

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой _____
Журабаев Э.К.
(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка мобильного приложения „CityInfo“
на платформе Android.

Специальность 5В070400 - Вычислительная техника и программное обеспечение

Выполнил (а) Срейманова Э.М. БТ-14-2
(Фамилия и инициалы) группа

Научный руководитель к.ф.-м.н., доцент Шайхин Б.М.
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмурзаева А.Ч., к.э.н., доцент
(Фамилия и инициалы, ученая степень, звание)
А-Ч « 04 » 06 2016 г.
(подпись)

по безопасности жизнедеятельности:

(Фамилия и инициалы, ученая степень, звание)
« _____ » _____ 20__ г.
(подпись)

по применению вычислительной техники:

Шайхин Б.М. к.ф.-м.н., доц.
(Фамилия и инициалы, ученая степень, звание)
Б.М. « _____ » _____ 20__ г.
(подпись)

Нормоконтролер: Шайхин Б.М. к.ф.-м.н. доц.
(Фамилия и инициалы, ученая степень, звание)
Б.М. « _____ » _____ 20__ г.
(подпись)

Рецензент: _____
(Фамилия и инициалы, ученая степень, звание)
« _____ » _____ 20__ г.
(подпись)

Алматы 2016 г.

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1.	Критерии разработки лит. цитат	22.02 - 18.03	
2.	База знаний	17.03 - 17.04	
3.	Практическая часть	18.04 - 25.05	
4.	Технико-эконом. обоснование	29.05 - 2.06.	
5.	Безопасность жизнедеятельности	18.03 - 02.06	

Дата выдачи задания « 15 » 10 20 15 г.

Заведующий кафедрой _____
(подпись) (Фамилия и инициалы)

Руководитель _____
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент _____
(подпись) (Фамилия и инициалы)

Аңдатпа

Ғылыми-зерттеу жұмысында картадағы мейрамханалар мен кинотеатрлар табу оңай, сондай-ақ олардың толық ақпарат назар аударатын Android платформасында «CityInfo» қосымшасы әзірленді.

Жұмыс SQLite технологиясын пайдаланып Android Studio әзірлеу ортасында Java технологиясына негізделген болатын.

«CityInfo» мобильді қосымшасы туристер мен Алматы қаласының тұрғындары мекемелердің егжей-тегжейлі сипаттамасын қарап, өз қалауына сай мейрамханаларды немесе кофеханаларды таңдауға мүмкіндік береді. Сонымен қатар, пайдаланушылар кинотеатрды таңдау және кино кестесін көре алады. қосымшаның өзектілігі - Алматы қаласының барлық орындарын қаланың картасында, және оған бағыт табуға болады

Аннотация

В дипломной работе было разработано мобильное приложение «CityInfo» на платформе Android, ориентированное на удобство поиска ресторанов и кинотеатров на карте, а также их полной информации.

Работа выполнена на основе технологии программирования Java в среде разработки Android Studio с использованием технологии SQLite.

Мобильное приложение “CityInfo” позволяет туристам и жителям города Алматы выбрать рестораны или кофейни на свой вкус, просмотрев подробное описание заведений в приложении. Также пользователи могут выбрать кинотеатр и посмотреть расписание фильмов. Актуальность приложения состоит в том, что все заведения Алматы вы можете найти на карте города, а также маршрут к ним.

Annotation

The diploma project has been developed a mobile application «CityInfo» on the Android platform that focuses on ease of finding restaurants and movie theaters on the map, as well as their complete information.

The work is based on the Java technology in Android Studio development environment using SQLite technology.

"CityInfo" mobile application allows tourists and residents of Almaty city to choose taste restaurants or coffee houses by looking at a detailed description of institutions in the application. Users can also select a movie theater and see the movie schedule. The urgency of the application is that all the places of Almaty can be found on the city map, as it's route.

Содержание

Введение.....	5
1 Принципы разработки.....	6
1.1 Информационные технологии и описание ОС Android.....	6
1.2 Принцип работы Android OS	7
1.3 Компоненты пользовательского интерфейса Android.....	15
2 Инструменты разработки	21
2.2. Эмулятор Android.....	23
3. Разработка базы знаний	25
3.1 Подготовка к работе.....	25
3.2 Проектирование базы знаний.....	28
3.3 Результаты работы приложения	31
4 Техничко-экономическая часть	6
4.1 Определение задач и объяснение эффективности	6
4.2 Расчет окончательных материальных затрат на все компоненты программного продукта.....	6
4.3 Расчет цены программного продукта.....	14
5 Безопасность жизнедеятельности.....	16
5.1 Анализ условий труда.....	16
5.2 Расчет системы кондиционирования	21
5.3 Расчет системы автоматического пожаротушения.....	25
Список используемой литературы	30
Приложение А	31

Введение

Мобильные смартфоны давно перестали быть чем-то удивительным и замечательно справляются со своей важнейшей функцией – средством связи между людьми. Но при этом, эти смартфоны, основательно вошедшие в нашу жизнь, в такой степени функциональны и полезны, что трудно представить жизнь без них. На данный момент в их функции входит и камера, и медиаплеер, и интернет браузеры, и очень много всего, что значительно облегчает нашу повседневную жизнь. Сейчас с полной уверенностью можно сказать, что смартфоны – это маленькая копия компьютера, который также функционален, и всегда под рукой в любой момент времени.

В настоящее время множество смартфонов, планшетов и других разновидностей устройств, предназначенных для удобства использования как в обычной жизни, так и в путешествиях, выпускаются на базе ОС Android. Так какие же основные причины распространения и популярности этой операционной системы?

Во-первых, ОС Android поддерживает очень большое количество устройств различных производителей. Во-вторых, ОС Android отличается высокой доступностью средств инструментальной разработки. Средства разработки для платформы Android бесплатные, в отличие от iPhone (компания Apple) которые требуют больших первоначальных финансовых инвестиций. Помимо всего этого перечисленного, одним из важных преимуществ ОС Android является наличие бесплатных библиотек для работы с другими ресурсами (Google Map API, Yandex Map др.), тем временем как на Windows Phone Mobile эти библиотеки не распространяются.

Целью данной дипломной работы является разработка мобильного приложения «CityInfo» для операционной системы Android, ориентированный на удобство поиска ресторанов и кинотеатров на карте, а также их полной информации. Указанные вышеперечисленные преимущества объясняют широкое распространение современных устройств на платформе Android, оснащенных разными функциями и приложениями, делающими отдых в городе Алматы максимально удобным. Ориентированность на студенческий отдых объясняется тем, что большое количество студентов являются пользователями современных смартфонов, планшетов и других устройств.

1 Принципы разработки

1.1 Информационные технологии и описание ОС Android

Двадцать первый век по праву можно считать веком информационных технологий. Во множествах сферах нашей жизни без них, невозможно было бы обойтись. Сфера туризма не исключение. В наши дни для того, чтобы комфортно чувствовать себя в чужом городе, стране достаточно иметь при себе современный коммуникатор. Так какие же основные потребности у туристов? Обзор достопримечательностей, питание, проживание, и на конец, шоппинг и покупка сувениров на память – все это становится намного удобнее, когда рядом есть смартфон и выход в интернет! Забронировать ресторан, посмотреть на карте интересующее место, выбрать кинотеатр и посмотреть расписание фильмов – все это можно быстро и удобно сделать в режиме онлайн. Поэтому нельзя уменьшать важность туристической информационной поддержки в сети Интернет.

В особенности, интернет-ресурсы Алматы – города, в котором, я считаю, организованы недостаточно удобно. Для того, чтобы построить план прогулки по городу, необходимо просмотреть сразу не один десяток сайтов. Однако Южная Столица ежегодно принимает множество туристов из разных уголков мира, желающих посмотреть Медео, Чимбулак, Большое Алматинское озеро и др достопримечательности. Вопрос об организации удобной и достаточно-необходимой интернет-поддержки туризма в городе Алматы становится все более и более актуальным.

Итак, Алматинские интернет-ресурсы по организации во многом проигрывают другим. В связи с данной проблемой, в рамках моей дипломной работы область интереса была сжата вопросами студенческого туризма в городе Алматы. Алматы – это город с богатым культурным наследием, красивой природой, что активно способствует привлечению туристов. Большой процент туристов посещают студенты. Из этого следует, что целесообразнее развивать интернет-поддержку интересных культурных мест, которые будут интересны молодому поколению.

Подводя итог, на основе вышерассмотренных фактов можем сделать следующий вывод: в настоящее время актуальность организации интернет-поддержки туризма в городе Алматы с уделением большого внимания разработке практичного в использовании интерфейса с наиболее необходимым набором функций наиболее актуальна.

Целью данной дипломной работы является разработка мобильного приложения «CityInfo» для операционной системы Android, ориентированный на удобство поиска ресторанов и кинотеатров на карте, а также их полной информации. Указанные вышеперечисленные преимущества объясняют

широкое распространение современных устройств на платформе Android, оснащенных разными функциями и приложениями, делающими отдых в городе Алматы максимально удобным. Ориентированность на студенческий отдых объясняется тем, что большое количество студентов являются пользователями современных смартфонов, планшетов и других устройств.

Операционная система Android одна из наиболее популярных платформ для мобильных устройств. Учитывая, что это не дешевый аналог продукта от Apple, а его альтернатива, основной конкурент с высочайшей динамикой развития. Платформа Android разработана на ядре Linux. Гибкость настроек системы Android наиболее удачно согласовывается с хорошим инструментарием, а это является наилучшим решением для создания новых приложений. Android (Андроид) – портативная (сетевая) операционная система для коммуникаторов, планшетов, Ebook, цифровых проигрывателей, наручных часов, нетбуков и смартбуков, основанных на ядре Linux. Сначала разрабатывалась компанией Android Inc., которая была куплена Google. Основным источником ПО для андроид-устройств является Android Market сайт с ассортиментом, перевалившим за 300 тысяч программ (у Apple App Store более 500 тысяч). При этом используют его как бесплатный андроид-маркет, так и платный.

И так что такое андроид рассмотрим более детально. ОС Android, если бы не Google, могла бы закончить свое существование так и не начавшись. В авторитет экспертов данной компании в цифровых разработках уже давно никто не сомневается. Специалисты Google, сделав ставку на Андроид, не прогадали. Другие компании, которые отказались от сотрудничества с разработчиками Андроид, поняли свою ошибку, когда уже Google занял доходное положение.

