подвержены кастомизации. Тем не менее, исключительные элементы дизайна IOS иногда воспринимаются как более удобные для пользователя.

IOS и Android используют сенсорные интерфейсы, которые имеют много общего – свайпы, нажатия, мультитач и масштабирование. Обе операционные системы загружаются на рабочий стол, который похож на рабочий стол компьютера. Хотя основной экран IOS содержит только строки значков приложений, Android позволяет использовать виджеты, которые отображают информацию об автоматическом обновлении, такую как погода и электронная почта. Пользовательский интерфейс IOS оснащен док-станцией, где пользователи могут подключать свои наиболее часто используемые приложения.

Строка состояния работает в верхней части как на IOS, так и на Android, предлагая информацию, такую как время, Wi-Fi или сигнал сотовой связи, и время работы от батареи; на Android в строке состояния также отображается количество вновь полученных писем, сообщений и напоминаний.

Сравнение некоторых функций установленных на Android N (Nougat) и IOS 10. Основные преимущества Android N:

- поддержка разделения экрана;
- настройки доступны в панели быстрого запуска;
- возможность настройки главного экрана;
- поддержка Google Daydream, платформы виртуальной реальности.

Пользовательский опыт показывает перевес в пользу IOS. Интерфейс IOS более плавный и создает меньшую нагрузку на глаза (рисунок 1.6).

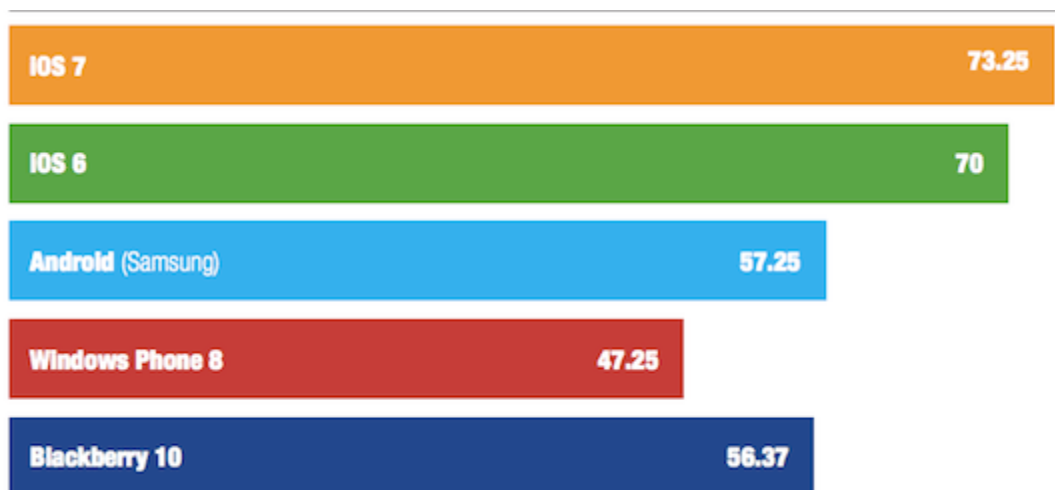


Рисунок 1.6 – Индекс удовлетворенности использования системы

Android получает приложения из Google Play, в которых в настоящее время доступно 600 000 приложений, большинство из которых будут работать на планшетах. Однако некоторые устройства Android, такие как Kindle Fire,

используют отдельные магазины приложений, которые имеют меньший выбор доступных приложений. Многие доступные изначально для IOS, приложения теперь доступны для Android, в том числе Instagram и Pinterest, а в более открытом магазину приложений Google также доступны некоторые эксклюзивные приложения, в том числе Adobe Flash Player и BitTorrent. Android также предлагает доступ к приложениям на базе Google, таким как Youtube и Google Docs.

Магазин приложений Apple в настоящее время предлагает 700 000 приложений, из которых 250 000 доступны для iPad. Большинство разработчиков предпочитают разрабатывать игры для IOS, прежде чем разрабатывать для Android. С недавнего обновления приложение Youtube стало недоступным на IOS, но IOS по-прежнему предлагает некоторые эксклюзивные приложения, в том числе популярную игру Infinity Blade и Twitter Twitterbot.

При сравнении магазинов приложений Google и Apple большинство популярных приложений доступны для обеих платформ. Но для планшетов есть больше приложений, разработанных специально для iPad, в то время как приложения для планшета Android часто расширяют версии приложений для смартфонов Android. Разработчики при запуске часто сосредотачиваются на одной платформе (обычно IOS), когда они впервые запускают свое приложение для смартфонов, потому что у них нет ресурсов для обслуживания нескольких платформ с самого начала. Например, популярное приложение Instagram, начато с IOS, и их приложение для Android появилось намного позже.

В отчете Mobile Experience от Crittercism, опубликованном в марте 2014 года, рейтинг Android KitKat был более стабильным, чем IOS 7.1. Другие выводы из отчета:

- Android 2.3 Gingerbread имеет самый высокий общий коэффициент крушения, на 1,7%. Другие версии Android – Ice Cream Sandwich, Jelly Bean и KitKat – имеют коэффициент сбоя 0,7%;

- IOS 7.1 имеет коэффициент сбоя 1,6%, а ставки для IOS 7.0 и IOS 5 составляют 2,1% и 2,5% соответственно;

- телефонные версии Android и IOS более стабильны, чем их версии для планшетов;

- коэффициенты сбоев для приложений варьируются в зависимости от категории – наиболее вероятны сбои в играх (4,4% крушения), а приложения для электронной коммерции имеют самый низкий коэффициент сбоя 0,4%.

iPhone 6s легко побил Samsung Galaxy Note 7 на тест скорости, который включал открытие нескольких приложений один за другим, циклическое использование набора приложений, в два раза. Один и тот же набор приложений использовался для обоих устройств. iPhone смог завершить два круга в минуту 21 секунду, пока Galaxy Note сделал то же самое за две минуты. Samsung Galaxy Note 7 обладает лучшими техническими

характеристиками и является гораздо более новым телефоном по сравнению с iPhone 6s.

Однако следует отметить, что разница может заключаться в том, как разработчики приложений разрабатывают свои приложения для Android и IOS. Просто потому что одни и те же приложения находятся на обеих платформах, не означает, что их внутренняя работа одинакова.

Хотя Google часто обновляет Android, некоторые пользователи могут обнаружить, что они не получают обновления на своем телефоне или даже приобретают телефоны с устаревшим программным обеспечением. Производители телефонов решают, предлагать ли и когда предлагать обновления программного обеспечения. Они могут не предлагать обновление до последней версии Android для всех телефонов и планшетов в своей линейке продуктов. Даже когда предлагается обновление, обычно это происходит через несколько месяцев после выхода новой версии Android.

Это одна из областей, в которой пользователи IOS имеют преимущество. Обновления обычно доступны для всех устройств IOS. Могут существовать исключения для устройств старше трех лет или для некоторых функций, таких как Siri, которые доступны для пользователей iPhone 4S, но не для более старых версий iPhone. Apple ссылается на аппаратные возможности, так как некоторые старые устройства могут не получать все новые функции при обновлении.

Широкое разнообразие устройств Android доступно в разных ценовых сегментах, размерах и аппаратных возможностях.

IOS доступен только на устройствах Apple: iPhone в качестве телефона, iPad в качестве планшета и iPod Touch в качестве MP3-плеера. Они, как правило, дороже, чем эквивалентные аппаратные средства, использующие Android.

Android позволяет пользователю отправлять один из нескольких самонастраиваемых текстов в качестве авторесурсов при отказе от вызова.

Телефонное приложение IOS обладает многими способностями, в том числе возможностью отвечать на телефонный звонок с помощью консервированного текстового сообщения вместо ответа или устанавливать напоминание о обратном вызове. Он также имеет режим «Не беспокоить».

Android позволяет пользователям регистрироваться на GTalk для мгновенных сообщений. IOS не предлагает собственный способ общения с пользователями, не принадлежащими Apple. Пользователи могут общаться с пользователями Apple, используя iMessage, или использовать приложения из Google для GTalk и Microsoft для Skype.

Google Hangouts на Android также можно использовать для видеочата, позволяя пользователям общаться через 3G или Wi-Fi. IOS использует Facetime, который может выполнять видеозвонки через 3G и WiFi. Однако он позволяет пользователям общаться с другими устройствами Apple.

IOS использует Siri, виртуальный голосовой помощник, способный понимать и реагировать как на диктовку, так и на голосовые команды. Siri

включает в себя множество функций, таких как чтение спортивных баллов и турнирной таблицы, резервирование в столика ресторане и поиск фильмов в местном театре. Вы также можете диктовать тексты и электронные письма, планировать события календаря и взаимодействовать с интерфейсом автомобильного аудио и навигации.

Android предлагает аналогичного помощника Google Now, который обладает вышеуказанными возможностями, плюс может отслеживать ваш календарь и давать вербальные напоминания, когда пришло время уйти. Это позволяет выполнять голосовые поиски и диктовки.

Такие приложения, как Google Maps, Waze и Bing, доступны как для IOS, так и для Android. Когда в декабре 2012 года Google выпустил приложение карт для IOS, версия IOS превзошла версию, доступную для Android с точки зрения возможностей, дизайна и простоты использования. Ожидается, что Android-версия не останется. Собственное приложение Apple для сопоставления, которое поставляется вместе с каждым устройством IOS, было широко распространено, когда оно было запущено с IOS 6.

Android использует Google Chrome как свой веб-браузер, в то время как IOS использует Safari. Оба интернет-браузера похожи по качеству и способностям, а Google Chrome также доступен для IOS. Safari недоступен для Android.

Android интегрирован с Facebook, позволяя пользователям обновлять свои статусы или загружать изображения из многих приложений, а также извлекать контактные данные у своих друзей из Facebook.

IOS также полностью интегрирована с Facebook, позволяя пользователям обновлять свой статус и загружать изображения из различных приложений, синхронизировать их контакты с Facebook и автоматически добавлять свои Facebook события в свой календарь IOS. IOS теперь предлагает гораздо более глубокую интеграцию с Facebook и Twitter из-за того, насколько сильно он взаимодействует с приложениями на IOS.

Android использует Google Кошелек, приложение, которое позволяет осуществлять мобильные платежи. Некоторые Android-телефоны оснащены чипом NFC (связь с ближним полем), который используется для совершения беспроводных платежей, просто нажав на номер телефона на счетчике. Эта услуга интегрируется с Google Кошельком, но не доступна на всех телефонах Android или беспроводных носителях. Мобильные платежи поддерживаются на устройствах с чипом NFC (например, Galaxy S4 и S5 от Samsung), при условии, что у них установлена версия Android, более поздняя, чем 4.4 KitKat.

Мобильная платежная система IOS называется Apple Pay. В то время как Apple Pay был запущен через год после использования функций «Tap2pay» на некоторых телефонах Android, его интеграция с идентификацией отпечатков пальцев, возможно, упрощает использование Apple Pay. Использование Apple Pay, безусловно, значительно выше. Фактически, Apple Pay заинтересовала мобильных платежей и побудила пользователей Android обнаружить эту функцию на своих телефонах, что привело к увеличению использования

Google Кошелек над NFC. IOS предлагает Passbook, приложение, которое собирает в одном месте билеты, наградные карты и кредитны/дебетовые карты.

Приложения Android изолированы от остальных ресурсов системы, если только пользователь не предоставляет доступ к другим функциям приложений. Это делает систему менее уязвимой для ошибок, но путаница разработчиков означает, что многие приложения запрашивают ненужные разрешения. Наиболее распространенная вредоносная программа на Android – это та, где текстовые сообщения отправляются на номера премиум-класса без ведома пользователя и отправка личной информации неавторизованным третьим лицам. Поскольку это более популярная операционная система для смартфонов, она, скорее всего, будет в центре внимания атак.

Вредоносные программы с меньшей вероятностью записывают приложения для IOS из-за обзора Apple всех приложений и проверки подлинности издателей приложений. Однако, если устройство IOS заблокировано и приложения установлены из-за пределов магазина Apple, он может быть уязвим для атак и вредоносного ПО. И IOS, и Android также уязвимы для ошибок, например, при сбое в воспроизведении определенного видео, что является типичной ошибкой программного обеспечения, которая затронула как IOS, так и Android-устройства.

В реальном мире безопасность устройства Android или IOS настолько же хороша, как и обновления программного обеспечения, которые были применены к нему. Именно здесь IOS выигрывает из-за фрагментарного характера экосистемы Android. Apple выпускает обновления программного обеспечения и делает их доступными для всех устройств IOS одновременно. На Android Google выпускает обновления программного обеспечения и исправления безопасности для устройств Nexus. Устройства других производителей отстают, потому что производитель должен получать эти обновления безопасности от Google и применять их на своих устройствах «в дикой природе». Практически все производители плохо справляются с этим. Большинство из них не выпускают исправления для устройств старше 12–18 месяцев. Даже когда они это делают, эти обновления безопасности выпускаются через несколько месяцев после их приема устройствами Nexus.

Android устройства менее безопасны, это подтверждается программой бонусов; компания, которая получает взломы безопасности от хакеров и продает их правительствам, имеет награду за 0-дневные (т. е. ранее неизвестные) эксплойты для IOS (1,5 млн. долл. США), Android (200 000 долл. США) и Flash (80 000 долл. США). Суммы щедрот являются приблизительным значением того, насколько легко взломать эти платформы на практике.

И IOS и Android «уязвимы» для определенного типа утечки личной информации: приложение, установленное на любой из платформ, может получить список всех других приложений, установленных на одном устройстве. Это означает, что ваше приложение калькулятора может узнать,

что вы используете Tinder и передаете эту информацию обратно своему издателю, который затем может использовать эту информацию любым способом. В ноябре 2014 года Twitter объявил, что теперь он отслеживает список приложений, установленных пользователями на своих телефонах. Twitter – далеко не единственная компания, которая это делает.

IOS выигрывает, когда дело доходит до защиты личной информации пользователей. Пока Android Marshmallow не был выпущен в 2015 году, при установке приложений на Android пользователю были предоставлены все разрешения, которые запрашивает приложение. Это было предложение «все или ничего». Пользователь может принять решение о приеме приложения для получения разрешений или вообще не устанавливать приложение. Разработчики приложений используют эту «функцию» и запрашивают много пользовательской информации. Например, мобильное приложение Pandora на Android запрашивает разрешения для вашего идентификатора Google, контактов, календаря, фотографий, мультимедиа, файлов и даже информации о вызовах (рисунок 1.7).

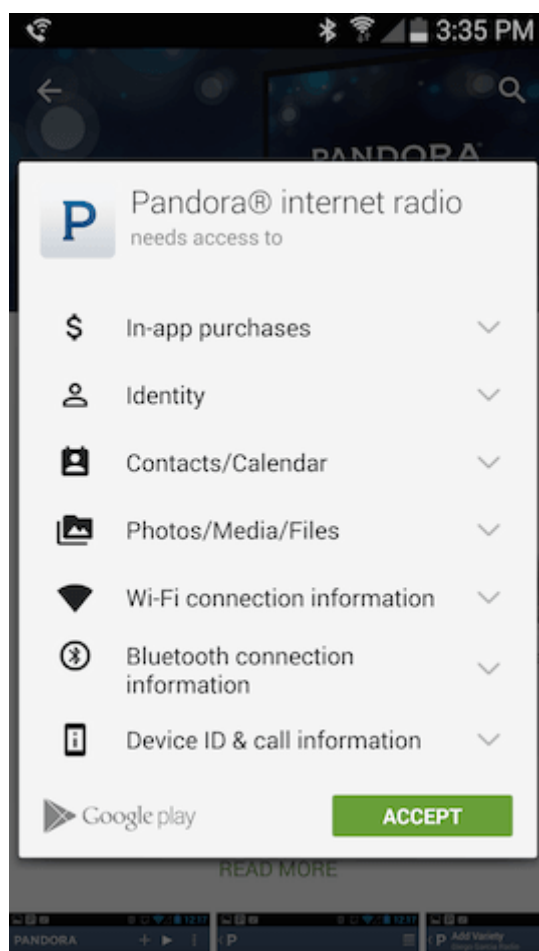


Рисунок 1.7 – Приложение Pandora

Приложение Pandora на IOS не получает таких разрешений. После того, как он будет установлен и открыт пользователем, приложение IOS может

запрашивать дополнительные разрешения, такие как местоположение и доступ к контактам. Но пользователь может отклонить эти запросы на разрешение. Даже после одобрения запросов на разрешение пользователи IOS могут быстро взглянуть на то, какие приложения имеют доступ к своим контактам и данным о местоположении, и отключить доступ для приложений, с которыми они больше не хотят делиться этими данными.

Android M (или Marshmallow) ввел новый режим разрешений, при котором приложения могли запрашивать разрешения по мере необходимости. Тем не менее, большинство приложений для Android по-прежнему используют подход, запрашивающий разрешения. Хотя можно управлять разрешениями приложений на Android на более узком уровне, этот параметр глубоко погружен в настройки [6].

Приложения Android созданы с использованием C, C++ и Java. Это «открытая» платформа; любой может бесплатно скачать исходный код Android и Android SDK. Любой может бесплатно создавать и распространять приложения для Android; пользователи могут бесплатно скачивать приложения из официального магазина Google Play. Тем не менее, для разработчиков, которые хотят публиковать свои приложения (бесплатные или платные приложения) в официальном магазине Google Play, взимается единовременная регистрационная плата в размере 25 долларов США. Приложения, опубликованные в Google Play, проходят обзор от Google. Android SDK доступен для всех платформ – Mac, ПК и Linux.

Приложения IOS запрограммированы с использованием Objective-C. Разработчики должны каждый год платить \$ 99 за доступ к SDK IOS и право публиковать в магазине приложений Apple. IOS SDK доступен только для платформы Mac.

Некоторые платформы разработки приложений, такие как Titanium Appcelerator и PhoneGap, предлагают способ кода один раз (скажем, в Javascript и/или HTML) и превращают платформу в «родной» код для платформ Android и IOS.

В Beyond Flat основатель SeatGeek Джек Гетцингер излагает много различий в том, как Android и IOS подходят к их эстетике дизайна и что это означает для разработчиков приложений. Например:

- кнопки: Android кнопки обычно монохромны, и, по возможности, склонны к использованию иконописи. Стандартная кнопка IOS 7 – простой монохроматический текст без фона или границы. Когда IOS 7 использует границы кнопок, они, как правило, довольно просты;

- панель навигации: панель навигации в IOS обычно представляет собой кнопку «Назад», которая ссылается на предыдущий экран. В Android навигационная панель обычно имеет несколько кнопок действий;

- intents: Intents на Android позволяет приложениям гибко взаимодействовать друг с другом. Например, приложения могут «регистрировать» себя как способные делиться, что позволяет пользователю делиться этим приложением с любым другим приложением;

IOS плюсы и минусы:

– массивная экосистема приложений: отличное преимущество для планшетных приложений, а на смартфонах популярные приложения обычно доступны для обеих платформ;

– более глубокая интеграция с Facebook и Twitter: проще публиковать обновления и делиться в социальных сетях с помощью IOS, чем Android, из-за того, насколько глубоко интегрированы эти платформы с IOS;

– приложения IOS, такие как Passbook, FaceTime и мобильное платежное приложение Square (доступно на IOS 3GS, 4,4S, 5 и выше, на Android доступно только для лимитированных версий);

– интерфейс заблокирован: ограниченные параметры настройки для домашних экранов; разрешены только строки значков приложений. Беспроводные несущие не устанавливаются сторонними приложениями. Пользователи могут устанавливать приложения только из App Store;

– обновление программного обеспечения: Apple предлагает обновления программного обеспечения для всех устройств, на которых установлено оборудование, способное обрабатывать новое программное обеспечение. Это означает, что устройства остаются в силе с программными функциями не менее двух–трех лет;

– улучшенный контроль конфиденциальности: IOS предлагает лучший контроль над приложениями доступа к личной информации пользователей, такой как контакты и местоположение.

Преимущества и недостатки Android

– массивный выбор оборудования: большое количество устройств Android доступно в разных ценовых категориях с различными аппаратными возможностями, размерами экрана и функциями;

– очень настраиваемый пользовательский интерфейс: на главном экране можно настроить не только иконки приложений, но и виджеты, которые позволяют пользователю оставаться на связи или информировать пользователя. Другие примеры включают SwiftKey, который изменяет клавиатуру вашего смартфона Android и приложения, которые имитируют старые игровые консоли. Google имеет меньше ограничений, чем Apple, о том, какие приложения он допускает в свой магазине Play Market. Кроме того, вы можете установить Android-приложения из других мест, кроме магазина Google Play.

Пользовательский интерфейс Android (UI) основан на концепции прямого манипулирования – это означает, что он использует различные сенсорные жесты, такие как прокрутка, ущемление и нажатие, чтобы манипулировать на экранных объектах. Также предоставляется виртуальная клавиатура для ввода текста. Сегодня, помимо устройств с сенсорным экраном, Google развивает Android TV для телевизоров, Android Auto для автомобилей и Android Wear для наручных часов. Android является самым крупным из всех мобильных операционных систем по состоянию на 2015 год.

Поскольку Android выпускается Google под лицензиями с открытым исходным кодом, он стал чрезвычайно популярен среди тех компаний, которые нуждаются в готовой, а также недорогой настраиваемой ОС для высокотехнологичных устройств, которые можно быстро популяризировать. Сегодня большое сообщество разработчиков, а также энтузиастов технологий используют открытый исходный код для управления проектами под руководством сообщества, одновременно добавляя новейшие и профессиональные функции для продвинутых пользователей. Честно говоря, Google Android привлек внимание, а также воображение пользователей, а также разработчиков по всему миру.

Говоря откровенно, Android OS обслуживает более широкую аудиторию, чем другие операционные системы для смартфонов. Эта более широкая аудитория включает энтузиастов книжных червей, энтузиастов хардкорных игр, жадных видео и наблюдателей за фильмами. Операционная система Android работает с учетом удобства пользователя. Android пакеты для многих.

Когда мы пишем об Android, мы не можем вычеркнуть Apple из уравнения. Apple может быть крупным продавцом по всему миру, но это не закрывает комнату для конкуренции. С этой целью давайте изучим некоторые статистические данные. В бизнесе смартфонов Apple владеет более чем 90% всей прибыли. Эта отрасль является одной из видов, в которой ни одна другая компания не делает ее большой в этом сегменте. Особенно после недавнего запуска iPhone 8 и X этот телефон продавался как сумасшедший.

Часто говорят, что Apple является царем индустрии смартфонов. Тем не менее, статистика также указывает на глобальность, Android делает гораздо больше с точки зрения объема, чем iPhone. Согласно данным IDC, поставки Android в прошлом году должны были превысить 1,15 миллиарда долларов, а доля рынка составит около 79,4%. Другими словами, продажи Android в пять раз больше, чем продажи iPhone на международном рынке.

По мнению некоторых аналитиков, многие пользователи iPhone нуждались в более крупных экранах iPhone, но Apple слишком долго придерживалась 4,0-дюймовых смартфонов. В то время как популярность Android продолжала расти из-за более крупных 5,0-дюймовых экранов, которые она представила.

Это еще одна причина, почему Android популярнее среди других продуктов, таких как iPhone. Сегодня Android поддерживает миллионы мобильных устройств по всему миру. Android действительно предлагает передовую платформу для разработки и создания приложений и игр для пользователей по всему миру.

Таким образом, Android стал самой быстрорастущей мобильной ОС с поддержкой более 300 аппаратных средств, программного обеспечения, а также партнеров-перевозчиков. Кроме того, из-за открытости Android, это стимулирует активное потребление приложений, так как каждый месяц загружаются миллиарды приложений и игр. Благодаря своим инновациям и

быстрой эволюции технологий Android пользователи и разработчики могут оставаться впереди с мощными, а также дифференцированными приложениями.

Еще одна причина, по которой Android пользуется популярностью – это мощная инфраструктура разработки. Разработчики получают все необходимое для создания лучших в своем классе приложений. Затем они могут развертывать приложения для миллионов пользователей по всему миру. Кроме того, Android Developer Tools предлагает полную интегрированную среду разработки Java (IDE) с помощью передовых и высокоточных функций для разработки, отладки и упаковки приложений для Android. Вы можете использовать IDE для разработки на любом доступном устройстве Android. Android IDE также позволяет создавать виртуальные устройства, которые эмулируют любую конфигурацию оборудования [4].

Кроме того, магазин Google Play является открытым рынком для распространения этих приложений на мгновенной основе. Так как Google Play также предлагает открытую платформу для распространения приложений для Android приложений, вы можете легко настроить базу с огромной установленной базой Android. Текущие данные оценивают 1,5 миллиарда загрузок в месяц, поэтому вы получаете возможность монетизировать то, что лучше всего подходит для вашего бизнеса. Например, вы можете оценить свое приложение или сделать его доступным бесплатно. Вы также можете добавить в приложения или подписки, чтобы получить самые высокие обязательства и доходы.

Философия, которой придерживается Google, заключается в привлечении открытого рынка для улучшения участника. Таким образом, Google Play ставит пользователей под контроль того, как они хотят продавать свои продукты, которые могут публиковаться по мере необходимости и часто, как они хотят, в том числе желающих клиентов.

Google Play также помогает издателям, которые не стремятся наращивать свою клиентскую базу, помогают им создавать видимость, а также помогают в приложениях и брендах. По мере роста популярности приложений они получают более высокое размещение в «топ-диаграммах» Google, а также в рейтингах. Кроме того, Google Play обычно устанавливается на миллиарды устройств Android по всему миру. Таким образом, Google Play может стать движущей силой бизнеса на базе Android.

1.3 Основные аспекты разработки приложения

Разработчиком Android обычно является кто-то, кто разрабатывает приложения для использования на сотовых телефонах, которые используют операционную систему Android или IOS. Эти телефоны, как правило, могут загружать и использовать множество различных приложений, называемых приложениями, которые могут повысить полезность телефона, предоставляя множество новых форм функциональности. Эти приложения могут включать

в себя такие вещи, как карты, которые могут включать данные с телефона со встроенным программным обеспечением глобальной системы позиционирования (GPS), а также игры, локаторы и обзоры ресторанов и расширенное воспроизведение мультимедиа. Разработчик Android обычно работает как часть команды, хотя отдельные программисты могут также разрабатывать приложения [2].

Поскольку беспроводные телефоны становятся все более мощными, они способны запускать приложения требующие больших ресурсов, за счет этого заметно улучшились программные продукты. Новые телефоны часто разрабатываются со стоковыми приложениями и различными сервисами, и могут воспользоваться многими другими программами и приложениями, которые затем могут работать на телефоне. Различные компании используют разные системы на своих телефонах, и одна из таких систем – это ОС Android, которая может использовать приложения, разработанные и распространяемые через «Google Play Market». Эти программы создаются разработчиком Android, который может быть компанией или просто отдельной единицей.

Типы программ, которые могут быть спроектированы разработчиком Android, сильно различаются и включают в себя все: от интерактивных игр до медиаплееров, а также приложения, которые взаимодействуют и обновляются на основе популярных веб-сайтов и даже электронных книг и глобального навигационного программного обеспечения. Разработчик Android может быть крупной организацией, в которой работают многочисленные сотрудники для создания этих приложений или отдельного разработчика, который делает приложения для личного использования и удовольствия других. Код, необходимый для разработки Android, в доступен для всех, кто заинтересован в разработке приложений для телефонов, использующих платформу [5].

Один из ключевых вопросов, который продолжает дезориентировать большинство предприятий: сколько времени потребуется на разработку мобильного приложения, и сколько денег оно будет стоить? Недавно MGI Research провела исследование индустрии мобильных приложений, посвященное разработке, управлению и безопасности мобильных приложений [3].

Последнее исследование мобильных приложений было ориентировано прежде всего на крупные и средние предприятия и бизнес единицы с ответами как от IT-организаций, так и управления бизнесом. Был проведен опрос по ряду ключевых вопросов об их восприятии успеха или неудачи в мобильных приложениях; время, стоимость и навыки, необходимые для разработки мобильного приложения; и в каких секторах их компании способны к конкуренции. Целевыми вопросами были:

– степень, в которой предприятия чувствуют, что они добиваются ощутимого прогресса со всеми аспектами мобильных приложений от проектирования и разработки до развертывания, управления, безопасности, стратегии, кадрового обеспечения, показателей и бюджетирования;

– каково видение компании различий выдающегося проекта от провального.

Когда дело доходит до временных рамок, время разработки проекта, как правило, относительно ограничено в отношении времени выхода на рынок. У почти половины (48 процентов) компаний, создание занимает шесть месяцев или менее (рисунок 1.8).

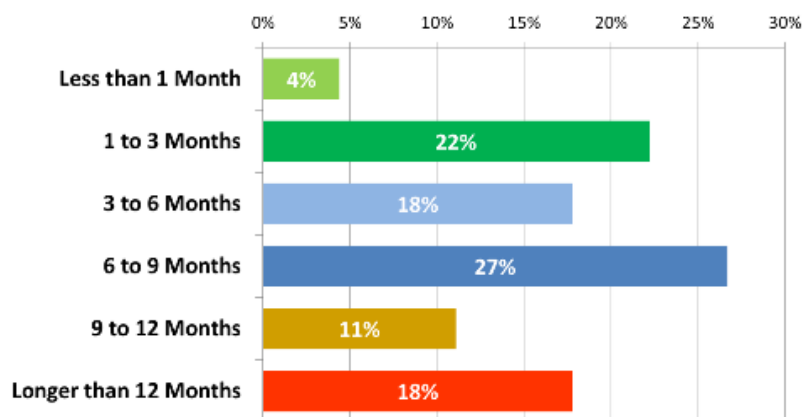


Рисунок 1.8 – Время разработки приложения

В совокупности 75% всех компаний сообщили о девятимесячном или менее продолжительном цикле разработки мобильных приложений. Это в значительной степени соответствует исследованиям прошлых лет и отражает быстрые темпы изменений в мобильной индустрии, а также прогресс во внедрении различных мобильных средств разработки.

Несмотря на короткие временные рамки для разработки, которые, как правило, выгодно отличаются от временных рамок для других проектов развития ИТ, когда дело доходит до своевременной доставки, проекты мобильных приложений характеризуются очень низким соответствием заданному времени. Всего 11% (1 из 10) компаний укладываются в установленные дедлайны (рисунок 1.9).

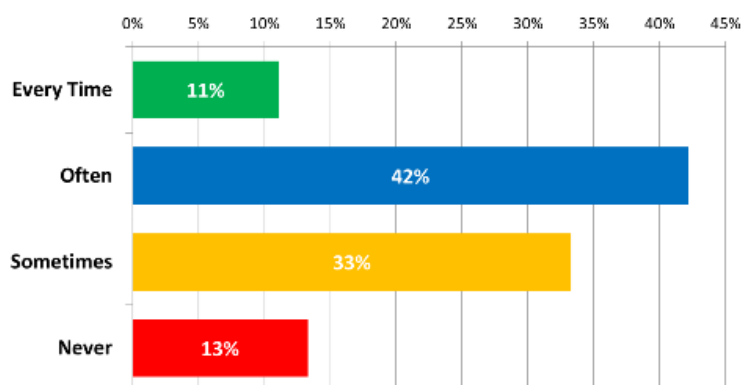


Рисунок 1.9 – Процент компаний уложившихся в установленные временные рамки

Стоимость и финансирование проекта – это две области, которые вызывают серьезные вопросы для всех предприятий, особенно для тех, кто только начинает заниматься мобильным развитием. Результаты исследования показывают, что в среднем 68% компаний тратят \$ 100 тысяч или меньше на мобильное приложение, а 48% (примерно половина) тратят меньше 50 тысяч долларов США за приложение (рисунок 1.10).

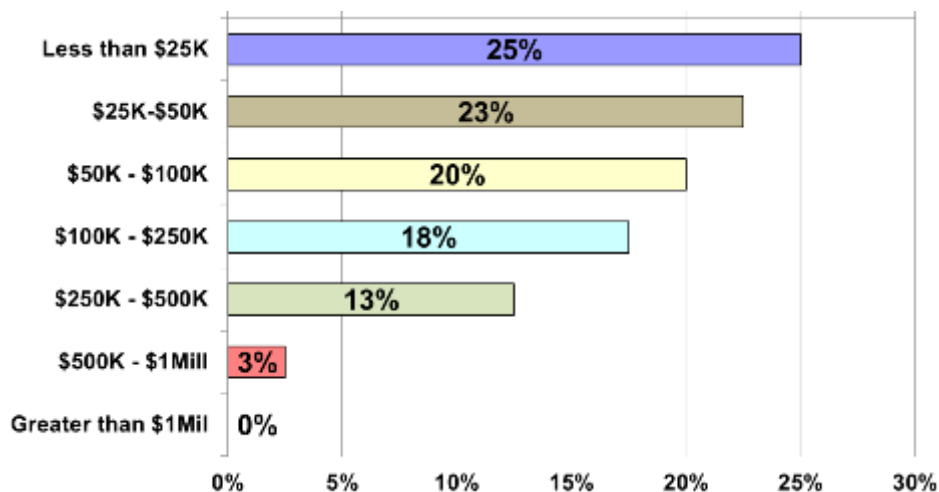


Рисунок 1.10 – Затраты на разработку

Когда это связано с бюджетом, только один из семи проектов мобильных приложений (около 16 процентов) последовательно доставляет результаты в первоначальном бюджете (рисунок 1.11).

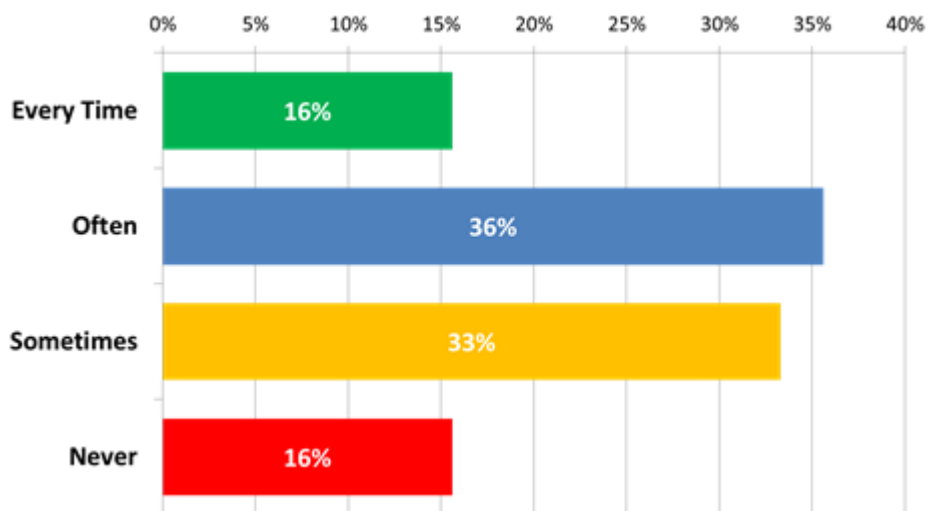


Рисунок 1.11 – Как часто компании укладываются в выделенный бюджет

Отсутствие бюджетной точности – это серьезная проблема, которая должна быть решена, по мере развития компании в плане получения опыта

работы с мобильной разработкой, и улучшения, а также усложнения используемых инструментов.

Между компаниями, достигшими серьезного успеха в разработке и теми, кто сообщал о средней удовлетворенности, есть существенная разница. В группе лидеров (те, у кого был «огромный успех» с разработкой), 20 процентов проектов мобильных приложений поставляются в рамках первоначального бюджета, а те, у кого не было «огромного успеха» с мобильными приложениями, сообщают, что только восемь процентов проектов мобильных приложений поставляются в рамках первоначального бюджета.

Очевидно, опыт в разработке мобильных приложений также является большим фактором. Среди тех, кто считает себя успешными на этом поприще, более 30% имеют опыт работы с мобильными приложениями в течение пяти или более лет.

В сочетании с несоблюдением временных рамок при разработке мобильных приложений это создает довольно печальную картину эффективности и гибкости мобильной разработки. Похоже, что существует большой разрыв между тем, что конторы считают эталонным стандартом для разработки мобильных приложений и текущих возможностей управления внутренними и внешними ресурсами. Уровень опыта работы с мобильными приложениями может дать частичное объяснение этого разрыва.

Установление четких рамок относительно времени доставки и стоимости имеет решающее значение, прежде чем приступить к мобильному проекту. Необходимо устранить пробелы между ожиданиями пользователей и реальностью того, чтобы обеспечить адекватное финансирование и долгосрочный успех.