В истории Андроид, 2005 год - наиважнейший год. Примерно в середине 2005 года руководство Google и директор Android Inc (на тот момент это была маленькая студия, которая вела всю работу в одиночку) подписали соглашение о слиянии. Вот начиная с этого момента и началась история Android — операционной системы, под которой сегодня работают миллионы мобильных устройств, и компании, ставшей частью Google. К слову, основателей Android Inc Google принимают на работу в качестве экспертов. На тот момент, много экспертов догадывались, что Google собирается пустить в ход все свои возможности для раскрутки многообещающей платформы. Оставалось не понятно, что же именно будет сделано компанией, так как ей еще не доводилось заниматься реализацией подобных проектов. На тот момент времени у Android Inc уже была первая версия ОС, которая была в работе. В основе этой платформы было ядро Linux 2.6, менеджеры Google стали предлагать ее разработчикам сотовых телефонов и операторам мобильной связи. Ближе к концу 2007 года инновационной операционной системой всерьез проявил интерес консорциум OpenHandsetAlliance. Разработчики консорциума

официально объявили, что готовы сделать все самое нужное для развития платформы Андроид. Наиболее ранняя версия была хорошим вариантом для готовых решений. Помимо этого, она поддерживала VGA, OpenGL и 2D библиотеки. Система Андроид на тот момент могла работать во всех известных диапазонах связи, она поддерживала WiFi, Bluetooth, 3G. Используя эту операционную систему, пользователи могли обмениваться мгновенными потоковыми сообщениями, еще SMS и MMS. В системе находился собственный браузер, полнейшая поддержка Java и ускоритель трехмерной графики. Еще пользователи могли заценить поддержку Android-видеокамеры, тачскрин и GPS-навигатор. В 2007 другие разработчики смогли показать, на что они способны — мероприятие Android DeveloperChallenge было очень интересным соревнованием. Цель соревнования заключалась в разработке приложения для платформы Android, программы любого направления. Организаторы объявили только одно примечание, что большое внимание при выборе победителя будет уделяться инновациям и полезности продукта. Андроид-приложения для социальных сетей, новостных лент, развлекательные программы и интернет-пейджеры были рассмотрены в особом порядке. Май 2009 года был очень знаменательным днем для разработчиков мобильного контента — мероприятие Android DeveloperChallenge прошло вновь, только на этот раз победителя ожидали 2 миллиона долларов. В 2008 году код ОС Андроид был опубликованным, а также с ним в открытом доступе появился и код встроенных в систему приложений. Настоящую огласку платформа Андроид получила 23 сентября 2008 года. В этот знаменательный день Google презентовала первый смартфон, оснащенный системой Android. Смартфон T-Mobile G2 заценили большие производители мобильных телефонов, после проведения этой презентации, производители решаются оптимизировать свои устройства под эту платформу. Система хорошо подходила для бюджетных смартфонов, а открытый код, естественно, не смог китайцам пройти мимо. Наибольшие мобильные производства поднебесной стали производить «китайский андроид» — телефоны с учетом особенностей Android ОС. Данное положение дел очень укрепило позиции платформы на рынке. Больше 76 % смартфонов, проданных во втором квартале 2013 года, были оснащены операционной системой Андроид. На данный момент каждый разработчик электронного устройства может переделать Android под свое устройство, значит гарантируя совместимость своего оборудования со сторонними приложениями для Андроид. Как оказалось, это очень выгодно. К примеру, если до выхода Android любой производитель электронного устройства собственноручно писал или закупал у кого-то операционную систему, потеряв массу нужных программ, которые создавались программистами всего мира, а значит после выхода ОС Android перед производителями часто встает вопрос, какую же версию Android им лучше поддерживать. Огромнейшую роль сыграло появление Android и в коммуникации между пользователями. Устанавливая его на телефоны, есть

возможность легко разрабатывать новые модели мобильных устройств, делая шире функционал – и телефонов, и самой операционной системы. После появления программ, предназначенных для оказания помощи пользователю в самых различных ситуациях (к примеру, есть приложение, которое использует встроенные датчики для замера углов поворота, скорости объекта и т.д.), и это привело к тому, что на сегодняшний день человеку, который отправляется в путешествие, довольно при себе иметь мобильное устройство на ОС Андроид. Людям предоставляются сервисы бронирования отелей, поиск авиабилетов, разные программы-гиды, а также специально загружаемые карты, которые служат для поиска и построения маршрута до нужного пункта назначения. При использовании SIP-телефонии можем иметь коммуникацию на междугородном и международном уровнях. Все это, вместе с легкой разработкой приложений, создает рассматриваемую нами платформу одной из наиболее перспективных для коммуникации в современном мире.

Раньше рынок мобильных платформ делился как минимум на 6-7 компаний (Symbian, Blackberry OS, Android, Palm, iOS, Bada и т.д.), но вот на данный момент мы можем выделить только 2 основные платформы: iOS и Google Android, которые разделяют около 90% всего рынка. Часть доли Google Android на четвертый квартал 2012 года достигла 76%, это значит три из четырех телефонов, работают под управлением мобильной платформы от Google. Ниже мы привели таблицу сравнения разных операционных систем (таблица 1) и данные по разным версиям Android (таблица 2). Статистика собрана за двухнедельный период на основе обращений к сервису Google Play.

Таблица 1. Мировые продажи смартфонов, по типам ОС, за 1-й квартал 2012

ОС	Продажи в 2012г.,	%	Продажи в 2011г.,	%
	тыс.	рынка	тыс.	рынка
Android	81 457,4	57,1	36 350,1	36,4
iOS	33 240,5	21,9	16 653,2	16,9
Symbian	12 498,9	8,6	27 523,5	27,7
Blackberry OS	9 989,3	5,9	13 354,0	13,0
Bada/Linux	3 072,2	3,7	1 562,2	1,9
Windows	2 784,5	1,9	2 242,1	2,6
Прочие	1 243,9	0,7	1 655,0	1,5
Суммарно	144 305,7	100,0	99 345,0	100,0

Таблица 2. Распределение версий ОС Android, на 1-е августа 2012

Название версии	Уровень API	Распределение
1.5 Cupcake	3	0.2%
1.6 Donut	4	0.5%
2.1 Eclair	7	4.2%
2.2 Froyo	8	15.5%
2.3 – 2.3.2 Gingerbread	9	0.3%
2.3.3 – 2.3.7	10	60.3%
3.1 Honeycomb	12	0.5%
3.2	13	1.8%
4.0 – 4.0.2 Ice Cream Sandwich	14	0.1%
4.0.3 – 4.0.4	15	15.8%
4.1 Jelly Bean	16	0.8%

Операционная система Android постоянно в развитии и не останавливается на одном. На данный момент она не считается абсолютным лидером, но все же тенденции говорят о том, что в конечном счете станет лидером именно Android. Google не привыкли уступать. Помимо того, что множество экспертов цифрового рынка уже не в первый раз объявляли, что будущее — за Андроид. На рынке мобильных устройств разворачивается настоящая война операционных систем, и почти все уже знают, кто станет победителем.

1.2 Принцип работы Android OS и Android приложений

Приложения под операционную систему Android разрабатываются в основном с использованием языка Java. Скомпилированный программный код (вместе со всеми файлами ресурсов и остальной нужной информацией) упаковывается в определенный файл-архив, Android Package. У данного файла расширение *.apk и упаковывается специальной утилитой aapt tool. Именно он в последствии распространяется как программа и устанавливается на мобильные устройства. Один такой файл связан с кодом одного приложения. И такое приложение в Android находится в своем собственном мире – в такой машине. По умолчанию, каждая программа выполняется в своем собственном процессе, ядро Linux занимается его управлением, которое тоже может осуществлять менеджмент памяти. Это значит, что чаще всего код приложения выполняется в

изоляции от остальных приложений. Android начинает процесс, когда возникает нужно выполнить какой-то программный код и завершает его, когда в нем больше нет нужды и системные ресурсы нужны остальным приложениям. По умолчанию, каждому приложению должен присваиваться свой уникальный ID Linux-пользователя. Права доступа устанавливаются так, что файлы приложения могут быть видны только одному пользователю данного приложения. Однако, есть возможности, которые позволяют экспортировать их в остальные приложения. К примеру, есть возможность «разделения» одного и того же пользовательского ID между двумя приложениями. В данном случае, у них будет возможность видеть файлы друг друга. Для этого нужно экономить системные ресурсы, приложения с одинаковым ID. Можно также обговорить запускать на одном и том же Linux-процессе, разделяя одну и ту же виртуальную машину. Отличительной чертой операционной системы Андроид является то, что приложение в ней может применять для своего функционала элементы других приложений, только если, они дают такой доступ. К примеру, если нашему приложению нужно отобразить листаемый список изображений, а другого приложения уже есть реализованный подходящий скроллер, который открыт для других приложений, то его можно просто вызвать для осуществления этого действия и не разрабатывать свой собственный. При этом код нашего приложения не будет смешиваться с кодом другого и не будет компоноваться с ним. Точнее, он просто запускает какой-то кусочек другого файла, когда появляется такая надобность. Для обеспечения данного принципа работы, система должна уметь запускать процесс приложения тогда, когда нужна будет любая его часть и создавать экземпляры java-объектов специально для этой части. Поэтому приложения Android нет единой точки входа, как это обычно принято в большинстве систем.

Помимо этого их код представляет собой набор каких то разных целостных сущностей, компонент, из которых система по мере их надобности может создавать экземпляры и использовать их. Можем выделить 4 типа таких сущностей-компонент. Опишем каждую в отдельности. Визуальное содержание окна строится с помощью иерархии визуальных компонентов (или представлений) – объектов, образованные от базового класса View. Каждый компонент показывает просто прямоугольное пространство внутри окна. Родительские компоненты включают дочерние и организуют их расположение. Иерархию компонент можно показать в виде дерева, а те элементы, которые располагаются в наиболее нижней его части (“листья”) и не имеют дочерних компонент, вырисовывают прямоугольные области и ждут действий пользователя на этом участке. Так происходит интерактивное взаимодействие с пользователем. На пример, данное представление может отобразить на экране маленькую иконку и инициировать как какое-нибудь действие, при нажатии пользователем на неё. В операционной системе Android существует набор готовых визуальных компонент, которые доступны для применения

разработчиками. Набор содержит в себе кнопки, текстовые поля, полосы прокрутки, меню, флажки-переключатели и др.



Архитектура операционной системы Android.

Если взять компонентную модель Android в виде какой то иерархии, то в конце, как самая фундаментальная и главная составляющая, расположится ядро операционной системы Андроид. Еще компонентную модель называют программным стеком. В самом деле, это определение правильно, так как разговор идет о наборе программных продуктов, работающие вместе для того, чтобы получить итоговый результат. Действия в этой модели происходят последовательно, и уровни иерархии тоже последовательно взаимодействуют между собой. Известный факт, что Андроид основан на нескольких урезанных ядрах ОС Linux и поэтому на этом уровне мы можем увидеть определенно его (версии 2.6.x). Оно дает функционирование системы и работает за безопасность, управление памятью, энергосистемой и процессами, а также дает сетевой стек и модель драйверов. Ядро еще работает как уровень абстракции между аппаратным обеспечением и программным стеком.

2.3 Архитектура и реализация

Программное приложение для ОС Android состоит из нескольких активностей, каждой из которых соответствует экран приложения. Каждая активность показана в проекте классом, реализованном на языке Java, хранящемся в одноименном файле с расширением .java. Каждой активности соответствует xml файл-описание. В xml-файле описано в виде xml-кода расположение визуализируемых объектов. При открытии активности система Android сразу распознает размер экрана мобильного устройства и приводит выводимый контент в соответствие с разметкой, описанной в xml-файле. Таким образом, одна и та же активность будет выглядеть одинаково независимо от диагонали используемого устройства. Также, для каждого приложения Android должен существовать xml-файл, в котором в виде xml-кода будут прописаны минимальные требования к системе, а также активность, вызываемая при запуске приложения. Приложение работает со встраиваемой реляционной базой данных SQLite. SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite.

Данный подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа.