Хотя приложения B2E могут быть быстрыми и относительно недорогими, большинство приложений для B2B и B2C требуют большего финансирования и времени разработки, чем обычно считает средний бизнес исполнитель.

Простота использования – признак успешных приложений. Почему? Занятые люди в 21 веке не успевают потратить время на изучение, работа приложения. Если приложение не является интуитивно понятным или не отвечает требованиям пользователя (сбои, медленное время загрузки и т. д.), мобильные пользователи просто будут искать другие варианты. В конце концов, есть миллионы приложений, что означает, что, вероятно, есть альтернатива вашим, к которым пользователи обратятся, если ваш пользовательский интерфейс не оправдает ожиданий.

Технологии предназначены для решения проблем и облегчения жизни. В качестве основы современного технологического мира мобильные приложения должны также сосредоточиться на решении проблем пользователя. В зависимости от отрасли и фокуса приложения, конкретная проблема должна быть решена. Это можно сделать, если получать больше информации от потенциальных клиентов о ваших продуктах и об их

ожиданиях от вашего продукта. Тем не менее, для создания мобильного приложения требуется видение картины рынка и ожиданий пользователя в целом.

Чтобы это увидеть следует изучить список 25 лучших мобильных приложений в вашем регионе. Facebook позволяет пользователям приложений оставаться на связи со своим миром, поскольку они живут активным образом жизни. YouTube позволяет просматривать контент с удобством со своего телефона. Эти приложения предоставляют функции, которые удовлетворяют требованиям пользователей смартфонов.

Мобильное приложение должно либо решить фундаментальную проблему, либо улучшить взаимодействие пользователя с компанией и отраслью. Каждый четвертый загружаемый мобильный контент не используется и, в конце концов, удаляется. Ваше приложение не должно быть частью этих ужасных статистических данных, если оно добавляет ценность конечному пользователю в вашей отрасли.

Если два приложения решают одну и ту же проблему, но одно делает это за меньшее количество шагов, ожидайте, что приложение будет иметь больший успех. Может возникнуть соблазн просто бросить кучу функций в приложение, чтобы сделать его самым многофункциональным приложением на рынке, но нужно понять, будут ли какие-либо из этих функций оценены и использованы потребителями. Если нет, вы можете просто тратить свое время и создавать сложные приложения, которые не будут оценены по достоинству на рынке.

Наконец, каждое приложение должно быть замечено. На переполненном рынке, с миллионом приложений, соперничающих за внимание и увеличение доли рынка, трудно привлечь внимание потенциальных клиентов без детального маркетингового плана.

Реализация этого плана должна начинаться до того, как приложение будет выпущено. Успешные релизы приложений происходят, когда маркетинг продолжается по мере того, как приложение развивается. Важно знание, где вы планируете распространять приложение (IOS, Android и т. д.), Так как это будет информировать вас о том, как ваш бренд должен продвигать приложение перед запуском, во время и после запуска. Другими словами, ваша маркетинговая стратегия должна отражать стадию разработки приложения.

Сегодня мы используем мобильные приложения практически для каждой задачи – будь то наш будильник, просмотр телевидения или просмотр последнего отчета, отправленного вашим боссом. Когда компании разрабатывают мобильное приложение, крайне важно, чтобы клиенты находили его простым в использовании. Однако данные из Отчета о мировом качестве 2016–2017 показали, что мобильное тестирование по-прежнему «рассматривается как относительно новый навык», несмотря на вездесущее использование мобильных устройств. В отчете было установлено, что самая большая проблема, с которой организации сталкиваются во время

тестирования, – это отсутствие правильного процесса или методологии, за которыми следует отсутствие экспертов по тестированию.

Передовые методы тестирования мобильных устройств могут преодолеть эти трудности. Команда оценки качества (ОК) часто является забытой частью разработки мобильных приложений, но их работа имеет решающее значение для подтверждения ожиданий пользователей.

Фрагментация устройства и операционной системы является основной задачей, стоящей перед командами ОК. (Например, Android Marshmallow – это OS, установленная на большую часть устройств, по данным на декабрь 2016 года). Эта фрагментация усугубляется частыми изменениями устройств и обновлениями ОС. Команды ОК должны подробно сравнивать, как приложения будут вести себя на нескольких устройствах с разными версиями операционных систем.

Вы можете пожертвовать охватом между комбинациями устройств и ОС; но когда вы уменьшаете количество типов устройств, с которыми вы тестируете, вы рискуете, что ваше приложение может не работать для некоторых клиентов.

Чтобы справиться с проблемой совместимости, есть три варианта:

- тестирование исключительно с использованием реальных устройств;
- тестирование исключительно с эмулированными устройствами;
- использование комбинации обоих.

Первым шагом, который следует определить при создании вашего плана, является подготовка списка целевых устройств, на которых необходимо провести тестирование вашего продукта. Он должен включать различные размеры и характеристики экрана (память, процессор и т. д.).

В плане тестирования вы можете включить реальные устройства и эмуляторы. Использование эмулятора в сравнении с реальным устройством имеет свои недостатки и преимущества. Эмулированные устройства просты в управлении, но их большой недостаток заключается в том, что использование эмулятора является синтетическим тестированием в отличие от использования реального устройства.

Помимо того, что приложение работает на разных устройствах и операционных системах, команда оценки качества должна проверить, как приложение реагирует на различные сетевые условия. Вот пять основных сценариев для тестирования:

- подключение к Wi-Fi;
- только соединение 4G;
- только 3G-соединение;
- в режиме самолета (или все соединения деактивированы);
- и, самое интересное, временные потери сигнала, имитирующие использование в реальном мире.

Помимо оценки приложения с использованием разных устройств и / или эмуляторов и анализа производительности в разных сетевых условиях, рекомендуется рассмотреть следующие четыре аспекта.

Прерывания. Все смартфоны многозадачны. Инженер команды ОК должен проверить, как приложение реагирует в различных ситуациях, например, когда устройство получает входящий вызов, SMS, уведомление о приложении, предупреждение о низком заряде батареи, событие календаря, уведомление о социальной сети, будильник или пользователь переключается между приложениями или приложениями.

Локализация. Является ли приложение локализованным? Тестеры должны проверять приложение на разных языках. Нужно убедиться, что каждая строка переведена на требуемый язык.

Потребление батареи. Следует убедиться что приложение потребляет энергию в пределах нормы.

Производительность. Важно сравнить выпущенную версию с текущей версией, чтобы убедиться, что производительность такая же, лучше или хуже. Установите приложение на старых устройствах, чтобы узнать, все еще работает на них. К основным аспектам оценки относятся:

- проверьте, как приложение реагирует на входящие телефонные звонки, SMS, MMS, твиты или другие уведомления. Проверьте расход батареи при использовании приложения;

- проверяйте использование батареи каждые 10 минут, чтобы узнать, потребляет ли приложение слишком много энергии;

- установите приложение на устройства с очень низким уровнем заряда батареи и проверьте, что происходит;

- проверьте использование памяти приложения. Если приложение хранит данные в локальной файловой системе, проверьте использование разных карт памяти. Подумайте, что произойдет, когда локальное хранилище будет почти заполнено – приведет ли приложение к сбою или сообщит пользователю правильное сообщение об ошибке;

- используйте приложение на разных носителях и сетевых подключениях, таких как GPRS, Wi-Fi или LTE.

Когда вы разрабатываете мобильное приложение, ваша команда оценки качества является важной частью процесса и поможет вам определить правильный план тестирования. Команда ОК играет решающую роль в обеспечении того, чтобы ваши клиенты и пользователи чувствовали себя комфортно с приложением и что у них хороший пользовательский интерфейс. Только при правильном плане тестирования вы можете обеспечить успешный выпуск и долгосрочную службу своего приложения.

Почти все приложения, которые находятся на торговой площадке, были созданы в коммерческих целях. Это значит, что приложение создавалось как средство для извлечения выгоды. Существует много способов заработать денег на разработке приложения, ниже представлены основные способы монетизации программного продукта.

Если для использования приложения нужно открывать, то встроенная реклама может быть отличным способом получения дохода. Для этого нужно встроить ссылки в код и прописать моменты выхода интерфейса рекламы, для

перехода на сайт рекламодателя, важно убедиться, что пользователи могут пропустить объявление, за каждый переход по ссылке, рекламодатель будет платить определенную сумму. Для такого способа важно, чтобы приложение завлекло, иначе потребитель просто выберет другой продукт.

Youtube запускает рекламу способную заинтересовать пользователя, которая не только помогает генерировать доходы, но и гарантирует, что пользователь не покинет платформу. Другое игровое приложение, Word Crossy, работает с этой формулой. Игра слов интересна, поэтому пользователь не против посмотреть рекламный ролик, как только одна игра заканчивается.

Идея заключается не в том, чтобы прервать пользователя во время использования приложения, а только по завершении определенного цикла. В конце сеанса воспроизводится реклама, а затем пользователь может продолжить использование приложения.

Баннерная реклама – отличный способ получить доход. Они будут частью приложения, в отличие от других объявлений, и они тоже принесут вам деньги.

Еще одним способом является продажа мерчендайза. Мерчендайз – это товар продвинутого бренда. Ярким примером мерчендайза являются футболки от различных поп артистов. Более близкий пример, это футболки Dota 2 и т. д. Недостаток этого способа заключается в том что для этого надо иметь сильный и популярный в массах, бренд. Но такой способ подходит только гигантам индустрии мобильной разработки и подходит больше всего для игр.

Отличным маркетинговым ходом является email рассылка. Это позволит увеличить охват аудитории и привлечь постоянный поток клиентов на программный продукт. Встает вопрос, как получить email? Делается это благодаря окну авторизации. Когда человек активирует свой аккаунт требуется ввести активную почту. Тут мы получаем его данные, ничего страшного в этом нет, ведь электронная почта находится в открытом доступе и это не является кражей личных данных.

Следующий способ это так называемый режим freemium приложения. Продукт доступен на площадке абсолютно бесплатно. При скачивании приложения пользователю доступны базовые функции которые удовлетворяют его потребностям. Но в то же время следует показать что приложения способно на большее стоит только заплатить. Как правило цена небольшая и для пользователя равна стоимости чашки кофе, но при большом количестве пользователей у разработчика выходит приятная сумма.

Evernote делает это довольно долгое время. У них есть много бесплатных функций, которые можно регулярно использовать с помощью приложения. Однако, чтобы поделиться аудио или другими такими функциями, вам необходимо обновить приложение до версии премиум-класса.

Некоторое время, используя приложение, вы довольно привыкли к этому, и вы знаете, что это лучшее решение для ваших нужд. Небольшая плата за премиум функции в месяц не кажется серьезной проблемой. Многие

пользователи склонны использовать премиальные функции в приложениях, когда это необходимо, и оплачивать их.

Это определено самая оптимальная стратегия получения прибыли от приложения, так как потребитель сам решает тратить ему деньги или нет.

В мобильных играх используется способ внутренних покупок. Таким способом пользуется значительная часть компаний разрабатывающих игровые приложения. Большинство игровых приложений, как правило, заманивают своих пользователей и зарабатывают деньги через покупки в приложении. Структура очень проста. До определенного момента игрок может проходить игру своими силами, но на каком-нибудь уровне требуется повышении мощностей, тут игрок и приходит к покупке внутренней валюты, которую в дальнейшем он может потратить на кастомизацию приложения.

Некоторые типы приложений используют так называемые подписки. Что это такое? Приложение также доступно бесплатно на торговой площадке, на определенный временной промежуток юзеру дается пробный период. По истечении пробного периода возможность использования приложения улетучивается и тогда программа предлагает купить подписку на определенное время, как правило на месяц, и обновлять ее по истечении срока оплаты. В основном такой метод используют при разработке крупных проектов, таких как музыкальные сервисы или облачные хранилища.

Большинство людей являются любителями визуала, то есть они чаще регистрируют информацию через визуальные среды. Визуальные среды оказывают прямое и большое влияние на ум. Когда мы видим что-то привлекательное, мы автоматически заинтересованы узнать больше об этом. Этот необычный и простой человеческий инстинкт может быть использован в интересах разработчиков мобильных приложений. Разработка мобильного приложения требует тщательного изучения и понимания эстетики.

Дизайн мобильных приложений – это первое, что пользователь замечает. Это как произвести первое впечатление на кого-то. Хороший дизайн заставит пользователя пойти дальше и изучить больше, в то время как скучный дизайн может заставить пользователя отвернуться даже до того, как они узнают, что такое приложение. Да, так важно, что хороший дизайн, он может сделать или сломать пользовательский интерфейс в считанные секунды. Сделать приложение привлекательным и привлекательным – это скорее необходимое, а не легкомысленное. При разработке мобильного приложения необходимо учитывать несколько аспектов и элементов. Самое важное, что нужно учитывать, – это обеспечить четкий и четкий внешний вид. Слишком много хаоса и путаницы не только нарушают дизайн-фактор, но и приводят к ольному беспорядку. При разработке мобильного приложения необходимо учитывать следующие моменты:

– информация. Дизайн приложения должен быть в состоянии выделить необходимую информацию, а не скрывать ее. Независимо от того, что вы хотите, чтобы ваши пользователи знали сначала, нужно проецировать его в

удобное для чтения и трудно упустить. Слишком много или слишком мало информации, и ваш пользователь может быть оставлен в замешательстве;

– визуальная привлекательность. Дизайн логотипа должен сочетаться с цветовой гаммой и логотипом бренда. Цвета и шрифты имеют большое влияние на пользователя и создают визуальное впечатление от вашего бренда. Соответствующие снимки, не слишком много, также могут добавить привлекательности приложению;

– ясность. Панель навигации, строка состояния, отображение содержимого и панель инструментов должны быть представлены четким образом, не заслоняя друг друга. Чем яснее информация, тем лучше эффект у пользователя.

Общие ошибки, которых следует избегать:

– создание беспорядка. Приложение, которое выглядит и загроможденным, никогда не принесет пользы. Ясное и четкое приложение всегда будет привлекать больше внимания и оказывать большее влияние. Следует избегать хаоса в интерфейсе;

– конкретная аудитория. У каждого бизнеса есть определенная целевая аудитория, а не понимание этой аудитории приведет к падению. Следует понимать точные потребности своей аудитории, а затем разрабатывать приложение соответственно;

– быстрые решения и обновления. Приложение, которое работает нестабильно, является ошибкой. Нужно убедиться, что приложение работает без каких-либо задержек или сбоев. Желательно регулярно обновлять приложение, чтобы поддерживать его в соответствии с текущими тенденциями и требованиями;

– гарантия использования. Ваше приложение не будет успешным только потому, что оно было загружено пользователем. Только регулярное использование приложения, которое делает его успешным. Нужно убедиться, что вы даете своим пользователям повод для возврата в приложение. Опыт в создании приложения и предоставленная вами информация помогут вам в этом.

В 2013 году Google запустил Android Studio, которая была официальной интегрированной средой разработки (IDE), которая может упростить разработку приложений для мобильной платформы с лучшими визуальными функциями. С тех пор разработчики приложений начали разрабатывать приложения для Play Store, которые теперь выросли почти до 1,6 миллиона приложений. С новой структурой в Android Studio 2.0 компания Google представила множество функций производительности и повышенную скорость разработки [6].

Функция Instant Run является одним из основных дополнений к списку, так как компания способствовала разработке превью за последний год с большей утонченностью, каждый месяц. После изменения кода разработчики могут наблюдать, как изменения влияют на запущенное приложение. Теперь им не нужно повторно компилировать или переустанавливать APK, когда

должны быть сделаны изменения. Эта функция, безусловно, ускоряет редактирование кода, общих циклов сборки и запуска.

Как только приложение будет готово, Лаборатория облачных тестов Google придет на помощь для тестирования против портфеля физических устройств. Он работает с собственными тестами, поставляемыми Google, которые могут покончить с конфликтами и другими ошибками. Разработчики игр могут просмотреть графический редактор, который позволяет идентифицировать и отлаживать проблемы, связанные с графикой.

Эмулятор Android-устройств использует многоядерные процессоры, что делает весь процесс в три раза быстрее, чем раньше. Официальный Android-эмулятор довольно быстр, чем настоящий смартфон, а новые функции могут работать лучше всего с новым эмулятором. Также новый интерфейс помогает в тонкой настройке приложения, позволяя поворачивать экран, устанавливать приложения с помощью простого процесса перетаскивания с потрясающим сенсорным управлением, которое включает в себя щепок для увеличения (рисунок 1.12).

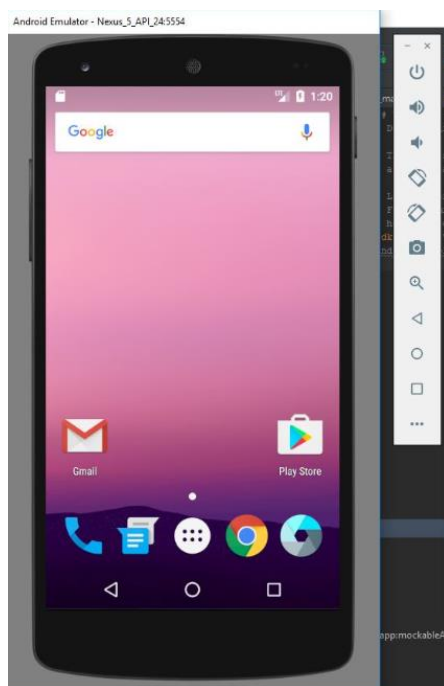


Рисунок 1.12 – Эмулятор Nexus 5

Google также внедрил индексирование приложений, которое поможет пользователям искать свое приложение в Google и Play Store. Нужно создать правильную структуру URL для приложения, с соответствующими атрибутами [7].

Все разработчики Android могут легко запускать Android Studio и Android Emulator. Для других пользователей можно пройти через некоторые учебные пособия в Интернете, в которых упоминается процедура установки и обучения.

Полный список функций Android Studio включает в себя:

– мгновенный запуск: эта функция должна улучшить рабочие процессы, позволяя просматривать изменения прямо на устройстве или эмуляторе. Все изменения можно увидеть в режиме реального времени, что означает, что вы можете постоянно писать и запускать приложение, ускоряя цикл редактирования, сборки и запуска. Когда вы нажимаете кнопку «Мгновенный запуск», изменения анализируются и определяются, как их можно развернуть самым быстрым способом. Instant Run работает со всеми типами устройств Android или эмуляторов, которые способны работать с API 14 (Ice Cream Sandwich) или выше;

– Android Emulator: новый Android-эмулятор почти в три раза быстрее в CPU, RAM и I / O по сравнению с предыдущей версией. Токовые скорости АБР почти в десять раз быстрее. Фактически, разработка на эмуляторе происходит быстрее, чем реальные операции с устройством. Новый пользовательский интерфейс и сенсорные элементы управления достаточно велики, чтобы APK упали непосредственно для быстрой установки. Легко изменять размер и масштабировать окно, использовать действия с несколькими касаниями и делать гораздо больше;

– лаборатория облачных тестов: эта новая служба позволяет разработчикам тестировать приложение на нескольких устройствах и конфигурациях устройств. Это также расширение общего процесса тестирования, включающего множество тестов против нескольких физических устройств, размещенных в центрах обработки данных самого Google. Лаборатория облачных тестов может выполнять все типы краш-тестов прямо из самой Android Studio;

– индексирование приложений. Пользователи могут легко найти свое приложение в Google Search с помощью API индексирования приложений. Android Studio может помочь получить лучшую структуру URL-адресов в коде приложения, а также поможет разработчикам добавлять атрибуты в файл AndroidManifest.xml, который хорошо дополняет службу индексирования приложений Google. Проверьте и подтвердите код позже, чтобы получить наилучшие результаты.

– предварительный просмотр отладки GPU: разработка игр OpenGL ES, особенно тех, которые особенно интенсивны с графикой, новый отладчик GPU может помочь разработчикам провести кадровый анализ, чтобы идентифицировать и отлаживать проблемы рендеринга графики со всеми видами информации о состоянии GL;

Также в Android Studio есть встроенная система автоматической сборки проектов. Называется данный инструмент Gradle.

Почти все проекты Gradle (включая все Android-проекты на базе Gradle) используют некоторые конфигурации. Что означает конфигурация в Gradle? Пример (рисунок 1.13):

```

1 dependencies {
2   annotationProcessor 'com.jakewharton:butterknife-compiler:8.4.0'
3   compile 'com.jakewharton:butterknife:8.4.0'
4   compile project(':api')
5   debugCompile 'com.squareup.leakcanary:leakcanary-android:1.4'
6   releaseCompile 'com.squareup.leakcanary:leakcanary-android-no-op:1.4'
7   androidTestCompile 'com.android.support.test:runner:0.5'
8   testCompile 'org.robolectric:robolectric:3.1.2'
9   testAnnotationProcessor 'org.robolectric:robolectric-processor:3.1.2'
10 }

```

Рисунок 1.13 – Конфигурация gradle

Конфигурации используются здесь для группировки зависимостей, они определяются с использованием шаблона `configurationName dependencyNotation`. В приведенном выше примере компиляция `testCompile` и т. д. – все имена конфигурации и «`com.jakewharton: butterknife: 8.4.0`», `project (: api)` – это обозначения зависимостей.

После оценки проекта каждая зависимость разрешается (с помощью репозиториев строфы и других правил) для файла URI, а конфигурация становится `FileCollection`. Последний может быть использован различными задачами Gradle для достижения своих целей. Например, элементы конфигурации `debugCompile` добавляются для отладки варианта компиляции `classpath`.

Имена конфигурации, добавленные плагином Android Gradle, состоят из двух основных частей:

- необязательный префикс, обозначающий вариант сборки, или тип сборки;
- требуемый суффикс, обозначающий область действия.

Например в `debugCompile: debug` – это тип сборки и компиляции. Если для всех вариантов сборки нет конфигурации префикса, например компиляция применима как к типам отладки, так и к выпуску. Каждая регулярная конфигурация также имеет соответствующий блок-тест, поэтому по умолчанию для компиляции есть также `testCompile`, `testDebugCompile` и `testReleaseCompile`. Конфигурация модульного теста наследуется от обычных.

Наконец, существует группа конфигурации `androidTest`, используемая для контрольных тестов (работает на устройстве / эмуляторе). С точки зрения конфигурации `androidTest` рассматривается как тип сборки. Обратите внимание, что в настоящее время (вплоть до Gradle plugin 2.2.0) может быть проверен только один тип сборки, поэтому нет соответствующих конфигураций отладки и выпуска. Существуют также конфигурации `wearApp`, связанные с Android Wear.

Область применения связана с фазой при использовании конфигурации (рисунок 1.14). Существует четыре фазы:

- `annotationProcassor/kapt`;
- `proccided/compileOnly`;
- `compile`;
- `apk`.

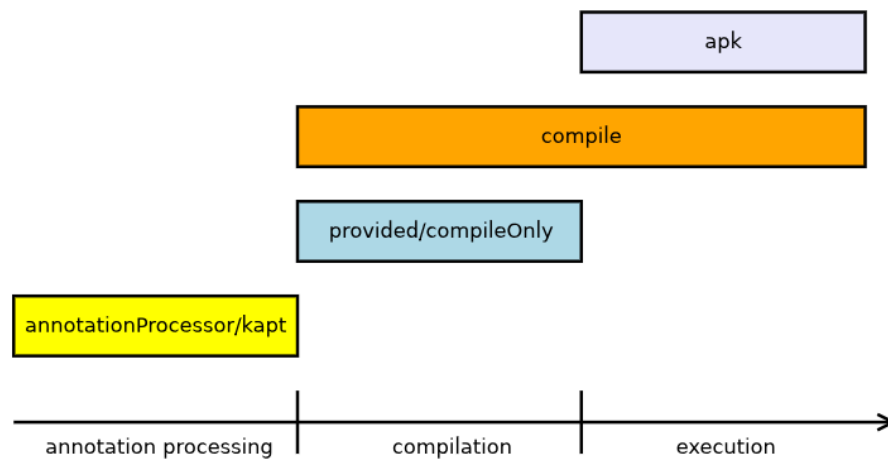


Рисунок 1.14 – Этапы сборки проекта

AnnotationProcessor / kapt. Эта область предназначена для обработчиков аннотаций, таких как dagger-compiler из Dagger 2. Использование этой конфигурации строго не требуется для обработки самой аннотации (при условии или даже компиляции в некоторых случаях также выполняет задание). Однако дополнительный объем гарантирует, что переходные зависимости (например, guava или javapoet, обычно используемые обработчиками аннотаций) не будут доступны для использования из кода приложения. Если они используются, то они доступны и используются ошибкой, что приведет к сбою во время выполнения из-за отсутствия классов. В случае компиляции все классы из транзитивных зависимостей будут доступны, но номер метода (который ограничен 65К для файла dex) часто будет резко возрастать [9].

AnnotationProcessor был представлен в Android Gradle Plugin 2.2.0. Раньше эта функциональность предоставлялась плагином Gradle для Android. В котлине эта область называется kapt.

Provided/compileOnly. В maven и Android Gradle Plugin он вызывается при условии compileOnly, так как плагины kotlin и Java Gradle называют эту область. Поскольку второе имя предполагает, что зависимости, объявленные в этой конфигурации, доступны только во время компиляции, но не во время выполнения. Они не упакованы в APK или AAR, поэтому попытка доступа к классам из этих зависимостей вызовет ошибку времени выполнения. Существует два важных ограничения этой области:

- зависимости не являются транзитивными, поэтому они не включены в зависимые проекты;
- зависимостями могут быть только JAR, а не AAR, что означает, что они не могут включать ресурсы, активы, манифестации и т. д.

Когда его следует использовать? Обычно здесь используются зависимости, содержащие аннотации только для источника. Некоторыми широко используемыми примерами проектов приложений для Android являются: Android Support Annotations (однако большое количество образцов

использует область компиляции для него) или `lombok` (также должно быть в `annotationProcessor`).

`Compile`. Это наиболее распространенный вид. Зависимости будут доступны как во время компиляции, так и во время выполнения, что желательно в большинстве случаев. Например, `butterknife` необходим во время компиляции, а также во время выполнения.

`Ark`. Это редко используется в масштабах, даже не упомянутых в официальной документации. Зависимости доступны во время выполнения, но не во время компиляции, это может быть полезно, например при использовании аннотации `@SneakyThrows` из `lombok`.

Конфигурация может расширять другую, что означает, что она содержит также элементы от родителей. Например, `testCompile` расширяет компиляцию, поэтому все производственные зависимости также могут использоваться в модульных тестах, но не наоборот. Робозлектрические классы могут использоваться в единичных тестовых источниках, но не в производственном коде. Аналогично `debugCompile` расширяет отладку и так далее. Конкретная конфигурация может наследоваться от нескольких родителей. Обратите внимание, что наследование должно быть явно объявлено, а схема именования не подразумевает его, например, `testAnnotationProcessor` не расширяет аннотацию `Processor`.

Существующие конфигурации также могут быть расширены вручную, что может быть использовано для достижения полезных результатов. Предположим, что `junit` и `assertj-android` необходимы в тестах `unit` (`src / test`) и инструментария (`src / androidTest`). Самый простой способ добиться этого - объявить каждую зависимость отдельно в соответствующей конфигурации (рисунок 1.15).

```
1 testCompile 'com.squareup.assertj:assertj-android:1.1.1'
2 testCompile 'junit:junit:4.12'
3 androidTestCompile 'com.squareup.assertj:assertj-android:1.1.1'
4 androidTestCompile 'junit:junit:4.12'
```

Рисунок 1.15 – Объявление зависимостей

Видно, есть повторения. Чтобы избежать их, можно ввести дополнительную конфигурацию, назовем ее `commonTestCompile`. Сначала нам нужно создать его в строках конфигурации. Наряду с созданием мы можем настроить наследование, сообщая, что соответствующие конфигурации из `Android` плагина простираются от вновь созданного (рисунок 1.16).

```
1 configurations {
2   [androidTestCompile, testCompile].each { it.extendsFrom commonTestCompile }
3 }
```

Рисунок 1.16 – Введение дополнительной конфигурации

Теперь все зависимости, добавленные к `commonTestCompile`, также будут доступны в `androidTestCompile` и `testCompile`, поэтому каждая общая зависимость может быть объявлена только один раз (рисунок 1.17).

```
1 commonTestCompile 'com.squareup.assertj:assertj-android:1.1.1'
2 commonTestCompile 'junit:junit:4.12'
```

Рисунок 1.17 – `AndroidTestCompile`

2 Проектная часть

2.1 Подготовка устройства для отладки

Перед началом разработки приложения следует подготовить все необходимые материалы для запуска проекта. Для начала нужно найти телефон для отладки приложения. Я решил пожертвовать своим устройством Samsung Galaxy Note 5.

Для корректной отладки приложения необходимо сперва включить режим разработчика на Android устройстве. Функцию может включить не каждый, но для ее запуска не требуется никаких специальных навыков.

Итак, заходим в «Настройки» далее листаем вниз и тапаем по пункту «Об устройстве». Открывается окно информации об устройстве (рисунок. 2.1)

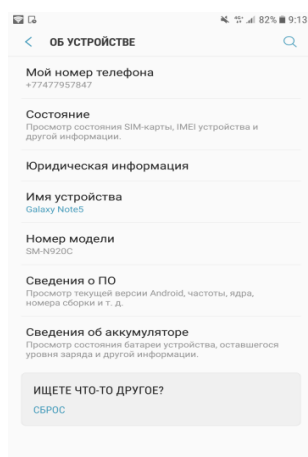


Рисунок 2.1 – настройки

Далее выбираем «Сведения о ПО» и в следующем окне находим пункт «Номер сборки» (рисунок 2.2)

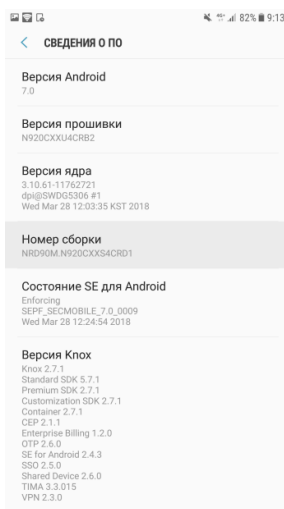


Рисунок 2.2 – Сведения о ПО

Быстро нажимаем на него 10 раз и у нас появляется окно, что теперь нам доступен режим разработчика. Все теперь вы разработчик! Что такое режим разработчика? Это инженерное меню позволяющее более гибко настроить ваше Android устройство. В нем есть следующие функции:

- создать отчет об ошибках;
- пароль резервного копирования;
- есть возможность сохранения всех пакетов Bluetooth HCI в файл;
- включение демонстрационного режима;
- возможность отключения автообновления системы;
- также есть возможность симуляции местоположения и т. д.

Нам нужен единственный пункт: Отладка USB(рисунок 2.3). Переводим этот параметр в активный режим, посредством нажатия. Предназначений у данного параметра масса. В основном это операции проводимые через компьютер. Нам же данный параметр нужен для постоянной отладки приложения и обновления приложения на телефоне.

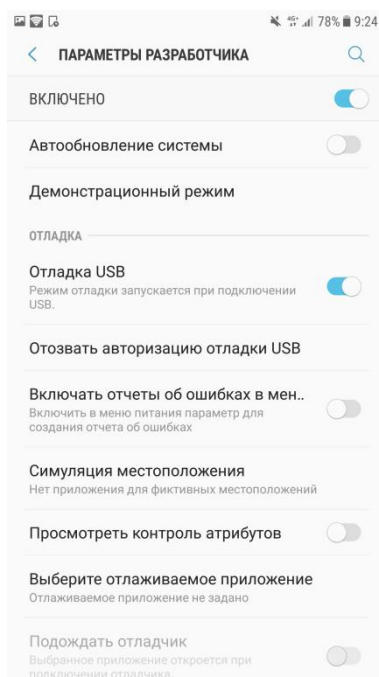


Рисунок 2.3 – Включение отладки по USB

2.2 Установка Delphi

Писать приложение я буду в среде программирования Delphi 10.1 Berlin.

Перед установкой основного софта, нам следует установить Java Development Kit(JDK) Скачиваем его с официального сайта oracle.com. На сайте принимаем лицензию, скачиваем установщик под свою систему (у меня это Windows 10 x64) запускаем и он все делает за нас. После этого нам следует прописать новую переменную в переменных средах. Для этого открываем свойства компьютера, далее выбираем пункт «Дополнительные

параметры системы», после чего жмем кнопку переменные среды. Появляется окно, нажимаем кнопку создать и далее как на скриншоте (рисунок 2.4).

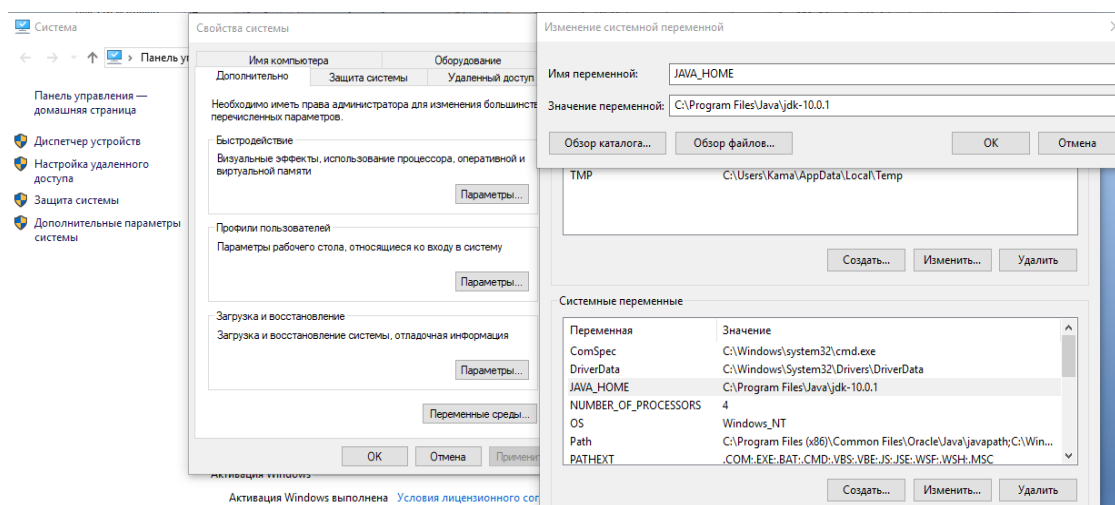


Рисунок 2.4 – Создание системной переменной

Теперь мы можем устанавливать нашу среду разработки. Скачиваем и устанавливаем Delphi 10.1 Berlin со встроенным GUI-фреймворком.

После установки программа занимает примерно 18Гб пространства на жестком диске

Как только закончится установка можем запускать утилиту и начинать создание нового проекта (рисунок 2.5)

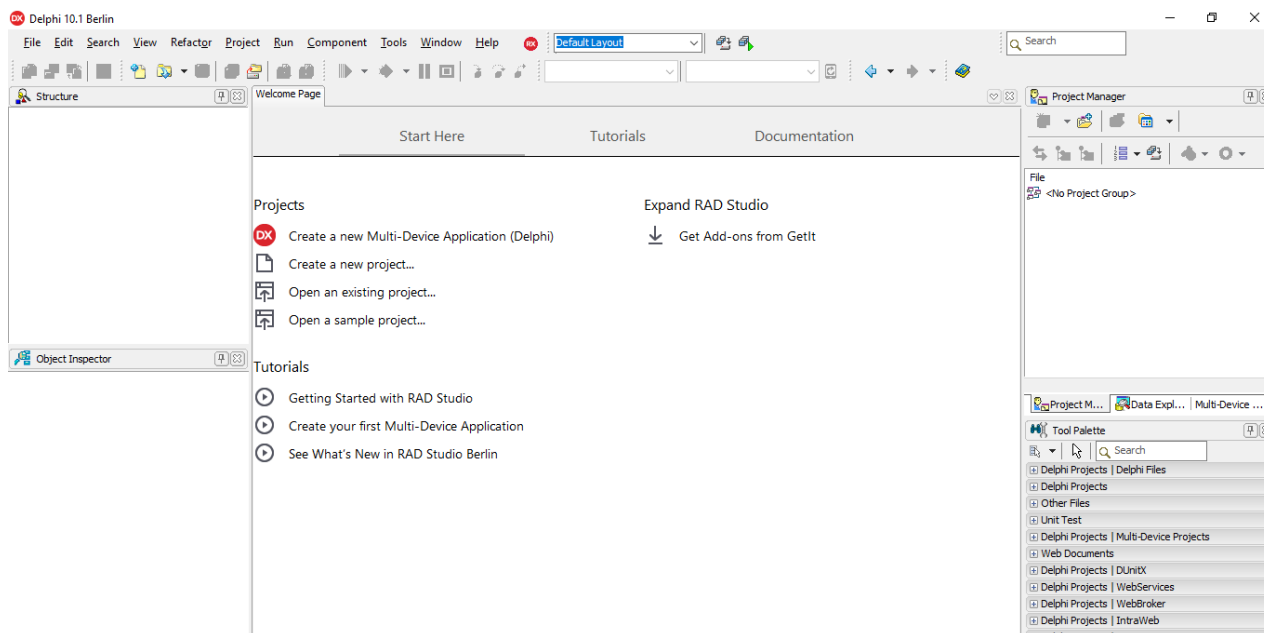


Рисунок 2.5 – Delphi 10.1

Delphi 10.1 Berlin – это среда разработки основным языком которой является всем знакомый Delphi. Особенностью данной среды является

встроенный фреймворк FireMonkey позволяющий разрабатывать мобильные приложения.