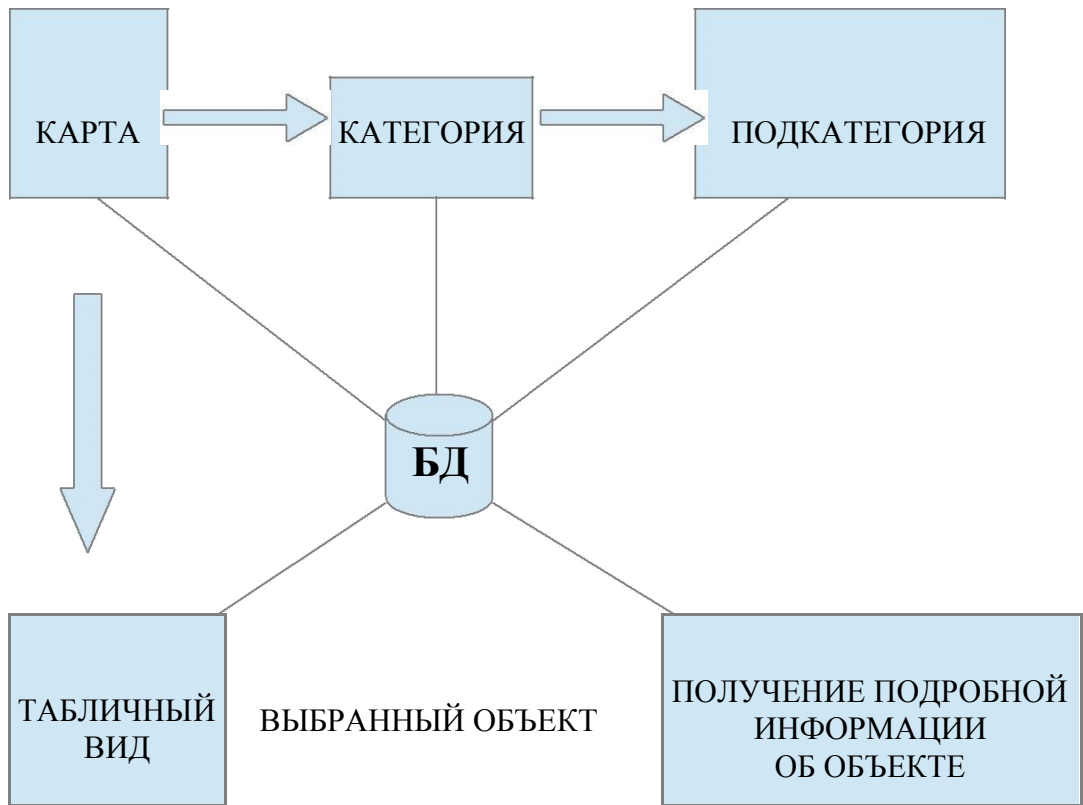


Рис. 5. Схема распространения запросов (указаны линиями) и потоков данных (указаны стрелками) от базы данных к модулям приложения

На рисунке 5 представлена диаграмма потоков данных. Потоки данных в рассматриваемой диаграмме распространяются от базы данных к остальным элементам посредством SQL-запросов. При загрузке приложения основная активность обращается к базе данных, находящейся на диске. Если на диске база данных не обнаружена, то создается пустая база данных и происходит её обновление через интернет.

Таким образом, пользователь может выбрать интересующие его категории объектов. При выборе категории в меню загружается активность «Категория», которая обращается к базе данных и считав возможные категории, отображает их в списке. При выборе категории загружается активность «Подкатегория». С помощью механизма передачи параметров между активностями ей передается информация о категории. Эта активность тоже обращается к базе данных (формируя запрос) и выводит все подкатегории. При выборе любой подкатегории формируется запрос на изменение базы данных. Таким образом, база данных всегда будет знать об интересующих подкатегориях и их не придется вводить после каждого перезапуска приложения.

1.3 Компоненты пользовательского интерфейса Android

В Android используется UI-фреймворк, сравнимый с другими полными функциями UI-фреймворками, используемые на локальных компьютерах. Он является более современным и асинхронным по природе. По существу, UI-фреймворк Android принадлежит уже к четвертому поколению, если учитывать, что первым поколением традиционный прикладной интерфейс программирования Microsoft Windows, основанный на C, а MFC (Microsoft Foundation Classes, библиотека базовых классов Microsoft на основе C++) - вторым. В этом случае UI-фреймворк Swing, основанный на Java, будет считаться третьим поколением, так как предлагаемые в нем возможности дизайна значительно превосходят по гибкости MFC. Android UI, JavaFX, Microsoft Silverlight и язык пользовательских интерфейсов Mozilla XML (XUL) относятся к новому типу UI-фреймворков четвертого поколения, в котором UI является декларативным и поддерживает независимую темизацию.

При программировании в пользовательском интерфейсе Android используется объявление интерфейса в файлах XML. Далее эти определения представления (view definitions) XML загружаются в приложение с пользовательским интерфейсом как окна. Даже меню приложения загружаются из файлов XML. Экраны (окна) Android часто называются активностями (activities), которые включают в себя несколько видов, нужных пользователю, чтобы выполнить логический элемент процесса. Виды (views) являются основными элементами, из которых в Android состоит пользовательский интерфейс. Виды можно объединять в группы (view groups). Для внутренней организации видов используются давно известные в программировании концепции холст (canvas), рисование (painting) и взаимодействие пользователя с системой (user interaction).

Эти составные представления, в которые входят виды и группы видов, работают на базе специального логического заменяемого компонента пользовательского интерфейса Android.

Одной из ключевых концепций фреймворка Android является управление жизненным циклом (lifecycle) окон явлений (activity windows). В системе применяются протоколы, поэтому Android может управлять ситуацией по мере того, как пользователи скрывают, восстанавливают, останавливают и закрывают окна явлений.

Базовые компоненты Android

Фреймворк пользовательского интерфейса Android вместе с другими компонентами Android базируется на новой сущности, называемой намерением (intent). Намерение - это сложное явление, в котором сочетаются такие идеи, как сообщения, выводимые в окнах (windowing messages), действия (actions), модели типа "публикация и подписка" (publish-and-subscribe), межпроцессный обмен информацией и реестры приложений. Ниже приведен пример использования класса Intent для активации или запуска веб-браузера:

```

public static void invokeWebBrowser (Activity activity)
{
Intent intent = new Intent (Intent. ACTION_VIEW):
intent. setData (Uri. parse ("http://www.google.com"));
activity. startActivity (intent);
}

```

На этом примере, используется намерение, мы имеем ввиду Android открыть окно, подходящее для отображения контента web-сайта. В зависимости от того, какие браузеры установлены на мобильном устройстве, Android выберет для отображения сайта наиболее подходящий.

Кроме того, в Android широко поддерживаются ресурсы (resources), к которым относятся хорошо знакомые вам строки (strings) и растровые изображения (bitmaps), а также некоторые менее известные элементы, в частности, основанные на XML определения представления. Использование ресурсов в данной фреймворке осуществляется новым способом, благодаря которому работа с ресурсами становится более простой, понятной и удобной. Ниже приведен пример, в котором автоматически генерируются ID ресурсов, определенных в файлах XML:

```

public final class R {
public static final class attr { }
public static final class drawable {
public static final int myanimation=0x7f020001;
public static final int numbers19=0x7f02000e:
}
public static final class id {
public static final int textViewId1=0x7f080003;
}
public static final class layout {
public static final int frame_animations_layout=0x7f030001;
public static final int main=0x7f030002;
}
public static final class string {
public static final int hello=0x7f070000;
}
}

```

В данных классах автоматически сгенерированные ID соответствуют либо элементу XML-файла, либо целому такому файлу. Если понадобится использовать такие определения XML, вместо них будут применяться данные ID. Такая опосредованность очень помогает при локализации. Еще одна

новаторская концепция в Android - это поставщик содержимого (content provider). Под поставщиком содержимого понимается абстракция источника данных, который можно представить как эмиттер и потребитель служб REST (Representational State Transfer, передача состояния представления). Лежащая в основе этой абстракции база данных SQLite делает данное свойство поставщиков содержимого мощным инструментом для разработки приложений. Уже было отмечено, что XML играет важнейшую роль в описании пользовательских интерфейсов Android. На примере рассмотрим, как XML выполняет эту задачу с простым шаблоном, в котором осуществляется представление текста:

```
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns: android=http://schemas. android.com/apk/res/android>
<TextView android: id="@+id/textViewId"
android: layout_width="fill_parent"
android: layout_height="match_content"
android: text="@string/hello"
/>
</LinearLayout>
```

Будет применяться ID, сгенерированный для данного XML-файла, чтобы загрузить этот шаблон в окно действия. В Android также широко поддерживаются меню, от стандартных до контекстных. Они также загружаются как XML-файлы и поскольку ID ресурсов для этих меню генерируются автоматически. Ниже показано, как объявлять меню в XML-файле:

```
<menu xmlns: android="http://schemas. android.com/apk/res/android">
<! - В этой группе используется категория, заданная по умолчанию. - ->
<group android: id="@+id/menuGroup_Main">
<item android: id="@+id/menu_clear"
android: orderInCategory="10"
android: title="clear" />
<item android: id="@+id/menu_show_browser"
android: orderInCategory="5"
android: title="show browser" />
</group>
</menu>
```

Хотя в Android и поддерживаются диалоговые окна, все они являются асинхронными. Работа с такими асинхронными диалоговыми окнами представляет особую сложность для разработчиков, привыкших к работе с

синхронными модальными диалоговыми окнами, которые используются в некоторых оконных фрейм-ворках

В Android также поддерживается анимация - эта функция входит в состав стека пользовательского интерфейса, который базируется на видах и отрисовываемых объектах. В Android используется анимация двух видов: с построением промежуточных кадров (tweening animation) и покадровая (frame-by-frame animation). Промежуточными в анимации называются такие рисунки, которые отображаются между основными. На компьютере эти рисунки создаются методом изменения средних значений через определенные промежутки времени и перерисовывания фона. Покадровая анимация состоит из серий кадров, которые прорисовываются один за другим через регулярные временные интервалы. В Android применяются оба варианта анимации, при этом используются функции обратного вызова анимации, интерполяторы и матрицы преобразований. Кроме того, в Android можно определять такие виды анимации в файле ресурсов XML. В следующем примере показаны серии пронумерованных изображений, которые воспроизводятся при покадровой анимации:

```
<animation-list xmlns: android="http://schemas.android.com/apk/res/android"
android: oneshot="false">
<item android: drawable="@drawable/numbers11"android: duration="50"/>
<item android: drawable="@drawable/numbers19"android: duration="50"/>
</animation-list>
```

Графические библиотеки, на которых основан такой процесс, поддерживают стандартные матрицы преобразований, позволяя масштабировать, перемещать и вращать рисунки. Объект Camera, присутствующий в графической библиотеке, обеспечивает поддержку глубины и проекции, благодаря чему на двухмерном интерфейсе удается имитировать трехмерные эффекты.

В Android поддерживается и трехмерная графика. Это происходит благодаря тому, что в систему внедрен стандарт OpenGL ES 1.0. Подобно OpenGL, он является плоским API на базе языка C. Поскольку интерфейс прикладного программирования Android SDK базируется на Java, для доступа к OpenGL в нем следует использовать Java-связывание. В Java ME такое связывание для OpenGL ES уже определено при помощи запроса на спецификацию Java QSR 239, и в Android для OpenGL ES используется такой же вид Java-связывания.

В Android используются новые подходы, связанные с идеей информации на кончиках пальцев (information at your fingertips), доступ к которой осуществляется через домашнюю страницу. Первая из таких идей называется живые каталоги (live folders). При помощи живых каталогов можно опубликовать коллекцию элементов в виде папки, расположенной на домашней

странице. Содержимое этой коллекции меняется по мере того, как изменяются лежащие в ее основе данные. Новые данные могут поступать на устройство либо со съемных носителей, либо из Интернета.

Вторая идея, связанная с домашней страницей, - это домашний виджет (home screen widget). Домашние виджеты используются для отрисовывания информации на домашней странице с применением виджета пользовательского интерфейса. Данная информация может изменяться через регулярные временные интервалы. Примером такой информации может быть несколько электронных сообщений, которые сохранены на устройстве.

Интегрированный поиск Android (Integrated Android Search) - это третья идея, связанная с использованием домашней страницы. Такой вид поиска позволяет искать информацию как на самом устройстве, так и в Интернете. Данная функция Android не ограничивается одним только поиском и позволяет давать команды при помощи элемента управления поиском.

Кроме того, в Android поддерживаются так называемые жесты, то есть интерпретация движений пальцев пользователя, работающего с устройством. Android позволяет записывать любые последовательности движений пальцев по экрану и сохранять их как жесты. Затем такие жесты могут использоваться приложениями для обозначения конкретных действий.

Кроме инструментария Android SDK, существуют другие самостоятельные инновации, которые делают процесс разработки интересным и несложным. Некоторые примеры таких явлений - XML/VM, PhoneGap и Titanium. Titanium позволяет использовать технологии HTML при программировании для основанного на WebKit браузера Android.

Другим интересным компонентом Android SDK являются сервисы, основанные на местоположении. Данный раздел SDK предоставляет разработчикам приложения интерфейсы прикладного программирования, предназначенные для работы с картами, а также для получения в реальном времени информации, связанной с местоположением устройства.