FireMonkey был разработан российским программистом Евгением Крюковым. Первым названием фреймворка было VG Scene. Позднее проект был выкуплен компанией Embadarcadero Technologies.

FireMonkey является кроссплатформенным и поддерживается на всех основных операционных системах.

2.3 База данных для Android, SQLite

Почти все приложения используют базы данных для хранения информации. Во время создания приложений перед многими разработчиками рано или поздно встает вопрос: какую базу использовать для своего приложения? В настоящий момент рынок перенасыщен различного рода приложениями. Гигантская часть этих приложений довольно просты в написании и не требуют больших интеллектуальных и материальных затрат на их разработку. В свою очередь немалая часть этих приложений работает используя БД SQLite.

На самом деле в этом нет ничего удивительного ведь она очень проста в использовании. Все действия выполняются на языке SQL. Каждый кто имел дело с разработкой сталкивался с необходимостью изучения этого языка. Главной отличительной чертой является отсутствие облачного сервера. Сам смартфон выступает в качестве сервера, т. е. все данные хранятся на смартфоне, что обеспечивает автономность в определенном плане.

Существует как минимум два способа использования SQLite при проектировании на Delphi:

- создание базы с помощью стороннего софта;
- создание базы с помощью консоли.

Рассмотрим первый вариант. Для того чтобы сделать базу SQLite, мы можем использовать предназначенные для этого программы. Мой выбор пал на DB Browser For SQLite. Находим в интернете, скачиваем, устанавливаем, открываем.

Нам сразу представляется окно в котором можно создать или импортировать. Если есть готовая БД, то чтобы редактировать ее наводим курсор на пункт меню «Файл», нажимаем и из выпадающего списка выбираем «Открыть базу данных». Также можно открыть просто базу SQL. Для просмотра мы можем использовать функцию «Open Database Read Only». В таком случае мы можем открыть файл только для чтения, без возможности редактирования.

Чтобы начать процесс создания, нажимаем на кнопку «Новая база данных». Программа предлагает выбрать путь для сохранения. Выбираем рабочий стол (для удобства) (рисунок 2.6).

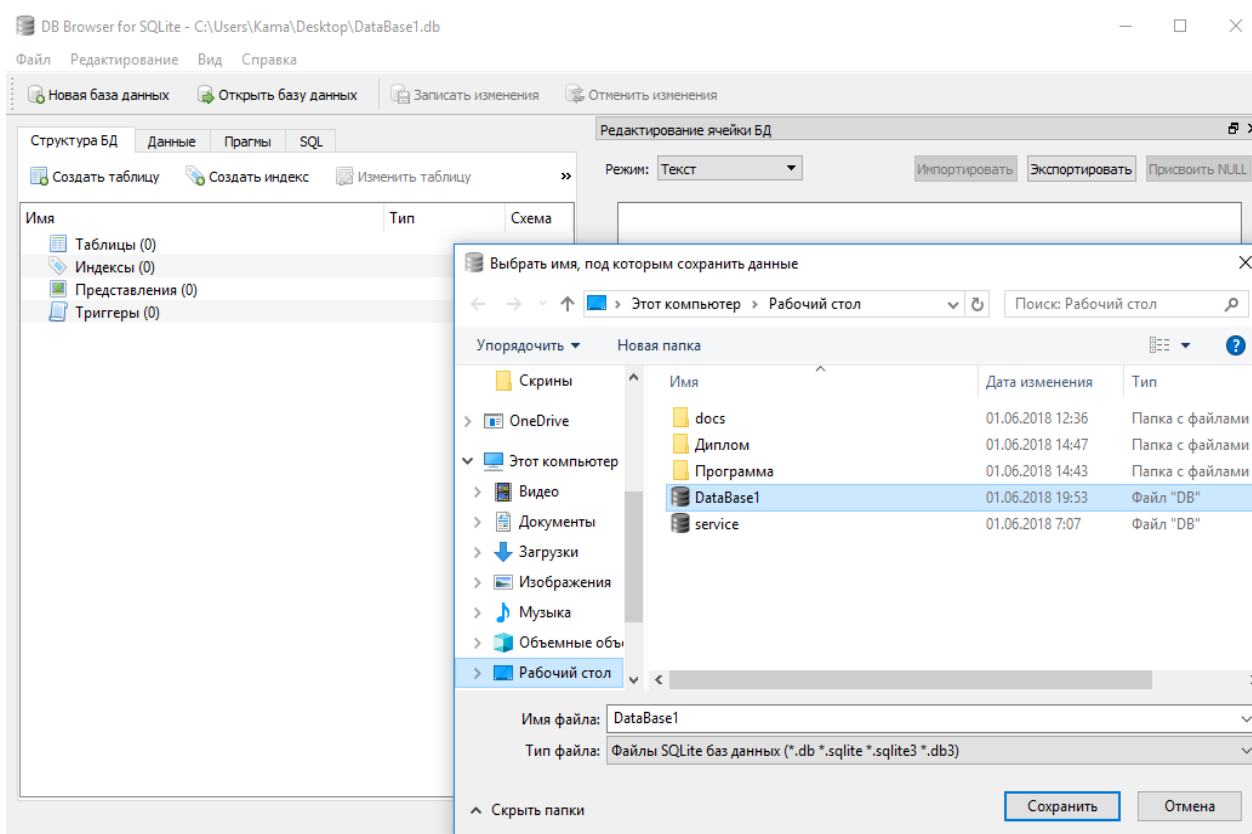


Рисунок 2.6 – Создание БД

После сохранения создаем необходимые таблицы и начинаем их заполнять. Следует сразу выбирать параметры. SQLite способен работать с целочисленным integer, пустым NULL, плавающим real, text, numeric и blob. Нет поддержки Boolean, вместо этого используется 0 для false и 1 для true.

Я создал две таблицы clients и parts. Первая для учета клиентов, вторая для учета запчастей т.к. приложение создается для сервис центра.

После создания таблицы начинаем создавать столбцы, для этого нажимаем кнопку создать поле и вводим названия столбцов. После этого из выпадающего меню выбираем тип данных которые будут храниться в столбце. Также можно выбрать некоторые характеристики. Значение не должно быть пустым, включить автоинкремент (для _id), сделать строку уникальной или установить первичный ключ(рисунок 2.7).

С помощью этой программы мы также можем выстраивать связи, но нам пока это не нужно.

В отличие от MySQL, интерфейс программы намного проще, что упрощает работу с ней, в ней есть только те функции которые требуются для создания полностью функционирующей базы предназначение которой интеграция с Android.

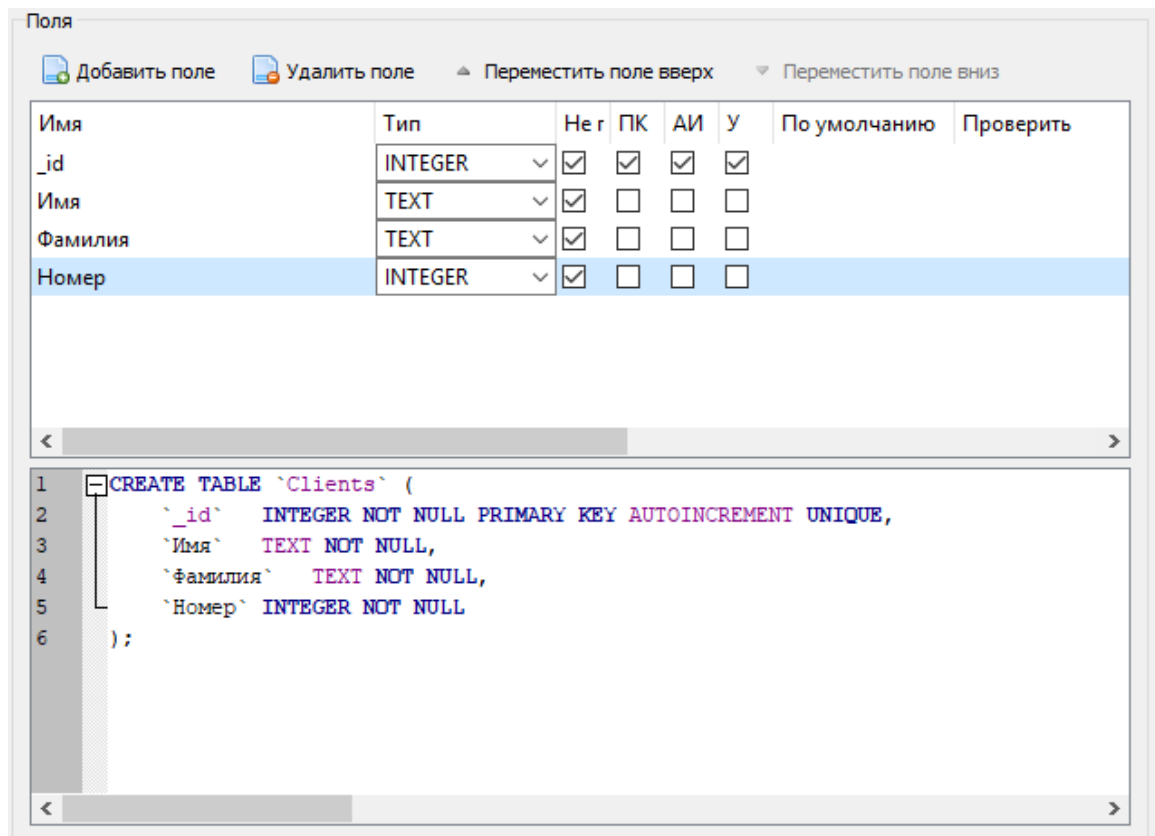


Рисунок 2.7 – Создание строк

Далее следует заполнить её некоторыми данными, для дальнейшего вывода в интерфейс приложения. Я ввел несколько клиентов, а также добавил пару запчастей.

Чтобы использовать эту базу во время разработки нам надо просто закинуть ее в папку к проекту. Затем при помощи `FDCConnection` мы указываем путь к нашей БД.

Второй вариант требует меньше времени, но важно наличие навыков и опыта в данной сфере. Осуществляется данный метод через консоль (рисунок 2.8).

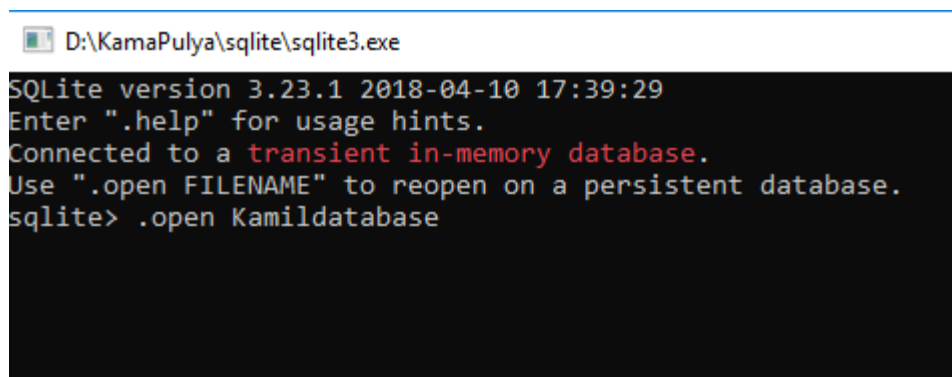


Рисунок 2.8 – Создание БД Kamildatabase

Так выглядит код для создания БД и добавления таблицы «zapchasti» (рисунок 2.9):

```
open.base

CREATE TABLE zapchasti(
id INTEGER PRIMARY KEY AUTOINCREMENT,
model TEXT,
zapchast TEXT,
cvet TEXT,
cena TEXT);
```

Рисунок 2.9 – Создание БД и добавление таблицы «zapchasti»

Далее надо создать вторую таблицу куда будут добавляться клиенты (рисунок 2.10):

```
CREATE TABLE client(
id INTEGER PRIMARY KEY AUTOINCREMENT,
imei TEXT,
name TEXT,
number TEXT,
email TEXT,
model TEXT);
```

Рисунок 2.10 – Добавление таблицы «client»

2.3 Разработка приложения

Прежде чем приступить к написанию основного кода, я начал с создания интерфейса. Интерфейс создается с помощью встроенного конструктора (рисунок 2.11)

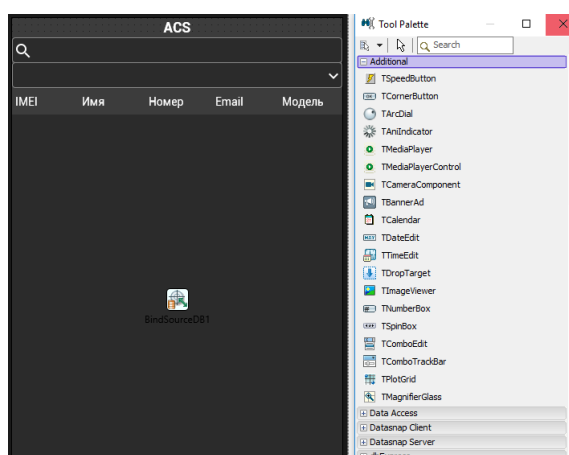


Рисунок 2.11 – Создание интерфейса

Присваиваем значения всем элементам (рисунок 2.12):

```
- type
20 TForm1 = class(TForm)
-     StyleBook1: TStyleBook;
-     ComboBox1: TComboBox;
-     FDConnection1: TFDConnection;
-     FDQuery1: TFDQuery;
-     Label1: TLabel;
-     StringGrid1: TStringGrid;
-     Edit1: TEdit;
-     BindSourceDB2: TBindSourceDB;
-     BindingsList1: TBindingsList;
30     Timer1: TTimer;
-     Button2: TButton;
-     Button3: TButton;
-     FDQuery2: TFDQuery;
-     BindSourceDB1: TBindSourceDB;
-     LinkGridToDataSourceBindSourceDB2: TLinkGridToDataSource;
-     StringGrid2: TStringGrid;
-     LinkGridToDataSourceBindSourceDB1: TLinkGridToDataSource;
-     Button1: TButton;
```

Рисунок 2.12 – Присваивание значений

Создаем события для элементов (рисунок 2.13):

```
- procedure FormKeyUp(Sender: TObject; var Key: Word; var KeyChar: Char;
40     Shift: TShiftState);
- procedure Timer1Timer(Sender: TObject);
- procedure StringGrid1Tap(Sender: TObject; const Point: TPointF);
- procedure FDConnection1BeforeConnect(Sender: TObject);
- procedure Edit1KeyUp(Sender: TObject; var Key: Word; var KeyChar: Char;
-     Shift: TShiftState);
- procedure ComboBox1Change(Sender: TObject);
- procedure Button2Click(Sender: TObject);
- procedure Button3Click(Sender: TObject);
- procedure Button1Click(Sender: TObject);
```

Рисунок 2.13 – Создание событий

На главном экране расположены три кнопки (рисунок 2.14). Кнопка «Удалить запись» отвечает за удаление выбранной записи из базы данных. Кнопка «Добавить запчасть» отвечает за переход в форму с добавлением запчасти. При нажатии на кнопку «Добавить клиента» происходит переход в форму добавления клиента (Рисунок 2.14).

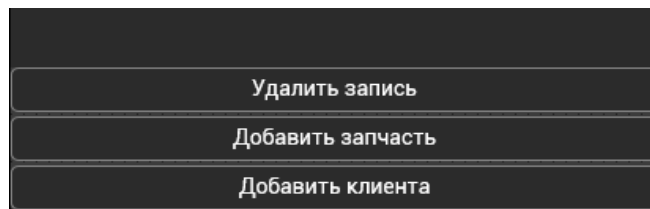


Рисунок 2.14 – Кнопки

Если мы нажимаем на кнопку «Добавить запчасть» то мы откроем форму добавления запчасти (рисунок 2.15):

The image shows a vertical form with a dark background. It contains four input fields, each with a label and a dropdown arrow on the right: 'Модель' (Model), 'Запчасть' (Part), 'Цвет' (Color), and 'Цена' (Price). At the bottom of the form is a prominent red button with the text 'Добавить' (Add).

Рисунок 2.15 – Форма добавления запчасти.

По нажатии на поле «Модель» выскакивает окошко с вариантами выбора, в зависимости от выбора, меняются данные которые заносятся в таблицу. Аналогичный процесс с полями «Запчасть» и «Цвет». В поле цена осуществляется ввод с клавиатуры.

После ввода всех характеристик нажимаем кнопку «Добавить». При нажатии происходит добавление внесенных изменений в таблицу «zapchast» и закрытие формы. Описание события выглядит следующим образом (рисунок 2.16):


```

procedure TForm2.Button1Click(Sender: TObject);
begin
50  with Form1.FDQuery1 do
begin
Form1.FDConnection1.Connected:=True;
SQL.Text:='INSERT INTO zapchasti (model, zapchast, cvet, cena) VALUES (:model, :zapchast,
ParamByName('model').Value:=ComboBox1.Items.Strings[ComboBox1.ItemIndex];
ParamByName('zapchast').Value:=ComboBox2.Items.Strings[ComboBox2.ItemIndex];
ParamByName('cvet').Value:=ComboBox3.Items.Strings[ComboBox2.ItemIndex];
ParamByName('cena').Value:=Edit1.Text;
OpenOrExecute;
ComboBox1.ItemIndex:=-1;
60  ComboBox2.ItemIndex:=-1;
ComboBox3.ItemIndex:=-1;
Edit1.Text:='';
ShowMessage('Запись добавлена');
Form1.Visible:=True;
Form1.ComboBox1.ItemIndex:=0;
Form1.FDConnection1.Connected:=True;
SQL.Text:='SELECT * FROM zapchasti';
OpenOrExecute;
Form2.Visible:=False;
70  end;
end;

```

Рисунок 2.16 – Описание кнопки «Добавить»

Если в основном меню нажать кнопку «Добавить клиента», то откроется третья форма по добавлению клиента в базу, выглядит он также как и форма добавления запчастей, но с другими характеристиками (рисунок 2.17):

Рисунок 2.17 – Форма добавления клиента

Описание кнопки «Добавить» аналогично , за исключением таблицы добавления (рисунок 2.18):

```

begin
with Form1.FDQuery2 do
begin
Form1.FDConnection1.Connected:=True;
SQL.Text:='INSERT INTO client (imei, name, number, email, model) VALUES (:imei, :name, :numb
ParamByName('imei').Value:= Edit1.Text;
ParamByName('name').Value:= Edit2.Text;
ParamByName('number').Value:= Edit3.Text;
ParamByName('email').Value:= Edit4.Text;
ParamByName('model').Value:=ComboBox1.Items.Strings[ComboBox1.ItemIndex];
OpenOrExecute;
Edit1.Text:='';
Edit2.Text:='';
Edit3.Text:='';
Edit4.Text:='';
ComboBox1.ItemIndex:=-1;
ShowMessage('Запись добавлена');
Form1.Visible:=True;
Form1.ComboBox1.ItemIndex:=1;
Form1.FDConnection1.Connected:=True;
SQL.Text:='SELECT * FROM client';
OpenOrExecute;
Form3.Visible:=False;
end;
end;

```

Рисунок 2.18 – Описание кнопки добавить

В итоге приложение работает по данной логике (рисунок 2.19):

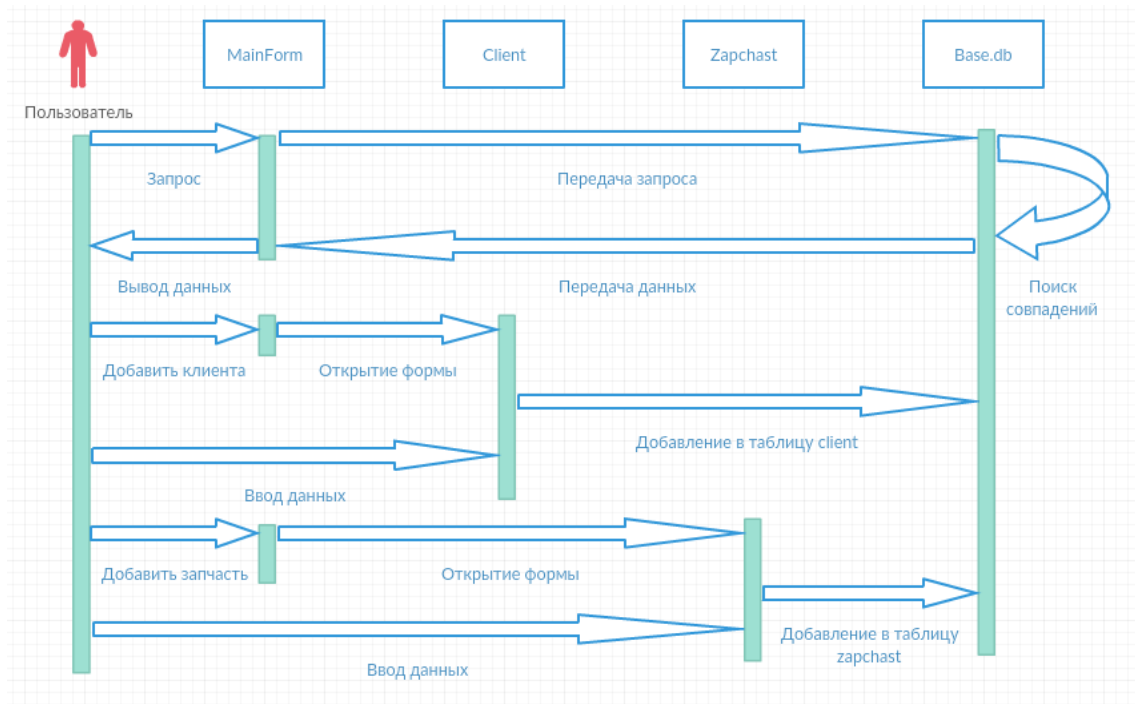


Рисунок 2.19 – Логика работы приложения

3 Экспериментальная часть

Тестирование приложения я проводил на своем телефоне Samsung Galaxy Note 5. Подсоединяем свое устройство к компьютеру, ждем когда программа определит наше устройство (рисунок 3.1).

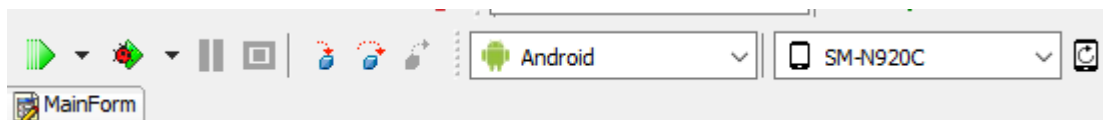


Рисунок 3.1 – Список устройств

Далее нажимаем на кнопку запуска, после нажатия появляется окно компиляции, оно сообщает нам о проверенных строках и количестве ошибок (рисунок 3.2).

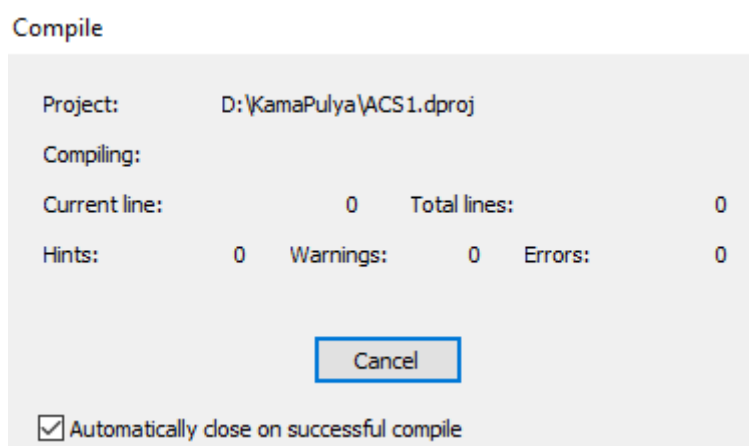


Рисунок 3.2 – Запуск отладки.

После нажатия программа начинает компиляцию нашего кода. После успешной сборки, на смартфон устанавливается обновленный APK-файл и приложение сразу само запускается на телефоне. При запуске мы сразу видим главное меню. При нажатии на поле, под областью ввода текста для поиска, выходит окошко по выбору таблицы которую необходимо вывести для осмотра и редактирования (рисунок 3.3). Данные выводятся на середину формы в виде таблицы со строками и столбцами (рисунок 3.4).

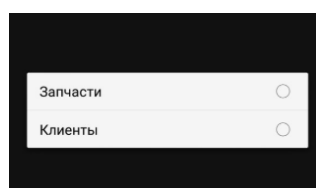


Рисунок 3.3 – Выбор отображаемой таблицы

Модель	Запчасть	Цвет	Цена
iPhone	Дисплей	Черный	30000
iPhone	Дисплей	Черный	12000
Samsung	АКБ	Белый	15000
Samsung	АКБ	Белый	15000
iPhone	Дисплей	Черный	28560

Удалить запись

Добавить запчасть

Добавить клиента

Рисунок 3.4 – Отображение таблицы

После того как приложение запущено, мы можем выбрать один из трех пунктов в зависимости от нужной нам цели. Если мы выберем пункт «Добавить клиента» то откроется следующее окно, которое предложит нам ввести данные нашего клиента (рисунок 3.5).

IMEI

Имя

Номер

Почта

Модель

Добавить

Рисунок 3.5 – Добавление клиента

В этой форме мы вводим имя клиента и его характеристики. IMEI телефона, который он делал в нашем сервис-центре. Его имя и фамилию. Номер сотового телефона, а также почту. Ввод происходит с клавиатуры телефона. По нажатии кнопки, данные добавляются в таблицу CLIENT. Данные каждого из полей записываются в соответствующие столбцы нашей таблицы.

В случае, когда мы нажимаем на кнопку «Добавить запчасть» открывается аналогичное окно, но с другими полями ввода (рисунок 3.6).

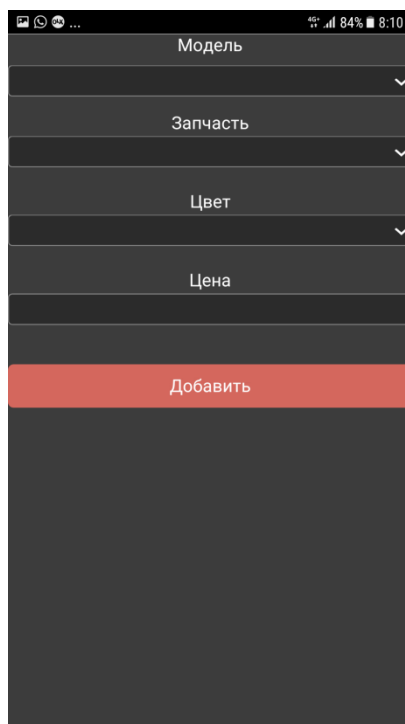


Рисунок 3.6 – Добавление запчасти

Форма аналогична первому окну по интерфейсу. Но отличие от предыдущего здесь абсолютно другие характеристики. Здесь мы вводим модель телефона для описания запчасти, цвет, выбираем тип запчасти (дисплей, корпус, шлейф, камера и т. д.) и конечно ставим цену. По нажатии кнопки происходит добавление данных в соответствующие столбцы. Данные записываются в таблицу `zarchast`.

4 Техничко-экономическое обоснование проекта

Техничко-экономическое обоснование проекта проводится для вычисления и анализирования затрат денежных и трудовых, использованных для воплощения дипломной работы.

Целевой аудиторией разрабатываемого ПП является малый бизнес. Не каждый предприниматель способен позволить себе дорогостоящее ПО имеющее широкий функционал, в котором предприниматель не нуждается. Мое приложение выполняет две главные функции незаменимые при ведении малого бизнеса, это:

- ведение клиентской базы;
- наличие складского учета;

В этой части представляется рассмотрение экономической составляющей по разработке мобильного приложения для платформы Android.

4.1 Трудоемкость разработки проекта

Чтобы определить трудоемкость разработки проекта нужно сперва составить список всех основных этапов и видов работ, которые должны быть выполнены. При этом особое внимание должно быть уделено логическому упорядочению последовательности отдельных видов работ и выявлению возможностей их параллельного исполнения, что позволяет существенно сократить общую длительность проведения разработки проекта.

Нужно выделить такие этапы как: постановка задачи, разработка алгоритма, составление программы по готовому алгоритму, отладка программы, подготовка соответствующей документации.

Определение трудоемкости (затрат времени) на разработку проекта такое же трудное занятие как и нормирование любого творческого труда, содержащего технические (рутинные) элементы. Творческие элементы труда программистов практически не нормируются, они могут быть определены либо на основе экспертных оценок опытных программистов, либо жестко заданными сроками разработки, в которые программист обязан найти решение [1].

Форма разделения работ по этапам с указанием трудоемкости их выполнения приведена в таблице 4.1.

Таблица 4.1 – Распределение работ по этапам и видам и оценка их трудоемкости

Этапы разработки проекта	Вид работы на данном этапе	Трудоемкость разработки ПП, чел.× ч.
1	Постановка задачи	5

Продолжение таблицы 4.1

2	Разработка алгоритма, функциональной структуры	40
3	Выбор комплекса программных и технических средств для реализации проекта	5
4	Реализация программы по готовой схеме, функциональной структуре	120
5	Отладка программы на ЭВМ	30
6	Подготовка соответствующей документации	50
7	Подведение итогов	70
ИТОГО трудоемкость выполнения проекта		320

4.2 Расчет затрат на разработку

При определении затрат на разработку проекта используется следующее:

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

4.2.1 Расчет материальных затрат

В «Материальные затраты» включаются расходы на основные и расходные материалы (бумага, картриджи и другие), энергию, необходимые для разработки проекта.

Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 4.2.

Таблица 4.2 – Затраты на материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Листы бумаги А4	упаковка	2	1 000	2 000
Картриджи	шт	1	10 000	10 000
ИТОГО затраты на материальные ресурсы				12 000

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

Таблица 4.3 – Стоимость оборудования и программного обеспечения

Наименование	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Microsoft Office 2016	ключ	1	33400	33 400
Microsoft Visual Studio 2017	ключ	1	158 600	158 600
Планшет ASUS Zenpad	шт	1	45 000	45 000
Ноутбук Lenovo G580	шт	1	80 000	80 000
Принтер HP Europe LaserJet Pro M102a	шт	1	38 800	38 800
ИТОГО затраты на оборудование и ПО				355 800

Общая сумма затрат на материальные ресурсы (З_{МР}) определяется по формуле (4.1):

$$Z_{MP} = \sum_{i=1}^n P_i * C_i, \quad (4.1)$$

где P_i – расход i -го вида материального ресурса, натуральные единицы;
 C_i – цена за единицу i -го вида материального ресурса, тг;
 i – вид материального ресурса;
 n – количество видов материальных ресурсов.

4.2.2 Расчет затрат на электроэнергию

Так. Как мы используем электрооборудование для разработки проекта, то у нас есть затраты на электроэнергию. Эти оборудованием является:

- принтер;
- ноутбук;
- планшет.

Необходимо рассчитать затраты на электроэнергию по форме, приведенной в таблице 4.4.

Таблица 4.4 – Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки, ч	Цена эл/э, тенге/ кВт-час;	Сумма, тенге
Принтер HP Europe LaserJet Pro M102a	1,5	0,9	50	27,48	1855
Планшет ASUS Zenpad	0,03	0,9	30	27,48	22
Ноутбук G580	0,065	0,9	320	27,48	514
ИТОГО затраты на электроэнергию					2391

Общая сумма затрат на электроэнергию (Z_{MP}) рассчитывается по формуле (4.2):

$$Z_{MP} = \sum_{i=1}^n M_i * K_i * T_i * Ц, \quad (4.2)$$

где M_i паспортная мощность i -го электрооборудования, кВт;
 K_i – коэффициент использования мощности i -го электрооборудования (принимается $K_i=0.7, 0.9$);
 T_i – время работы i -го оборудования за весь период разработки ч;
 $Ц$ – цена электроэнергии, тг/кВт×ч;
 i – вид электрооборудования;
 n – количество электрооборудования. Выполним расчеты для оборудования:

Принтер HP Europe LaserJet Pro M102a

$$Z_{э}=1,5*0,9*50*27,48=1855 \text{ тг.}$$

– Планшет ASUS Zenpad

$$Z_{э}=0.03*0,9*30*27,48=22 \text{ тг.}$$

– Ноутбук Lenovo Y50–70

$$Z_{э}=0,065*0,9*320*27,48=514 \text{ тг.}$$

Итого общая сумма затрат на электроэнергию составит 2020 тг.

4.2.3 Расчет затрат на оплату труда

Для того чтобы рассчитать затраты по оплате труда нужно суммировать расходы по оплате труда всех работников, занятых разработкой проекта (дипломника, руководителей и консультантов дипломной работы, привлеченных лиц).

Затраты на оплату труда рассчитываются по форме, приведенной в таблице 4.5.

Таблица 4.5 – Затраты на оплату труда

Категория работника	Месячная зарплата тг	Трудоемкость разработки проекта, чел.×ч	Часовая ставка, тг/ч	Сумма, тг
Разработчик приложения	150 000	240	1013	243 120
Руководитель	200 000	80	1351	108 080
ИТОГО затраты на оплату труда				351 200

Общая сумма затрат на оплату труда (ЗТР) определяется по формуле (4.3):

$$З_{ТР} = \sum_{i=1}^n ЧС_i * T_i, \quad (4.3)$$

где ЧС_і – часовая ставка і-го работника, тг;

Т_і – трудоемкость разработки ПП, чел.×ч;

і – категория работника;

n – количество работников, занятых разработкой ПП.

Часовая ставка работника может быть рассчитана по формуле (4.4):

$$ЧС_i = \frac{ЗП_i}{ФРВ_i}, \quad (4.4)$$

где ЗП_і – месячная заработная плата і-го работника, тг;

ФРВ_і – месячный фонд рабочего времени і-го работника, ФРВ_і = 148 час.

Трудоемкость разработки проекта определяется по данным таблицы 4.1.

Таким образом, произведем расчеты для сотрудников:

$$\text{ЧС}_1 = \frac{150000}{148} = 1013 \text{ тг/ч} \quad \text{З}_{\text{ТР1}} = 1013 * 240 = 243120 \text{ тг}$$

$$\text{ЧС}_2 = \frac{200000}{148} = 1351 \text{ тг/ч} \quad \text{З}_{\text{ТР2}} = 1351 * 80 = 108080 \text{ тг}$$

4.2.4 Расчет отчислений на социальные нужды

В пункт «Социальный налог» включается сумма, которая рассчитывается как 9,5 % от затрат на оплату труда всех работников ($\text{З}_{\text{тр}}$), занятых разработкой проекта. При расчете нужно учитывать тот факт, что пенсионные отчисления (10% от $\text{З}_{\text{тр}}$) не облагаются социальным налогом (ставки указаны на 2018 год).

Отчисления по заработной плате определяются по следующей формуле (4.5):

$$\text{О}_c = (\text{З}_{\text{тр}} - \text{ПО}) * 0,095, \quad (4.5)$$

где ПО – пенсионные отчисления, составляют 10% от $\text{З}_{\text{тр}}$,

$$\text{ПО} = 351\,200 * 0,1 = 35\,120 \text{ тг.}$$

Итак, отчисления из заработной платы составили:

$$\text{О}_c = (405\,760 - 40\,576) * 0,095 = 30\,028 \text{ тг.}$$

4.2.5 Расчет амортизационных отчислений

Амортизация - метод учета, используемый для постепенного снижения стоимости конечного срока службы или нематериального актива за счет плановых платежей за доход. Амортизация - это погашение долга с фиксированным графиком погашения в виде регулярных платежей в течение определенного периода времени. Это также относится к распределению капитальных затрат на нематериальные активы в течение определенной продолжительности (обычно в течение срока полезного использования актива) для целей учета и налогообложения.