Медийные компоненты и компоненты связанные с телефонией

В Android имеются интерфейсы API, предназначенные для работы с аудио-, видео - и телефонными компонентами. Ниже приведен пример того, как воспроизвести аудиофайл из Интернета по URL:

```
private void playAudio (String url) throws Exception
```

```
{  
mediaPlayer = new MediaPlayer ();  
mediaPlayer. setDataSource (internetUrl);  
mediaPlayer. prepare ();  
mediaPlayer. start ();  
}
```

А вот как аудиофайл воспроизводится с локального устройства:

```

{
// файл расположен в каталоге /res/raw и называется "music_file. mp3"
mediaPlayer = MediaPlayer. create (this, R. raw. music_file);
mediaPlayer. start ();
}

```

Ниже показан пример кода для отправки SMS-сообщения:

```

private void sendSmsMessage (String address,String message) throws Exception
{
SmsManager smsMgr = SmsManager. getDefault();
smsMgr. sendTextMessage (address, null, message, null, null);
}

```

До версии 1.5 в Android можно было записывать только аудио, а видео - нет. В версии 1.5 была обеспечена возможность записи как аудио, так и видео. Это было сделано при помощи MediaRecorder. В версиях Android 2.0 и выше применяется движок Pico для преобразования текста в речь (синтеза речи). Интерфейс, используемый в Android для преобразования текста в речь, очень прост, как и соответствующий код:

```

TextToSpeech mTTS;
mTTS. speak (somestring, TextToSpeech. QUEUE_ADD);
mTTS. setOnUtteranceCompletedListener (this);
mTTS. stop ();
mTTS. shutdown ();
mTTS. synthesizeToFile (...);

```

Еще несколько методов, относящихся к этой сфере:

```

playSilence
setLanguage
setPitch
setSpeechRate
isSpeaking

```

Важно отметить, что в Android все эти концепции объединены в приложение путем создания единого XML-файла, в котором определяется, каким будет пакет прикладных программ. Этот файл называется файлом описания (manifest file) (AndroidManifest.xml). Пример такого файла:

```

<? xml version="1.0" encoding="utf-8"? >
<manifest xmlns: android=http://schemas. android.com/apk/res/android
package="com. ai. android. HelloWorld">
android: versionCode="1"

```

```
android: versionName="1.0.0">
<application android: icon="@drawable/icon"
android: label="@string/app_name">
<activity android: name="HelloWorld"
android: label="@string/app_name">
<intent-filter>
<action android: name="android.intent.action.MAIN"
<category
android: name="android.intent.category.LAUNCHER"
</intent-filter>
</activity>
</application>
</manifest>
```

В файле описания Android содержатся определения действий, регистрируются поставщики содержимого и поставщики служб и обозначаются права доступа. Мы будем возвращаться к файлу описания на протяжении книги по мере развития тех или иных идей.

2 Инструменты разработки

2.1 Инструменты Android-разработчика

Естественно переход к практике программирования под любую платформу начинается с просмотра имеющегося в наличии инструментов разработки. Только после того, как инструменты мы выбрали (руководствуясь какими-то соображениями и предпочтениями), полностью включены в работу для того, чтобы опробовать в написании некоторых простых инструкций.

Разработка приложений для платформы Android ведется преимущественно на языке Java. Для создания программ на языке Java необходимо специальное программное обеспечение. Самые последние версии этого ПО можно загрузить с официального сайта разработчика, Oracle Corporation. К этому программному комплексу относятся такие инструменты как JRE (Java Runtime Environment) и JDK (Java Development Kit). Первый инструмент представляет собой среду выполнения – минимальную реализацию виртуальной машины, в которой запускается и выполняется программный код на Java. Второй инструмент – это в свою очередь целый набор инструментов, комплект разработчика приложений на языке Java. На

самом деле, JRE также входит в состав JDK, равно как и различные стандартные библиотеки классов Java, компилятор `javac`, документация, примеры кода и разнообразные служебные утилиты. Весь этот набор распространяется свободно и имеет версии для различных ОС, поэтому любой может его скачать и использовать. В JDK не входит интегрированная среда разработки, предполагается, что её разработчик будет устанавливать отдельно. Существуют многочисленные IDE для Java-разработки, например, NetBeans, IntelliJ IDEA, Borland JBuilder и другие. Таким образом, прежде чем приступить к разработке приложения на базе ОС Android, необходимо подготовить инструментарий. При разработке на базе ОС Android необходимо использовать среду Eclipse IDE for Java Developers. На сайте компании Google можно найти и скачать SDK для своей платформы (см. рисунок 1).

Choose the SDK package for your OS from the table below.

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

If you prefer to use an existing version of Eclipse or another IDE, you can instead take a more customized approach to installing the Android SDK. See the following instructions.

- ▼ [USE AN EXISTING IDE](#)
- ▼ [SYSTEM REQUIREMENTS](#)
- ▼ [DOWNLOAD FOR OTHER PLATFORMS](#)

ADT Bundle

Platform	Package	Size	MD5 Checksum
Windows 32-bit	adt-bundle-windows-x86.zip	425429957 bytes	cca97f12904774385a57d542e70a490f
Windows 64-bit	adt-bundle-windows-x86_64.zip	425553759 bytes	c51679f4517e1c3ddefa1e662bbf17f6
Mac OS X 64-bit	adt-bundle-mac-x86_64.zip	390649300 bytes	f557bc61a4bff466633037839771bffb
Linux 32-bit	adt-bundle-linux-x86.zip	418614971 bytes	24506708af221a887326c2a9ca9625dc
Linux 64-bit	adt-bundle-linux-x86_64.zip	418889835 bytes	464c1fbe92ea293d6b2292c27af5066a

SDK Tools Only

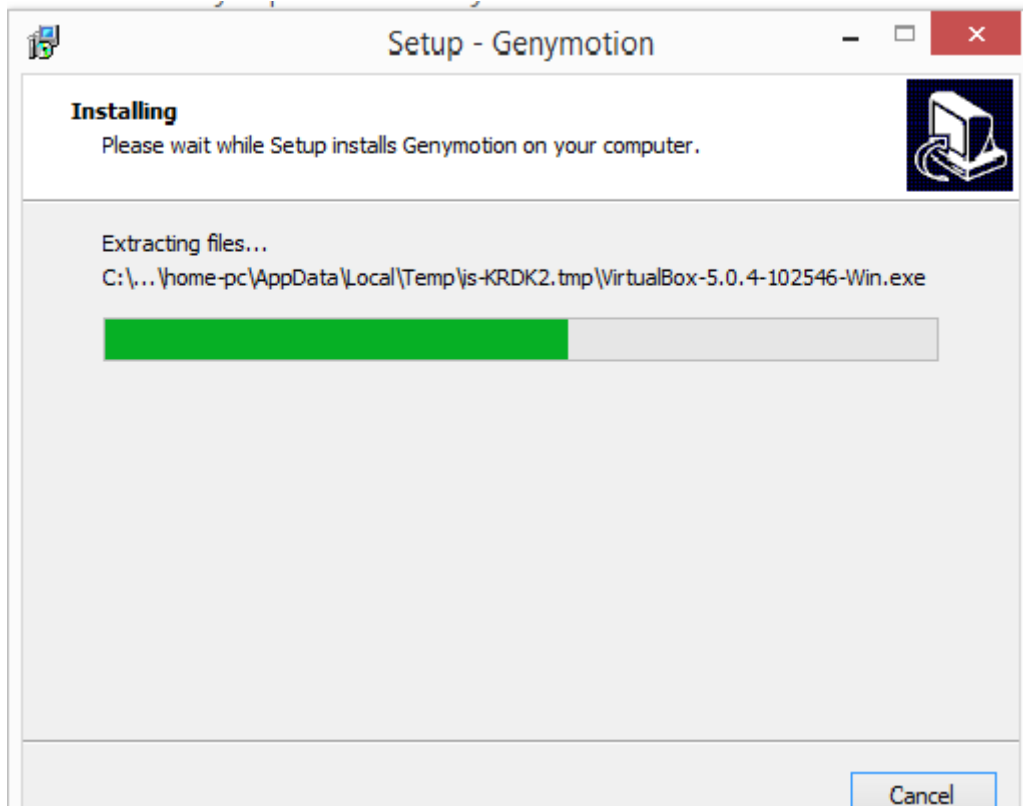
Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r21.1-windows.zip	99360755 bytes	dbece8859da9b66a1e8e7cd47b1e647e
	installer_r21.1-windows.exe (Recommended)	77767013 bytes	594d8ff8e349db9e783a5f2229561353
Mac OS X	android-sdk_r21.1-macosx.zip	66077080 bytes	49903cf79e1f8e3fde54a95bd3666385
Linux	android-sdk_r21.1-linux.tgz	91617112 bytes	3369a439240cf3dbe165d6b4173900a8

Рис. 1. Android SDK

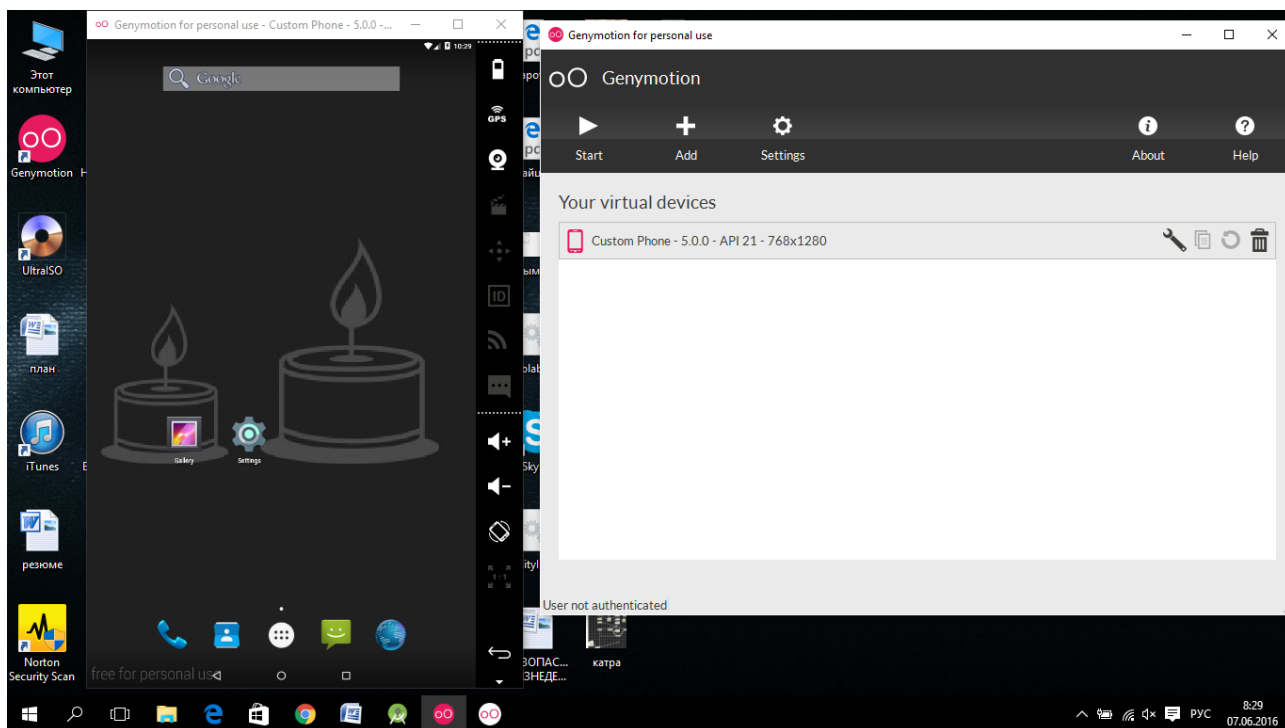
2.2. Эмулятор Android

Магазин приложений для Android насчитывает несколько сотен тысяч различных программных продуктов, как платных, так и бесплатных. Такое обилие приложений делает данную операционную систему, наряду с iOS, одной из самых популярных и востребованных мобильных операционных систем в мире.