Пункт «Амортизация основных фондов» состоит из суммы амортизационных отчислений от стоимости оборудования и программного обеспечения (ПО), используемых при разработке проекта. Амортизационные отчисления рассчитываются по форме, приведенной в таблице 4.6.

Таблица 4.6 – Амортизация основных фондов (ОФ)

Наименование оборудования	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Эффективный фонд времени работы оборудования и ПО, ч/год	Время работы оборудования и ПО для разработки и ПП, ч	Время работы оборудования и ПО для разработки проекта, Ч.	Сумма, тг
Ноутбук Lenovo G580	80 000	20	5760	240	320	889
Принтер HP Europe LaserJet Pro M102a	38 800	20	5760	0	50	67
Планшет ASUS ZenPad	45 000	20	5760	30	30	46
Microsoft Office 2016	33 400	20	5760	0	120	139
Microsoft Visual Studio 2017	158 600	20	5760	240	240	1322
ИТОГО амортизация основных фондов						2463

Общая сумма амортизационных отчислений определяется по формуле (4.6):

$$Z_M = \sum_{i=1}^n \frac{\Phi_i * H_{Ai} * T_{НИРi}}{100 * T_{ЭФi}}, \quad (4.6)$$

где Φ_i – стоимость i -го ОФ, тг;
 H_{Ai} – годовая норма амортизации i -го ОФ, %;
 $T_{НИРi}$ – время работы i -го ОФ за весь период разработки ПП, дней;
 $T_{ЭФi}$ – эффективный фонд времени работы i -го ОФ за год, дней/год; i – вид ОФ;
 n – количество ОФ.

При определении стоимости ОФ необходимо учесть также затраты на доставку и монтаж, установку ПО. Эти затраты могут быть приняты в размере 10–25 % от затрат на приобретение ОФ.

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ (4.7):

$$H_{Ai} = \frac{100}{T_{Ni}}, \quad (4.7)$$

где T_{Ni} – возможный срок использования;

i –го ОФ, год;

Возможный срок полезного использования ОФ 5 лет.

Срок полезного использования показывает в течении какого периода времени, использование объекта основных средств способно приносить доход и служить для выполнения целей деятельности организации, определяемых для принятых к бухгалтерскому отчету основных средств.

В расчетах принимается максимально возможный фонд времени работы оборудования. Эффективный фонд времени работы оборудования (для предприятий вторичных ресурсов с прерывным режимом работы) определяется числом рабочих дней в году и числом часов работы оборудования в сутки по формуле (4.8):

$$T_{эф} = T_{ном} * 12 - T_{ппр}, \quad (4.8)$$

где $T_{эф}$ – эффективный фонд времени работы оборудования, дни;

$T_{ном}$ – номинальный фонд времени, дни;

$T_{ппр}$ – время остановки работы оборудования на проведение плановопредупредительных ремонтов (включая время на подготовку к ремонту и пуск оборудования после ремонта), дни.

Проведем расчет эффективного фонда времени работы оборудования:

– Ноутбук Lenovo G580

$$T_{эф1} = 21 * 12 - 12 = 240 * 24 = 5760 \text{ ч.}$$

– Принтер HP Europe LaserJet Pro M102a

$$T_{эф2} = 21 * 12 - 12 = 240 * 24 = 5760 \text{ ч.}$$

– Планшет ASUS ZenPad

$$T_{эф3} = 21 * 12 - 12 = 240 * 24 = 5760 \text{ ч.}$$

– Microsoft Office 2016

$$T_{эф3} = 21 * 12 - 12 = 240 * 24 = 5760 \text{ ч.}$$

– Microsoft Visual Studio 2017

$$T_{эфз}=21*12-12=240*24=5760 \text{ ч.}$$

Приведем расчеты амортизационных отчислений для каждого оборудования:

– Ноутбук Lenovo G580

$$З_M = \frac{80000*20*320}{100*5760} = 889 \text{ тг.}$$

– Принтер HP Europe LaserJet Pro M102a

$$З_M = \frac{38800*20*50}{100*5760} = 67 \text{ тг.}$$

– Планшет ASUS ZenPad

$$З_M = \frac{45000*20*30}{100*5760} = 46 \text{ тг.}$$

– Microsoft Office 2016

$$З_M = \frac{33400*20*120}{100*5760} = 139 \text{ тг.}$$

– Microsoft Visual Studio 2017

$$З_M = \frac{158600*20*240}{100*5760} = 1322 \text{ тг.}$$

Итого общая сумма амортизационных отчислений составляет 2463 тг.

4.2.6 Расчет прочих затрат

В пункт «Прочие затраты» записываются расходы на арендную плату, включая коммунальные платежи, затраты на лицензирование и сертификацию, расходы на рекламу, канцелярские и прочие хозяйственные расходы.

Затраты на арендную плату определяются в зависимости от стоимости аренды 1 кв. м арендованной площади и в зависимости от длительности проекта. Длительность проекта составляет 40 рабочих дней, а это 8 рабочих недель, учитывая выходные, что составляет два месяца аренды. Данные приведены в таблице 4.7.

Таблица 4.7 – Затраты на арендную плату

Площадь, кв.м.	Цена за кв.м., тг	Цена за месяц, тг	Срок, месяц	Сумма, тг
25	2000	50000	2	100000

Во время создания приложения интернет обеспечивал провайдер Beeline. Стоимость тарифа предоставлена в таблице 4.8.

Таблица 4.8 – Затраты на интернет–услуги

Ежемесячная плата, тг	Срок, месяц	Сумма, тг
3500	2	7000

Затраты на продвижение и рекламу отсутствуют т. к. приложение распространяется через Play Market.

Стоимость регистрации в Play Market 25\$ или по курсу национального банка РК на 21.05.2018, 8207 тг.

Таким образом, прочие затраты составляют 115 207 тенге.

4.2.7 Составление сметы

На основании полученных данных по отдельным пунктам составляется смета затрат на разработку проекта по форме, приведенной в таблице 4.9.

Таблица 4.9 – Смета затрат на разработку проекта

Пункты затрат	Сумма, тг
Материальные затраты, в том числе:	
– материалы	12 000
– электроэнергия	2 391
Затраты на оплату труда.	351 200
Отчисления на социальные нужды.	30 028
Амортизация основных фондов.	2463
Прочие затраты.	115 207
ИТОГО по смете	513 289

4.3 Определение возможной (договорной) цены проекта

Величина возможной (договорной) цены проекта считается с оглядкой на эффективность, качество и сроки выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена (Ц_д) для прикладных проектов рассчитывается по

формуле (4.9):

$$C_{\text{д}} = Z_{\text{НИР}} * \left(1 + \frac{P}{100}\right), \quad (4.9)$$

где $Z_{\text{НИР}}$ – затраты на разработку проекта (из таблицы 4.9), тг;

P – средний уровень рентабельности проекта. % (принимается в размере 20– 30% по согласованию с консультантом по экономической части).

Далее определяется цена реализации с учетом налога на добавленную стоимость (НДС), ставка НДС устанавливается законодательно Налоговым Кодексом РК. На 2018 год ставка НДС установлена в размере 12%. Цена реализации с учетом НДС рассчитывается по формуле (4.10):

$$C_{\text{р}} = C_{\text{д}} + C_{\text{д}} * \text{НДС}, \quad (4.10)$$

Приняв во внимание все предыдущие расчеты определим возможную (договорную) цену НИР. Примем рентабельность равную 25%.

$$C_{\text{д}} = 579591 * \left(1 + \frac{25}{100}\right) = 641\ 611 \text{ тг.}$$

Цена с учетом НДС:

$$C_{\text{р}} = 641611 + 641611 * 0,12 = 718\ 604 \text{ тг.}$$

4.4 Оценка социально-экономических результатов проектирования

Конечная стоимость проектирования и реализации ПП под платформу Android составляет 718 604 тг. В эту сумму входят все вероятные затраты на разработку приложения.

Большей частью расходов, а это целых 50%, является оплата труда.

Создание мобильного приложения не является дорогим проектом, но требует крупные интеллектуальные затраты.

По данным исследования BranchFire, 49% пользователей готовы оплачивать ежемесячную подписку. Из этого следует что, продвигая приложение через Google Play Market бесплатно с подпиской 2,19\$ (718тг) в месяц, при 1000 скачиваний приложение окупится через 2,5 месяца.

На текущий момент, платформа Android является лидирующей ОС среди всех остальных. На первый квартал 2016 года, доля рынка составляла 83,4%, а на первый квартал 2017 года, доля составляла 85%. Это характеризует относительно низкий, но все же существующий рост количества устройств на ОС Android. Поэтому разработка именно на эту платформу является более актуальной, т.к. большая часть смартфонов работают именно под этой операционной системой, и количество смартфонов, следовательно и пользователей, растет с каждым годом (рисунок 4.1).

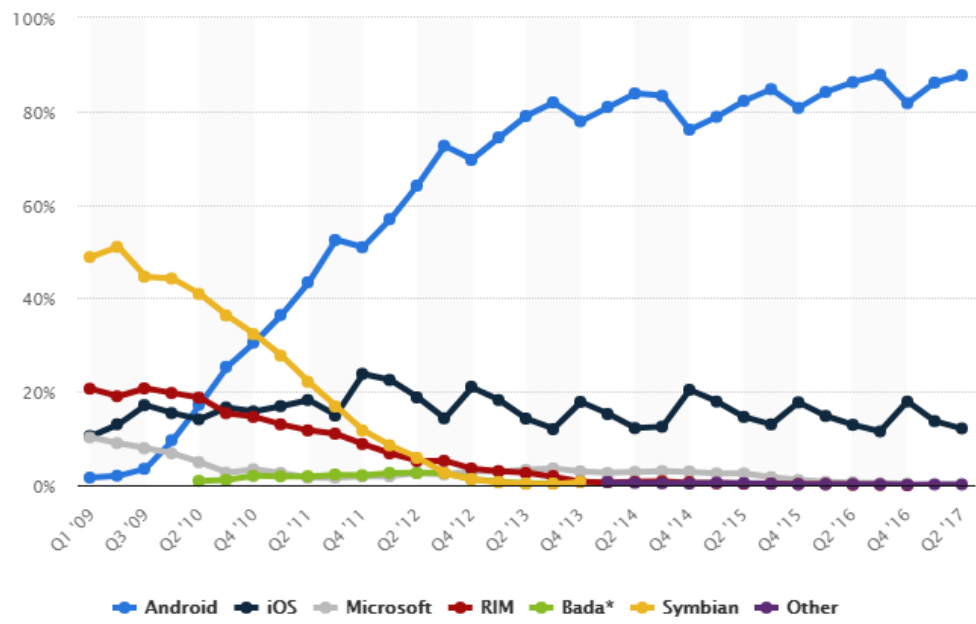


Рисунок 4.1 – Темпы роста использования платформы

5 Безопасность жизнедеятельности

Основной функцией разрабатываемого приложения является ведение базы данных учет запчастей на складе. Функционал направлен на ремонтные сервисы занимающиеся починкой мобильных телефонов.

В рассматриваемом помещении ведёт свою деятельность сервис-центр по ремонту сотовых телефонов, что подразумевает наличие склада и хранение на нем различных телефонных запчастей.

К запчастям относятся также и аккумуляторные батареи (АКБ). Зачастую именно неправильное хранение и эксплуатация АКБ ведет к трагическим последствиям. Поэтому при хранении АКБ следует придерживаться следующих правил:

- АКБ должны храниться в помещении с низкой влажностью и контролируемой постоянной температурой;
- в случае если в помещении имеются нагревательные приборы, расстояние должно быть не менее 1м от точки нагрева;
- не стоит накрывать чем-либо АКБ;
- обязательно избегать попадания прямых солнечных лучей.

5.1 Анализ условий труда

В помещении где ведет свою деятельность сервис центр, работают два специалиста. Рабочий день длится с 11:00 до 21:00. Допускается перерыв на обед не более чем на 1 час. Время обеда специалист выбирает сам. Площадь помещения составляет 30м².

Инженеры работают за специальным столом. Рабочее место оснащено двумя удобными офисными стульями и дополнительным освещением. В помещении стоит кондиционер кассетного типа (рисунок 5.1).



Рисунок 5.1 – Кондиционер кассетного типа

Ниже приведена схема помещения с отображением рабочего места (рисунок 5.2).

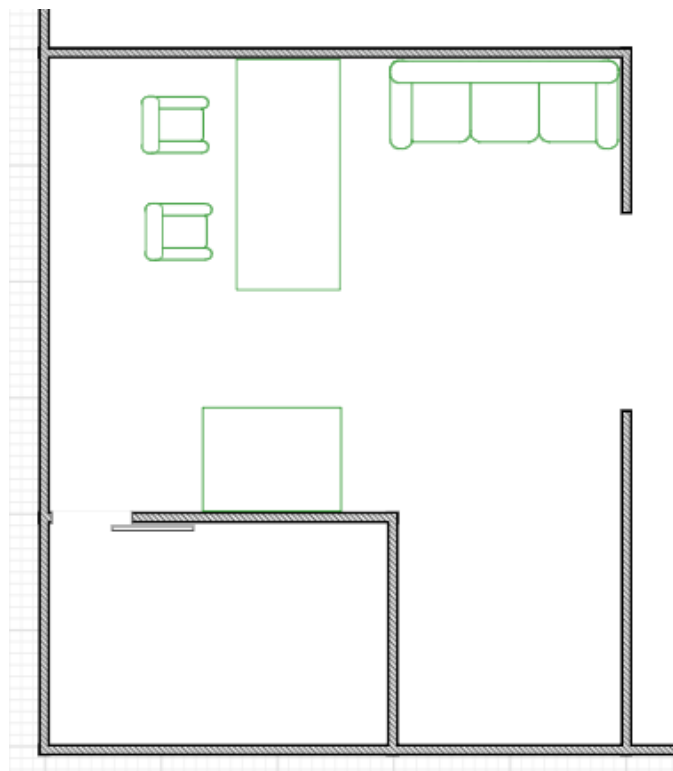


Рисунок 5.2 – Схема помещения

В помещении есть потенциальные источники возгорания. Основными являются электрооборудование и складское помещение. Во избежание несчастного случая следует спроектировать системы пожарного извещения. Также для обеспечения должной безопасности деятельности сотрудников нужно рассчитать необходимое количество тушащего средства для своевременного устранения пожара.

5.1.1 Системы пожарной сигнализации

Рассматриваемым объектом является центр мобильной связи. Здание одноэтажное. В помещениях размещены ценные товары и предметы утрата которых может привести к значительному материальному и финансовому ущербу. В связи с этим здание относится ко второй категории, согласно рекомендации Р 78.36.007-99.

Автоматические пожарные извещатели, за исключением световых устанавливаются в помещениях объекта на потолке. Так как потолок торгового дома является металлическим, извещатели подвешены на тросах длиной не более 300 мм от потолка.

При проектировании пожарной сигнализации были использованы дымовые и тепловые извещатели.

Дымовые извещатели (рисунок 5.3). будут устанавливаться для своевременного определения задымления какого-либо объекта (рисунок 5.4).



Рисунок 5.3 – Обозначение дымового извещателя

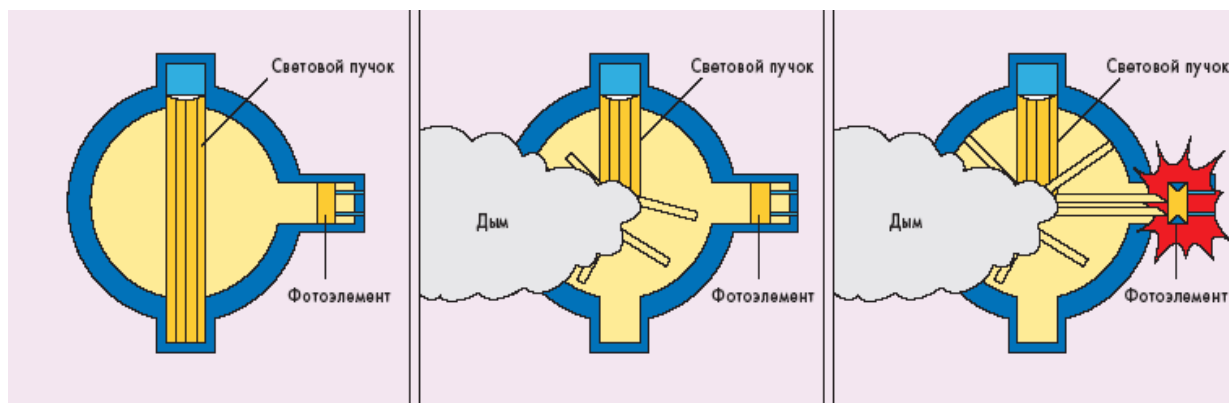


Рисунок 5.4 – Принцип работы дымового извещателя

Тепловые (рисунок 5.5) пожарные извещатели или тепловые датчики будут определять изменения температуры вокруг себя, чтобы передать данные при превышении норм (рисунок 5.6).

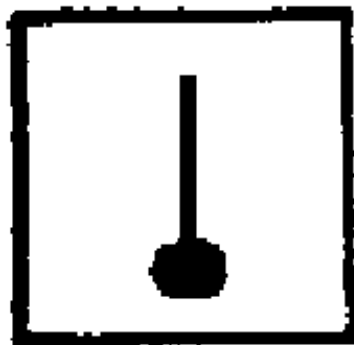


Рисунок 5.5 – Обозначение теплового извещателя

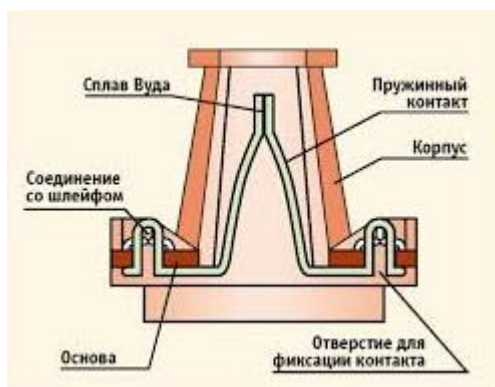


Рисунок 5.6 – Схема теплового извещателя

Площадь контролируемую одним извещателем, а также максимальное расстояние между извещателями и от извещателя до стены согласно СНиП 2.04.09 – 84 определяют по таблице 5.1

Таблица 5.1 – Расстояние при установке

Высота установки извещателя, м	Площадь контролируемая одним извещателем, м ²	Максимальное расстояние между извещателями, м	Максимальное расстояние извещателя от стены, м
Дымовые извещатели			
До 3,5	До 85	9	4,5
Свыше 3,5 до 6	До 70	8,5	4
Свыше 6 до 10	До 65	8	4
Свыше 10 до 12	До 55	7,5	3,5
Тепловые извещатели			
До 3,5	До 25	5	2,5
Свыше 3,5 до 6	До 20	4,5	2
Свыше 6 до 10	До 15	4	2

Площадь здания равна 288 м², высота потолков 3,2 м. Из этого следует, что теоретически на все помещение хватит 4 дымовых и 12 тепловых извещателей. На практике устройств требуется больше, т. к. следует учитывать архитектурные особенности здания.