Совершенно логично, что при такой, не побоюсь этого слова, огромной популярности появляются различные эмуляторы, позволяющие открывать приложения разработанные для Android на обычной платформе.



Установка эмулятора Genymotion



Вид эмулятора на экране

Данный эмулятор отлично подойдет для тестирования ваших приложений созданных для операционной системы Android, все это доступно благодаря встроенному аппаратному ускорению OpenGL. В прошлом данный проект (эмулятор) носил название Android VM, постепенно он вырос в полноценный, и на мой взгляд, один из самых качественных продуктов подобного рода из всех доступных.

Программный инструмент Genymotion является мультиплатформенным эмулятором, это значит, что он доступен для большинства операционных систем, таких как: Windows, MacOS, Linux. Из большого количества достоинств и возможностей данного программного инструмента, хочется отметить следующие:

- Довольно внушительная база поддерживаемых, для эмуляции, устройств. При этом база постоянно пополняется, что, конечно, не может не радовать;
- Позволяет эмулировать соединение по сети Wi-Fi;
- Эмуляция задней и передней видеокамеры на устройстве;
- Возможность создавать скринкасты.

Это, конечно, не полный список всех достоинств данного инструмента. Некоторые возможности я специально не стал затрагивать т.к. они вряд ли будут понятны обычному, рядовому пользователю PC.

3. Разработка базы знаний

3.1 Подготовка к работе

Для разработки приложений для платформы Android нам нужно установить на компьютер Android SDK. Кроме остальных инструментов разработчика в него входит эмулятор нашей мобильной операционной системы, так что для программирования и отладки даже необязательно иметь под рукой устройство на базе Android. SDK имеется в свободном доступе, и скачать его можно с официального сайта — <http://developer.android.com/sdk/index.html>. Отсюда нужно загрузить версию SDK для своей операционной системы.

К примеру, для Windows даются для выбора версии в виде архива zip и исполняемого файла exe. И в том, и в другом случае Android SDK представляет собой просто сжатую папку с файлами, содержимое которой будет распаковано в удобное нам место. Если вдруг во время распаковки архива (в случае с exe-файлом) появляются уведомления об отсутствии в системе установленного JDK (а он при этом установлен), то достаточно нажать кнопку Back (Назад) и снова попробовать, тогда всё станет корректно.

После проделанных действий мы пойдём в папку с распакованным контентом и прочитаем “SDK Readme.txt”, в котором говорится о том, что данный архив содержит в себе только базовые инструментальные средства. Для полноценной разработки нам предстоит воспользоваться утилитой “SDK Manager”, которая позволяет устанавливать и модифицировать компоненты SDK, пользуясь репозиторием на сервере Google (да-да, понадобится соединение с Интернет).

Запустим исполняемый файл с названием SDK Manager. Здесь может возникнуть проблема с его работой. Если выдаются ошибки и приложение не запускается, то тут дело в переменной среды PATH, которая хранит в себе пути поиска для исполняемых файлов и используется различными программами и скриптами. Для корректной работы в этой переменной должен быть указан путь до каталога инструментов Android SDK, а также до каталога bin для JDK (в дальнейшем он нам понадобится).

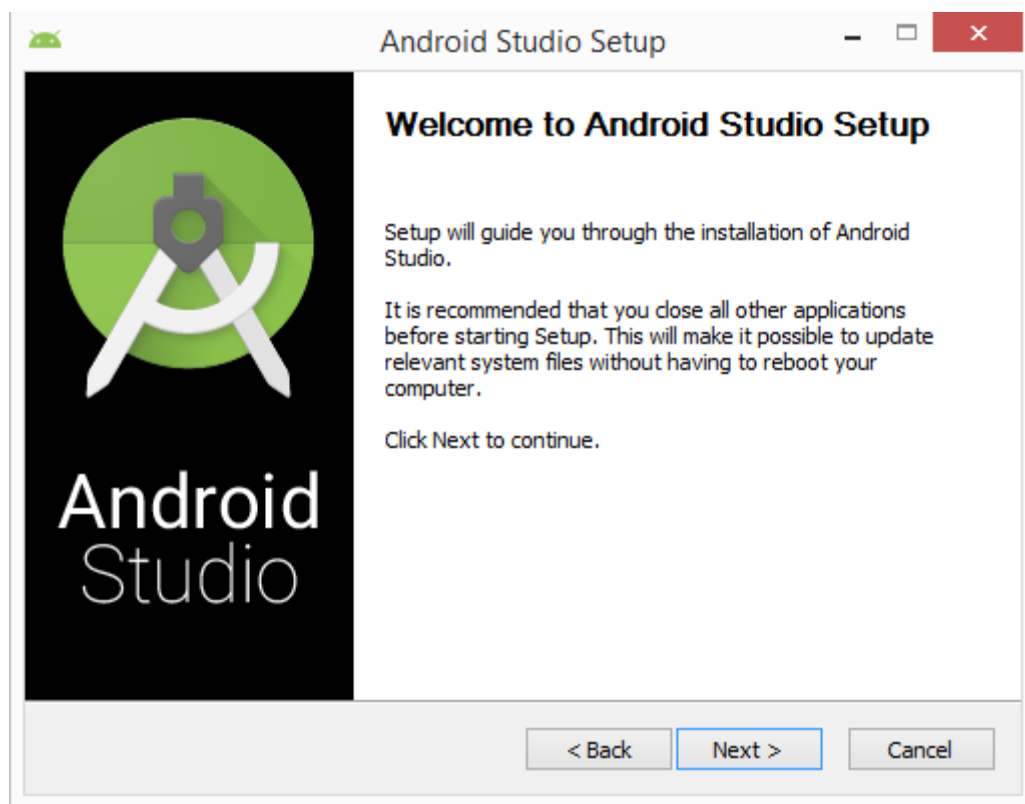





Рисунок 3.1 – Установка Android Studio

Следующий пункт – это регистрация на портале GitHub, основная часть изображена на рисунке 3.2. Данное действие необходимо для хранения бэкапов проекта на сервере портала, также на форуме данного портала можно найти множество подсказок и полезных статей по проектированию различных работ для мобильного приложения.

Join GitHub

The best way to design, build, and ship software.

 Step 1: Set up a personal account	 Step 2: Choose your plan	 Step 3: Go to your dashboard
-------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------

Create your personal account

Username

This will be your username — you can enter your organization's username next.

Email Address

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

[Create an account](#)

You'll love GitHub

- Unlimited** collaborators
- Unlimited** public repositories
- ✓ Great communication
- ✓ Friction-less development
- ✓ Open source community

Рисунок 3.2 – Регистрация в GitHub

После регистрации появляется возможность связать свою работу в Android Studio с компьютера разработчика с аккаунтов на GitHub, и проект будет храниться в хранилище, как показано на рисунке 3.3. В нужный момент разработчик может получить доступ к своему проекту на сервере, даже если у него отсутствует личный ПК, разработчик может зайти на аккаунт портала и скачать собственный проект на иной переносной либо персональный компьютер, необходимо лишь наличие Android Studio соответствующей или новейшей версии и доступа к сети интернет.

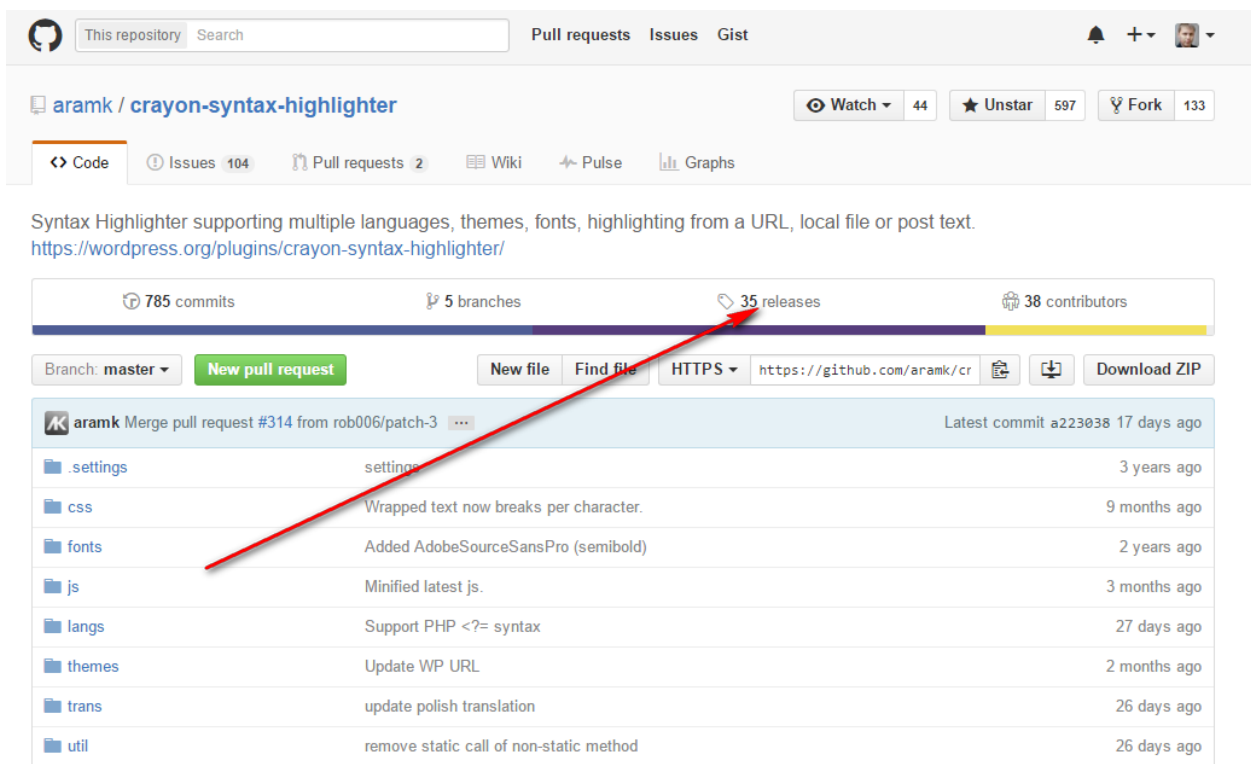


Рисунок 3.3 – Хранилище проектов на GitHub

3.2 Проектирование базы знаний

Начало работы в Android Studio, создаем новый проект, даем название и на экране выводится главное окно программы, которое включает в себя панель управления, панель инструментов, панель отладки, панель структуры приложения и эмулятор смартфона по центру экрана. Все вышеперечисленное отображено на рисунке 3.4.

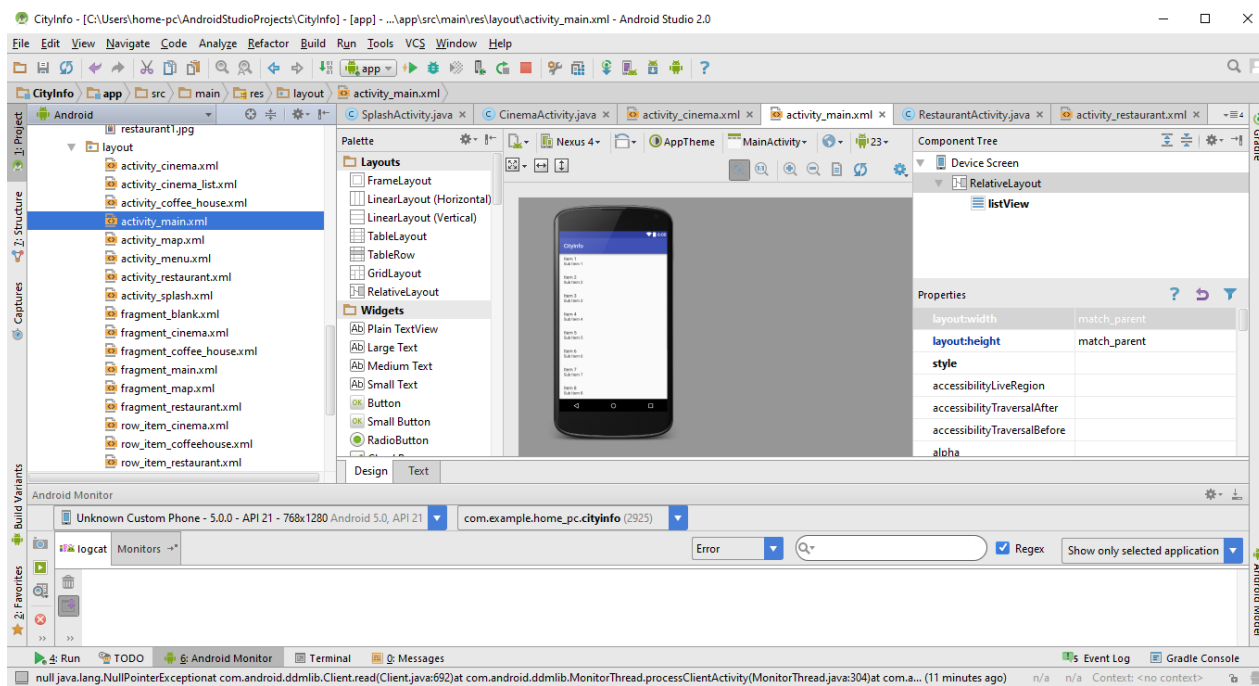


Рисунок 3.4 – Основное меню работы в Android Studio

Поскольку основными элементами приложения для ОС Android являются активности, то схема работы реализуемого приложения представляет собой схему связей между активностями. На рисунке 4 показана диаграмма активностей UML, демонстрирующая работу всего разрабатываемого приложения.

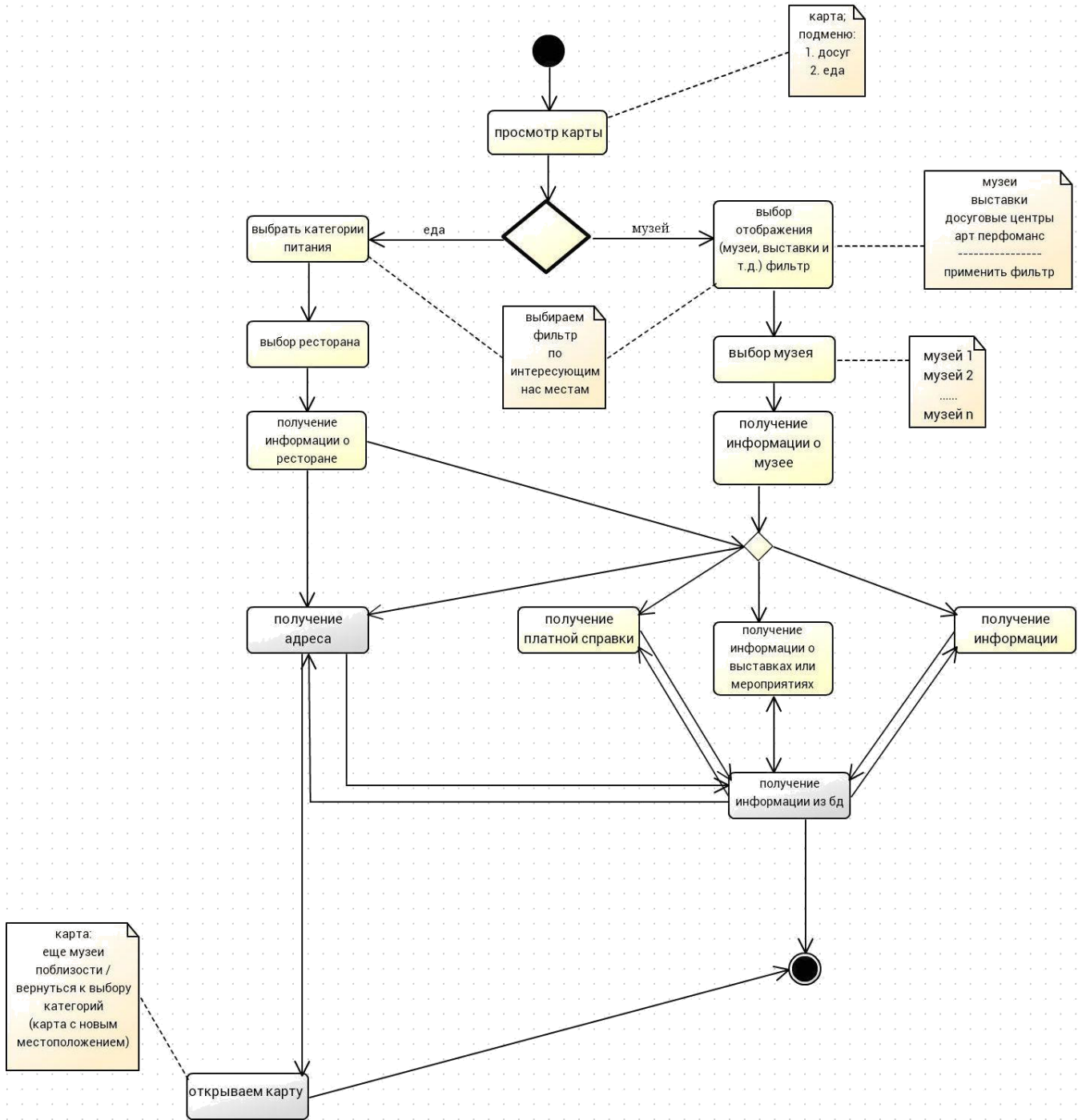
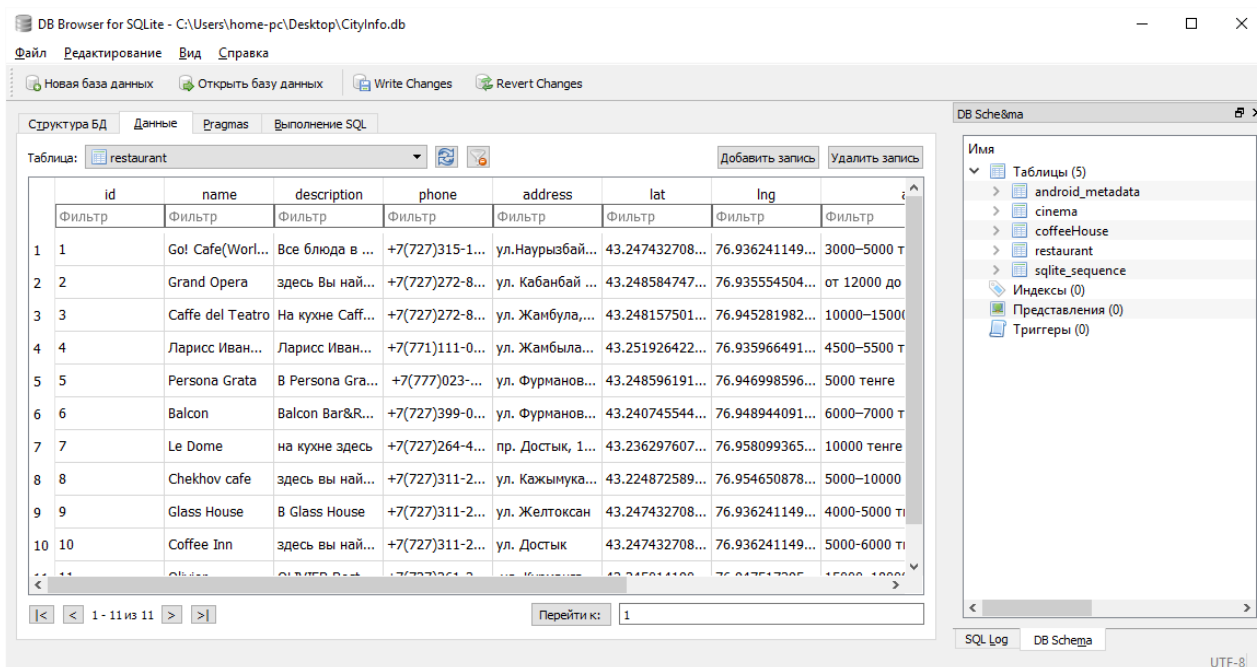


Рис. 4. Пошаговая схема работы всего приложения



База данных

3.3 Результаты работы приложения

По итогам работы получено мобильное приложение CityInfo – приложение на платформе Андроид, основной целью которого является удобство поиска ресторанов и кинотеатров на карте, а также их полной информации.

Главное меню включает в себя три разделов, которые доступны пользователю.

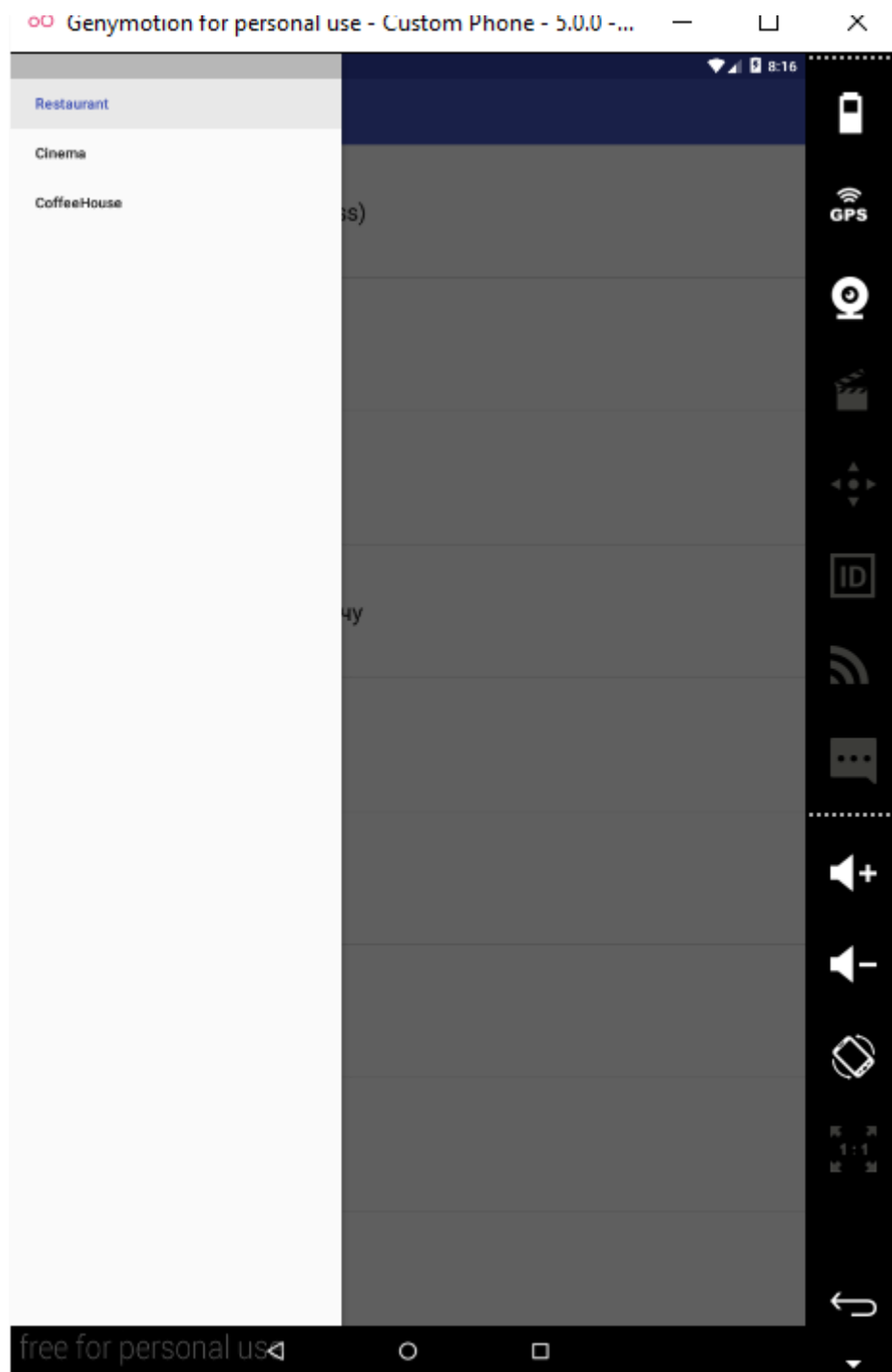


Рисунок 3.8 – Главное меню

По умолчанию сразу открывается активити со списком ресторанов(рис 3.9)