На практике было установлено 6 дымовых извещателей, один из которых находится в нашем помещении. При планировке были учтены все требуемые максимальные расстояния, как от стены, так и от друг друга. При данной планировке извещатели контролируют все здание, что способствует своевременному определению очага возгорания и должной реакции на него (рисунок 5.7):

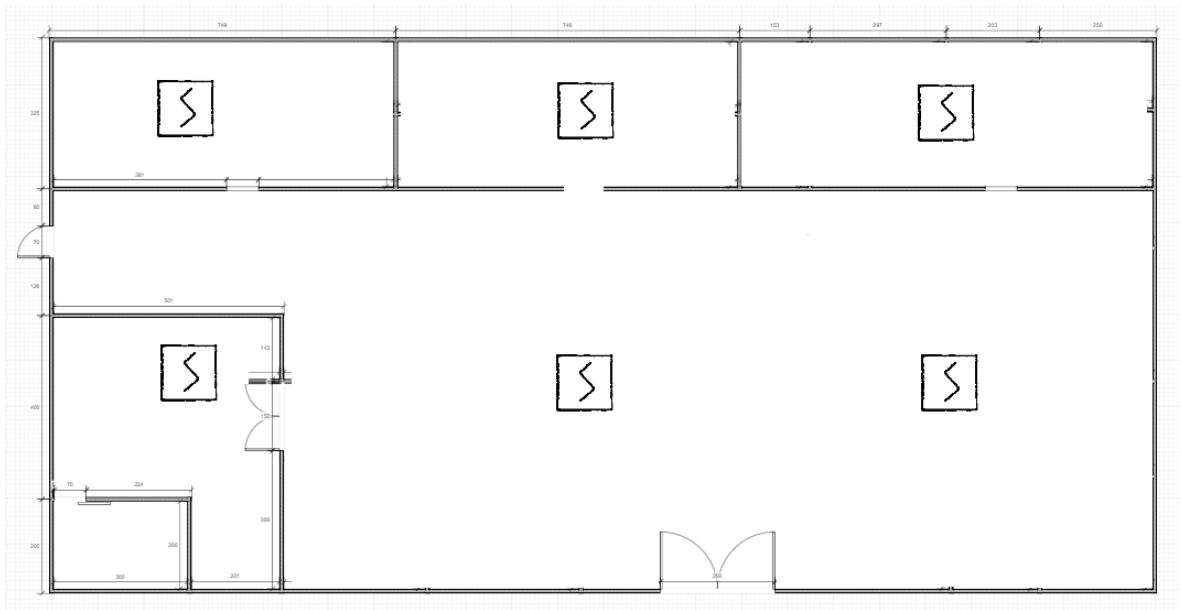


Рисунок 5.7 – Схема расположения дымовых извещателей

Также тепловые извещатели были установлены по схеме приведенной ниже (рисунок 5.8):

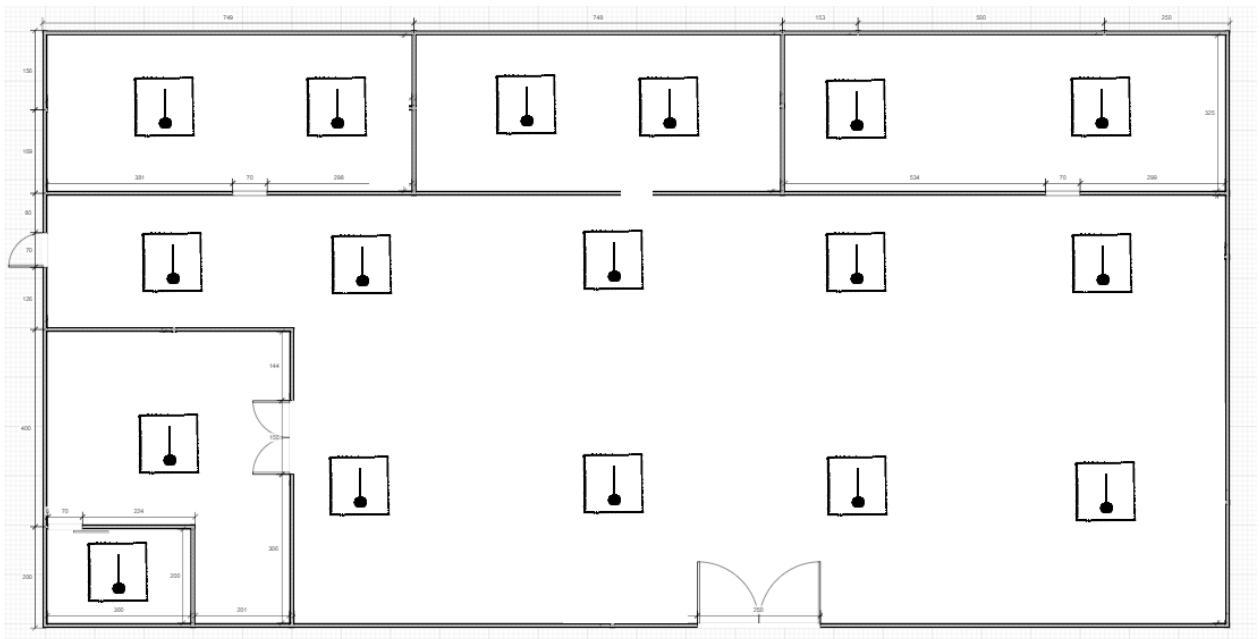


Рисунок 5.8 – Схема расположения тепловых извещателей

На практике потребовалось 6 дымовых и 17 тепловых извещателей для полного покрытия площади здания с целью обеспечения максимальной пожарной безопасности. Один из тепловых извещателей был установлен в нашем складском помещении, в качестве контроля над повышенным пожароопасным объектом.

5.1.2 Система автоматического пожаротушения

Согласно СНиП 2.04.09–84 и РНТП–01–94, сооружение относится ко второй категории так как оснащено ценными материалами потеря которых приведет к серьезным финансовым и материальным потерям. Возможные причины пожара следующие:

- несоблюдение техники безопасности при работе с оборудованием;
- несоблюдение норм хранения горючих материалов;
- высокая нагрузка на сеть, способная вызвать короткое замыкание.

Очагами возгорания могут служить различные электрически приборы, такие как паяльники, нагревательные печи, аппарат нагнетания давления, кондиционеры и так далее. В здании находится большое количество дорогостоящего товара, в основном это сотовые телефоны. Возможный пожар может принести колоссальный ущерб как в виде сгоревшего дорогостоящего товара, так и в виде человеческих жизней. Поэтому следует принять меры для увеличения вероятности борьбы с очагом пожара еще до его распространения.

В соответствии с требованиями пожарной безопасности помещений для хранения огнеупорных огнетушителей ОУ–5, с учетом – одного огнетушителя на 100 м². Общая площадь помещения составляет 30 м², поэтому установлен 1 огнетушитель. В качестве огнетушащего средства используется комбинированная смесь углекислого газа-хлорона. Рассчитанная масса объединенной композиции двуокиси углерода-хлорона m_d для объемного пожаротушения определяется формулой (5.9):

$$m_d = k g_n V, \quad (5.9)$$

где $k=1,2$ – коэффициент компенсации не учитываемых потерь углекислотно-хлоридного состава;

$g_n=0,04$ – нормативная массовая концентрация углекислотно-хлоридного состава.

V – объем помещения определяется по формуле (5.10):

$$V=A \cdot B \cdot H, \quad (5.10)$$

где $A = 6\text{ м}$ – длина помещения;

$B = 5\text{ м}$ – ширина помещения;

$H = 3,2\text{ м}$ – высота помещения.

Тогда:

$$V = 6 \cdot 5 \cdot 3,2 = 96 \text{ м}^3.$$

Следовательно:

$$m_d = 1,2 \cdot 0,04 \cdot 96 = 4,6 \text{ кг}.$$

Расчетное число баллонов x определяется из расчета вместимости в 20литровый баллон 12 кг углекислотно-хладонового состава.

Внутренний диаметр магистрального трубопровода d_i (мм), определяется по формуле:

$$d_i = 12 \cdot \sqrt{2} = 17 \text{ мм.}$$

Эквивалентная длина магистрального трубопровода l_2 определяется по формуле (5.11):

$$l_2 = k_1 \cdot l, \quad (5.11)$$

где $k_1 = 1,2$ – коэффициент увеличения длины трубопровода для компенсации не учитывающих местных потерь;

$l = 3,2 \text{ м}$ – длина трубопровода по проекту тогда:

$$l_2 = 1,2 \cdot 3,2 = 3,84 \text{ м.}$$

Расход углекислотно-хладонового состава Q , в зависимости от эквивалентной длины и диаметра трубопровода равна 1,4 кг/с.

Расчетное время подачи углекислотно-хладонового состава t , определяется по формуле (5.12):

$$t = m_d \div V \cdot Q, \quad (5.12)$$

Тогда:

$$T = 4,06 \div 96 \cdot 1,4 = 0,067 \text{ мин.}$$

Масса основного запаса углекислотно-хладонового состава определяется по формуле (5.13):

$$M = 1,1 \cdot m_d \cdot \left(1 + \frac{k_2}{k_1}\right), \quad (5.13)$$

где $k_2 = 0,2$ – коэффициент учитывающий остаток углекислотно-хладонового состава в баллонах и трубопроводах.

Тогда:

$$m = 1,1 \cdot 4,6 \cdot \left(1 + \frac{0,2}{1,2}\right) = 5,9 \text{ кг.}$$

Таким образом, из полученных результатов можно сделать вывод, что для обеспечения нормального функционирования системы автоматического пожаротушения потребуется 1 баллон углекислотно-хладонового состава

вместимостью 20 литров, с массой смеси 7,2 кг. Автоматические установки газового пожаротушения имеют устройства для автоматического пуска в соответствии с ГОСТ 12,4.009–83 (рисунок 5.9).

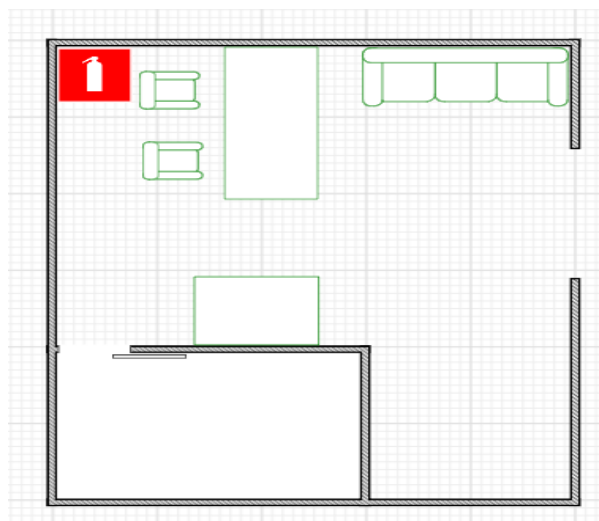


Рисунок 5.9 – Схема помещения

К возникновению пожара при работе на электрооборудовании может привести короткое замыкание, возникшее в цепях питания из-за выделения в них значительного количества теплоты и статического электричества, образующееся на экране монитора. Эксплуатация электрооборудования связана с необходимостью проведения обслуживающих, ремонтных и профилактических работ с использованием различных смазочных веществ, легковоспламеняющихся жидкостей; ведется пайка и чистка отдельных узлов и деталей. Возникает дополнительная пожарная опасность, требующая принятия соответствующих мер пожарной профилактики по очищению от пыли и грязи, а также смазываются механизмы в печатающем устройстве.

Стены помещения собраны из огнестойкого гипсокартона. Данный материал способен выдерживать до 25 минут прямого огня. Покрываются штукатуркой. Пол выполнен из кафеля, который в принципе не горит.

Для ликвидации пожара в начальной стадии в помещении предусмотрены следующие первичные средства пожаротушения; огнетушители типа ОУ-5.

Также в помещениях размещены; план эвакуации сотрудников и материальных ценностей в случае возникновения пожара, таблица с указанием номера телефона для связи с пожарной охраной и службой техобслуживания.

В здании имеются дымовые и тепловые извещатели установленные во всех кабинетах. Они предназначены для раннего обнаружения возгораний, сопровождающихся появлением дыма в помещениях различных зданий и сооружений. Применяется в системах пожарной сигнализации в качестве порогового датчика дыма. Передача сигналов о пожаре осуществляется путем

изменения тока, протекающего через шлейф. Извещатель включается в двухпроводные пороговые шлейфы любых ППКП (прибор приемо-контрольный пожарный).

Эвакуация людей в случае возникновения пожара будет производиться согласно плану эвакуации (рисунок 5.10).

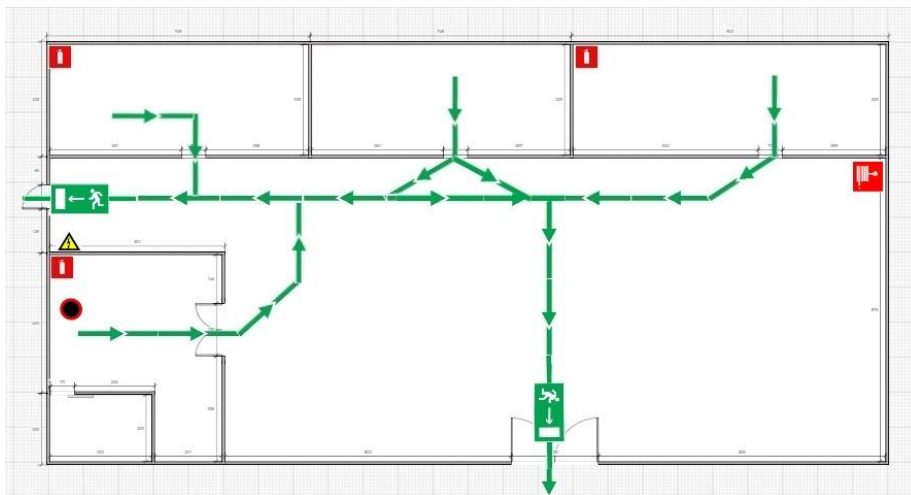


Рисунок 5.10 – План эвакуации

Согласно плану эвакуации, в здании имеются два выхода: главный и запасной. При пожаре люди распределяются и покидают здание по ближайшему к ним выходу.

Персонал каждый год проходит инструктаж по технике безопасности.

Заключение

Android развивается с колоссальной скоростью. С каждым годом платформа обновляется и появляются новые функции которые разработчики могут применить при создании приложений. Популярность разработки под андроид также растет из года в год. Это неудивительно ведь со временем это становится не таким уж и сложным делом.

Во время написания дипломной работы было разработано приложение способное вести базу данных. Также был изучен рынок мобильных телефонов. Проведен анализ трендов в создании мобильных приложений

Целевая аудитория моего приложения это мелкие предприниматели занимающиеся ремонтом сотовых телефонов. Данная отрасль очень развита в Алматы, но не у всех предпринимателей есть хоть какой-то учет. Вследствие этого теряется важная часть потока клиентов. Приложение очень понятно и с ним разберется даже человек далекий от понимания ведения учета. Существует великое множество систем учета: 1С, различные CRM системы, Excel и т. д. Но не все способны ими овладеть, ведь у 1С слишком много функций, из-за которых люди боятся братья за него, а Excel не имеет приятного интерфейса.

В долгосрочной перспективе у приложения должен расширяться функционал и охват возможностей. При этом простота не должна теряться, ведь это её главный козырь.

На сегодня по всему миру отслеживается рост, как числа мобильных устройств, так и их пользователей. Многие люди не представляют жизнь без смартфонов. А работа у немалого количества людей непосредственно связана с мобильными устройствами. С каждым годом производятся множество мобильных устройств, и каждые новые версии мощнее, быстрее, практичнее предыдущих моделей.

Список литературы

- 1 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 – Информационные системы – Алматы: АУЭС; 2013. –24 с.
- 2 Курсы программирования <https://developers.google.com/training/android/> [Электронный ресурс].
- 3 Forbes, How Much Do Average Apps Make. [Электронный ресурс].
- 4 "Build Overview". android.com. [Электронный ресурс].
- 5 Haslam, Oliver (May 16, 2013). "Download Android Studio IDE For Windows, OS X And Linux". Redmond Pie. Retrieved May 16, 2013. [Электронный ресурс].
- 6 Dobie, Alex (May 15, 2013). "Android Studio unveiled at Google I/O keynote". Android Central. Mobile Nations. Retrieved May 16, 2013. [Электронный ресурс].
- 7 Forrester Research, Тед Шадлер [Электронный ресурс].
- 8 Блох Д. «Эффективное программирование», 3 – издание БХВ-Петербург, 2016. - 560 с. (2014).
- 9 Фаронов В.В. Delphi 6: Учебный курс. – СПб.: Питер, 2002.
- 10 Климова, Л. М. Delphi 7. Основы программирования. Решение типовых задач. Самоучитель / Л.М. Климова. - М.: КУДИЦ-Образ, 2017. - 480 с.
- 11 Б. Харди, Б. Филлипс, К. Стюарт, К. Марсикано, «Android. Программирование для профессионалов», 2016
- 12 Культин, Никита Основы программирования в Delphi XE / Никита Культин. - М.: "БХВ-Петербург", 2011. - 416 с.
- 13 Санников, Е. В. Курс практического программирования в Delphi. Объектно-ориентированное программирование / Е.В. Санников. - М.: Солон-Пресс, 2013. - 188 с.