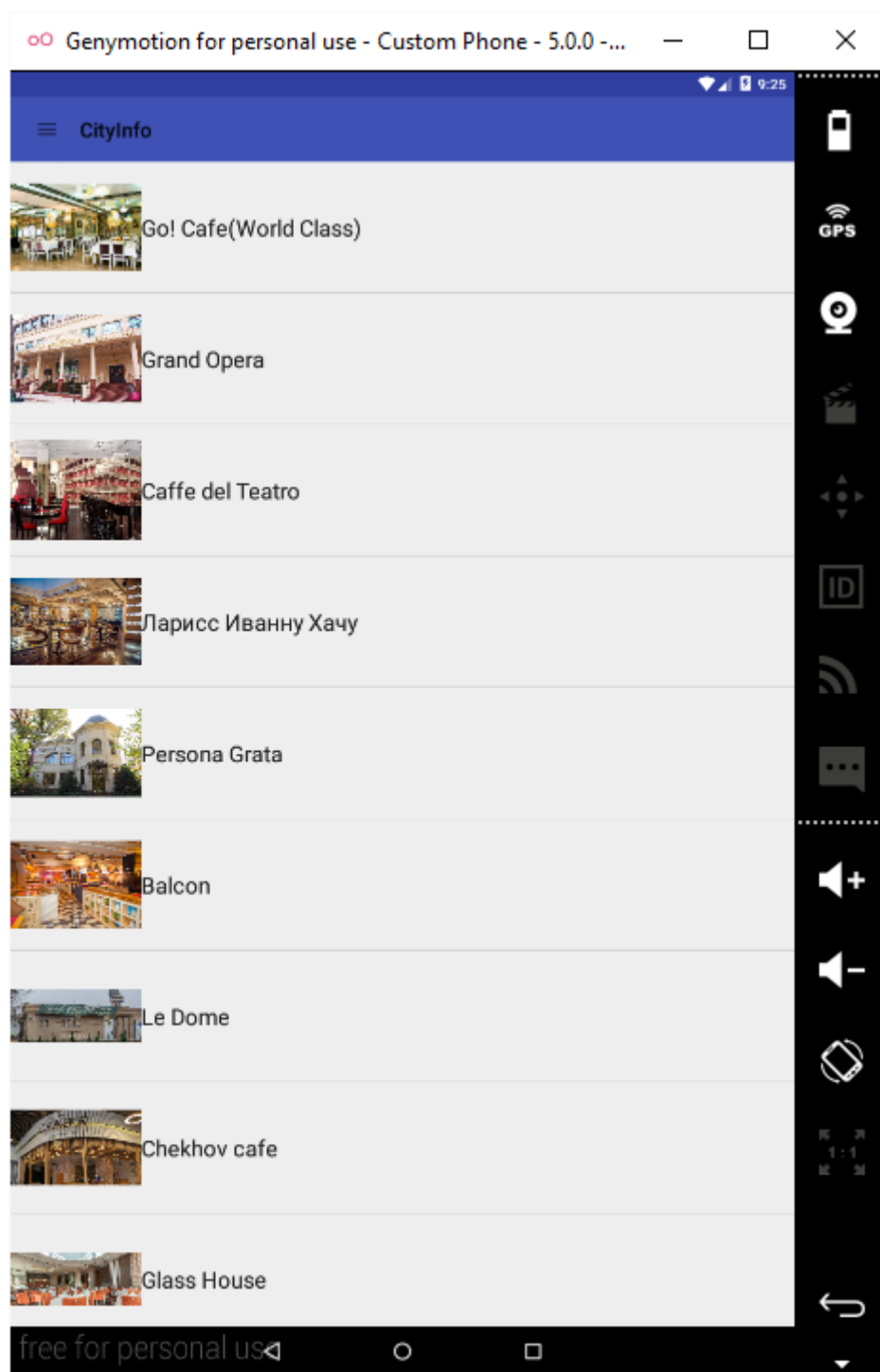


Рисунок 3.9

Далее, после выбора нужного ресторана пользователь может перейти на определенный ресторан из списка. Переход осуществляется мгновенно и пользователю сразу доступно описание, на рисунке 3.10.

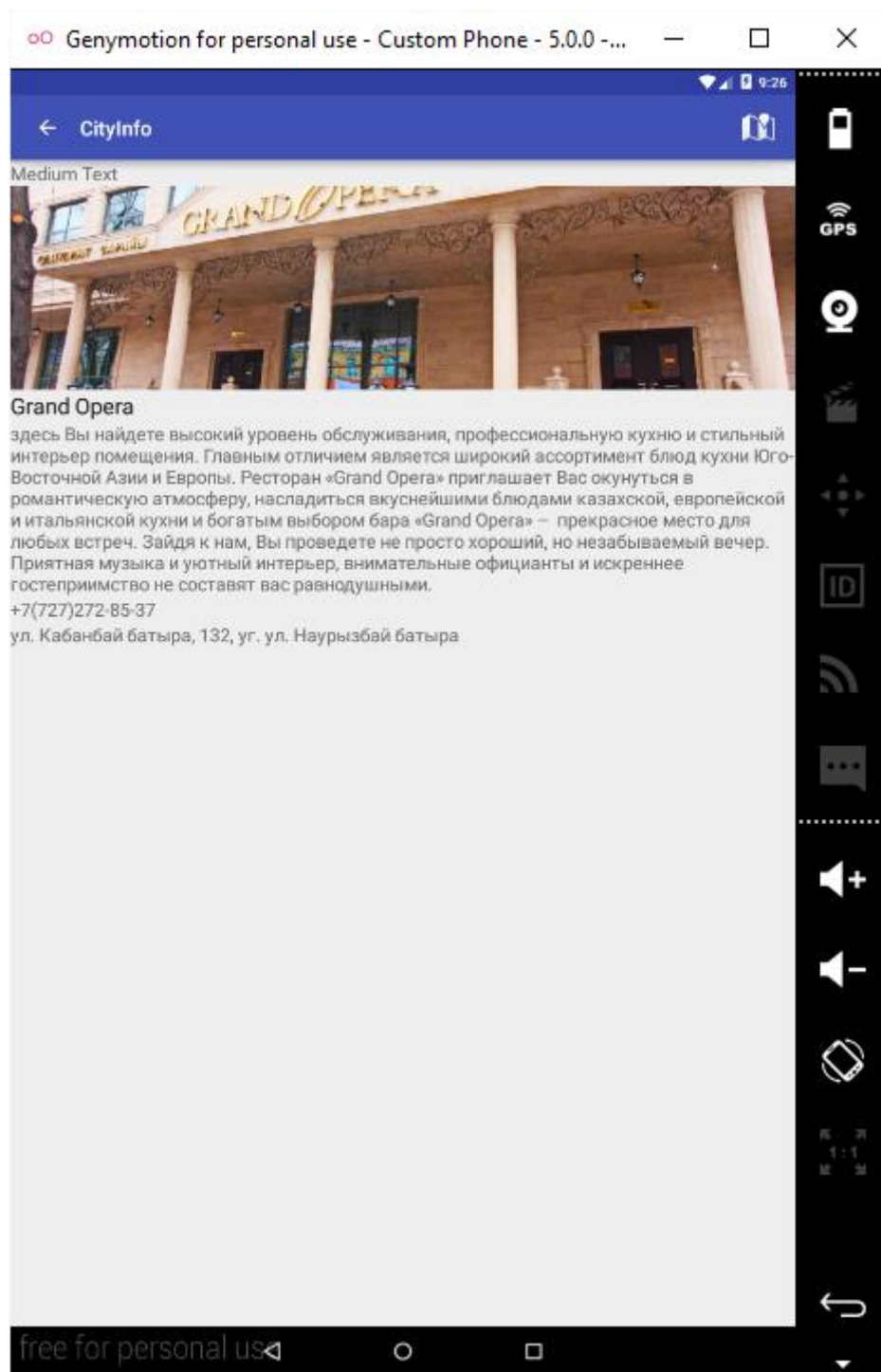
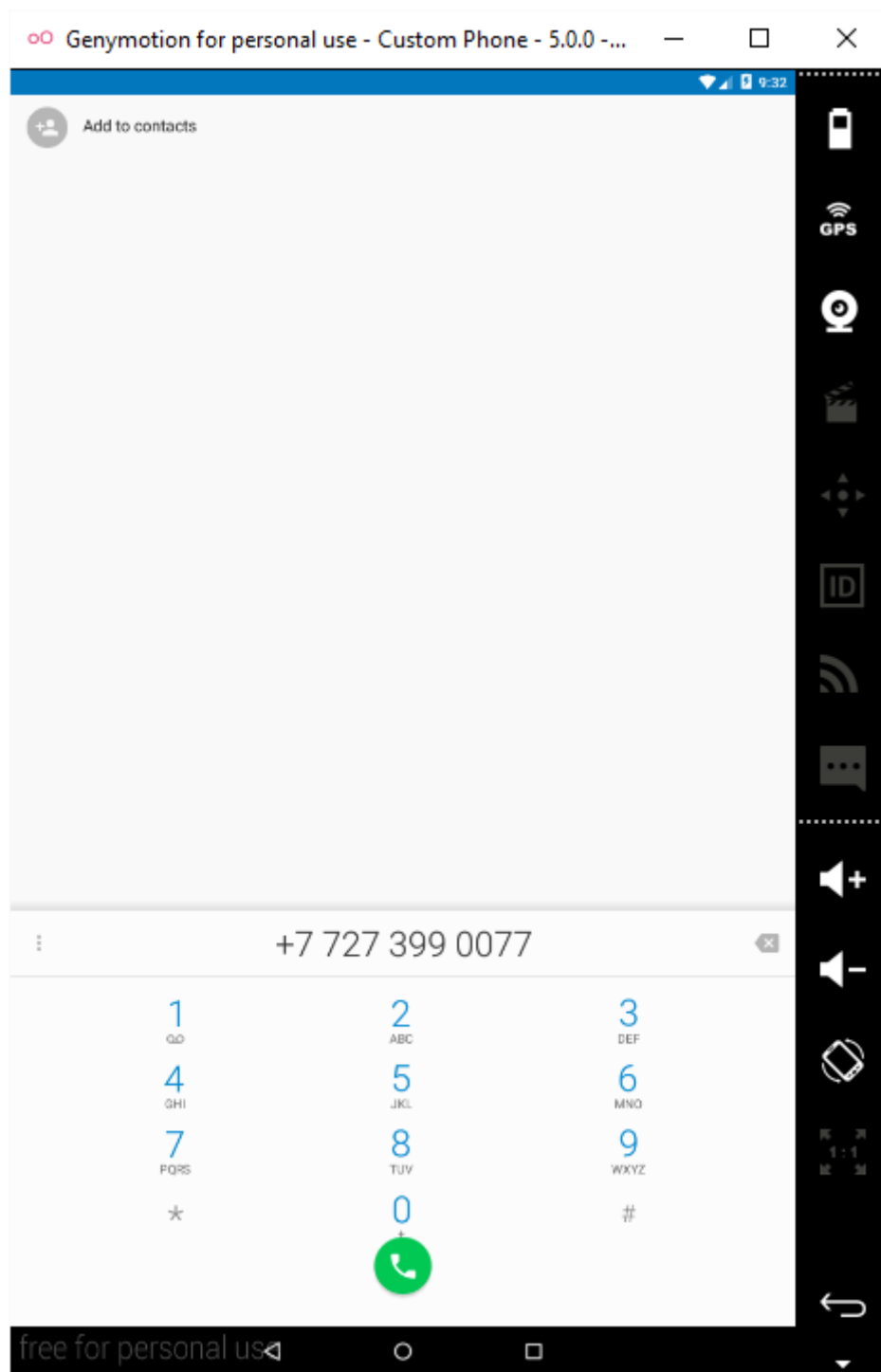
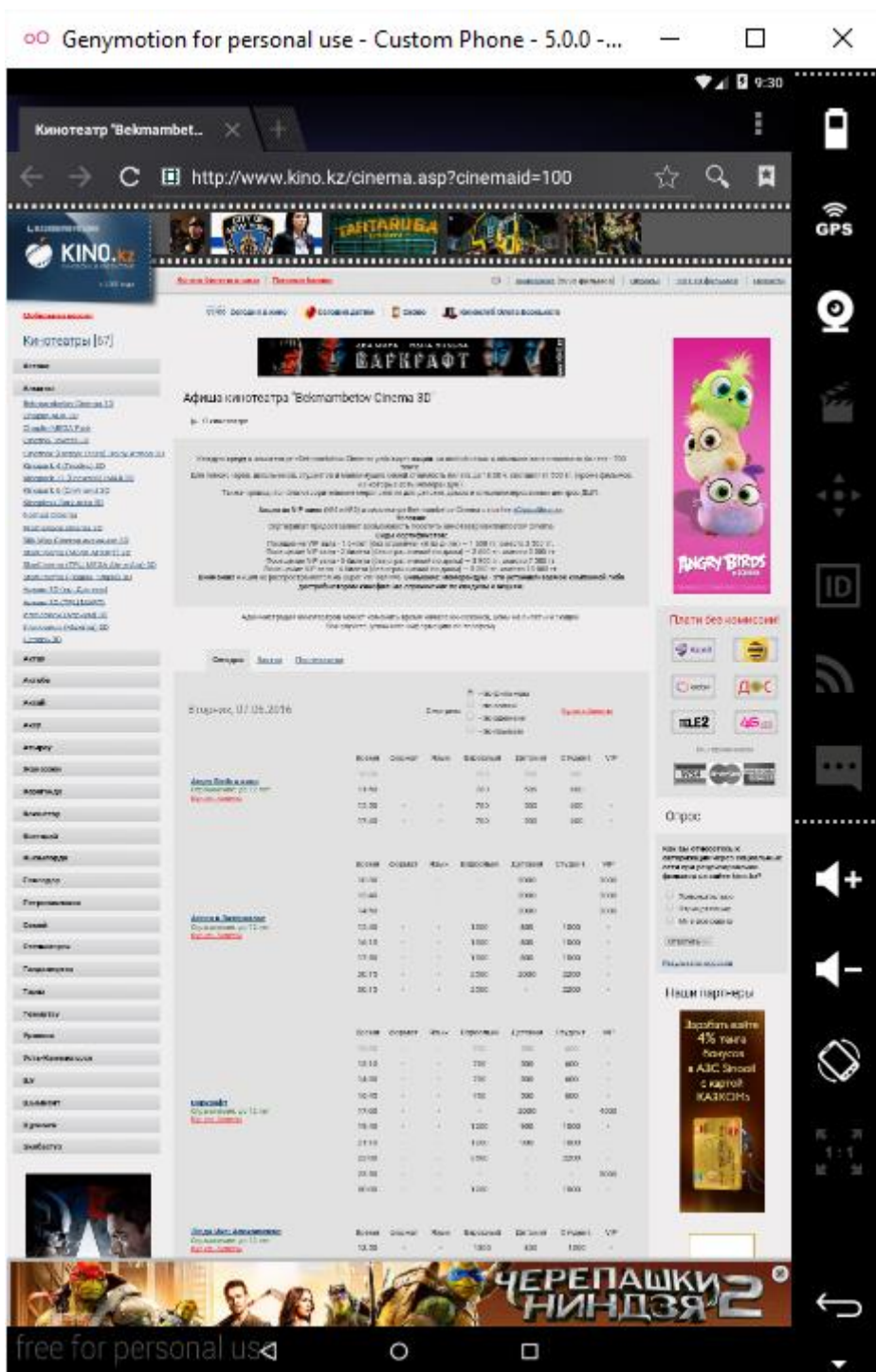


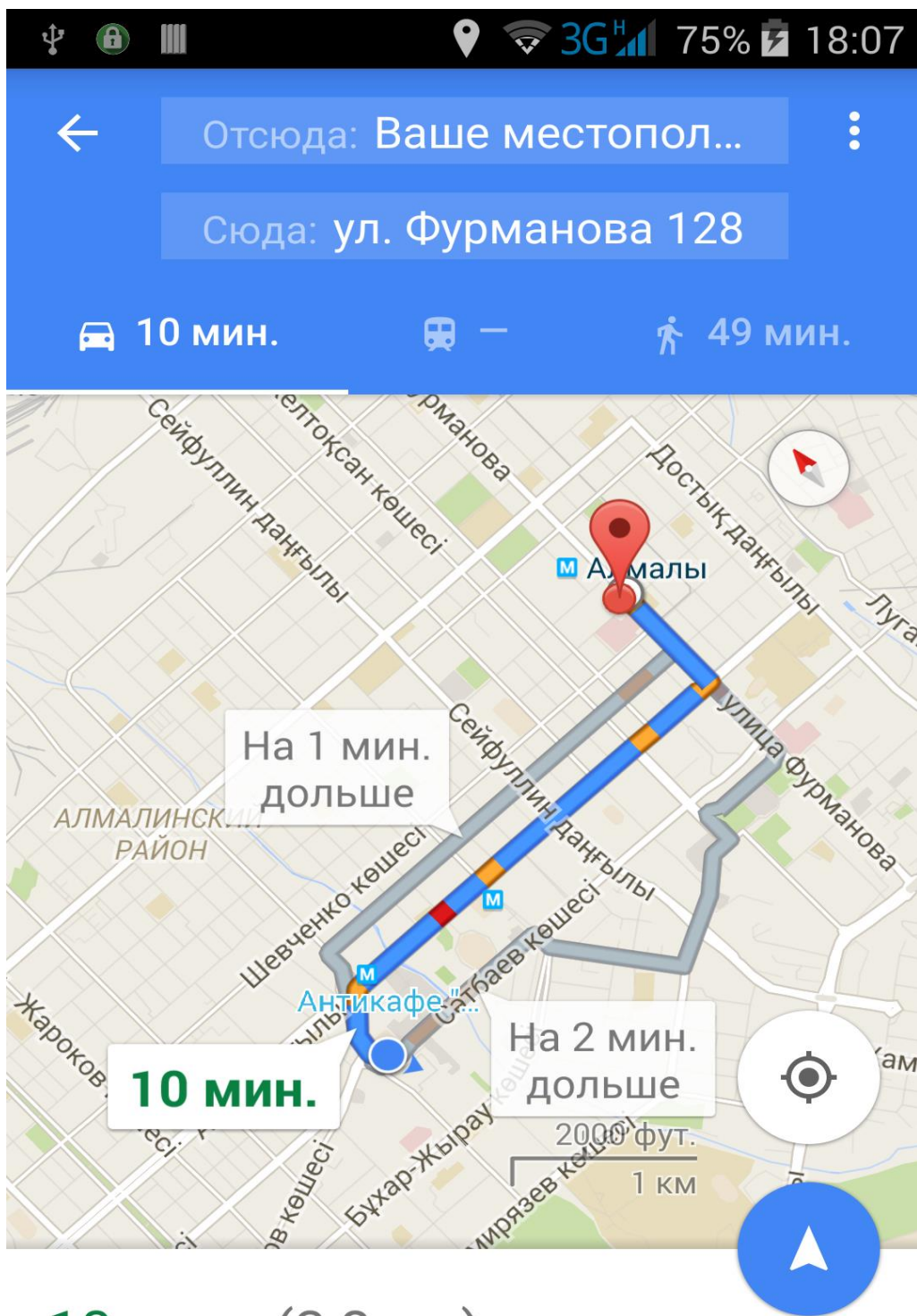
Рисунок 3.10



Звонок в ресторан



Пройдя по ссылке, мы можем посмотреть расписание кинотеатров.



10 мин. (3,8 км)

Самый быстрый маршрут,
пробки как обычно

При нажатии на кнопку Маршрут, на карте показывается ближайший маршрут и время за которое можно дойти до него.

3.2 Тестирование

В процессе разработки приложения производилось поэтапное тестирование с целью выявления программных ошибок и несоответствий ТЗ (техническому заданию). Для этого нами были созданы эмуляторы смартфона и планшета с разными диагоналями экрана для разных версий Android.

Тестируемый программный продукт последовательно запускался на этих эмуляторах, его поведение анализировалось, и при необходимости по результатам анализа вносились изменения в код.

Для тестирования отдельных модулей работы с базой данных в текст программы были внесены специальные функции, позволяющие анализировать базу данных и, при подозрении на ошибку, выводящие сообщение в системный журнал. Они также известны как юнит-тесты. Например, при изменениях в базе данных проводилась проверка целостности базы данных (проверка на соответствие ключей – индексам), после чего при необходимости выводилось сообщение в системный log.

Были проведены приведенные ниже тесты.

1. Каждая активность была подвергнута юнит-тестированию с целью выявления ошибок, вызванных несоответствием ожидаемых и полученных параметров. Для этого для каждой активности был создан специальный юнит-класс, посылающий в активность различные верные и неверные параметры. При аномальном поведении активности или ее сбое, мною анализировалось поведение и ошибка исправлялась.
2. В базу данных намеренно вносились недопустимые данные в соответствующие поля, которые могли быть неверно интерпретированы программой. Затем мною анализировалось поведение активности во время обработки недопустимых данных.

3. Приложение было запущено на устройствах, работающих под управлением разных версий Android с целью выявления особенностей работы приложения, запущенного в разных операционных системах.
4. После завершения цикла разработки, программный продукт тестировался на реальных устройствах. По результатам тестирования была добавлена виртуальная кнопка «Меню» для устройств, не имеющих аппаратных кнопок.

3.3 Возможные варианты развития программного приложения

Возможными путями развития приложения являются:

1. интеграция приложения со сторонними базами данных;
2. реализация двухкомпонентной базы данных (локальной базы данных и базы данных в облаке);
3. реализация возможности подгрузки пакетной информации;
4. обновление мест интереса с использованием удаленного сервера;
5. реализация интерфейса на различных языках.

4 Техничко-экономическая часть

4.1 Определение задач и объяснение эффективности

Тема данного дипломного проекта – Разработка базы знаний для мобильного приложения «Прогнозирование спортивных событий».

В наше время резко повысилась популярность букмекерских контор. Люди пользуются их услугами и рассчитывают получить прибыль, основываясь на предположениях и догадках о некотором спортивном событии. Данный проект поможет людям в достижении успеха, предоставляя всю возможную информацию и статистику. Экспертная система, основываясь на своей базе знаний, помогает человеку рассчитать риски и выбрать оптимальную ставку на спортивное событие.

4.2 Расчет окончательных материальных затрат на все компоненты программного продукта

Расчет итоговых затрат [17] на разработку компонентов дипломного проекта в виде информационных технологий (C_{ni}) находится по формуле:

$$C_{ni} = Z_{фот} + Z_{сзи} + M_i + P_{ci} + P_{mi} + P_{zi} + P_{ni} \quad (4.1)$$

где $Z_{фот}$ – полный объем оплаты работы разработчиков, тенге;
 $Z_{сзи}$ – платеж по социальному налогу, тенге;
 M_i – издержки на сырье и материалы, тенге;
 P_{ci} – издержки на нужные программные компоненты, нужные для разработки проектного решения, тенге;
 P_{mi} – платежи, основанные на пользовании машинной аппаратурой, тенге;
 P_{zi} – побочные затраты, тенге;
 P_{ni} – накладные расходы, затраты на содержание средств тенге.

Размер полного объема награждения работы разработчиков ($Z_{фот}$) находится по формуле:

$$Z_{фот} = Z_{oi} + Z_{di} \quad (4.2)$$

где Z_{oi} – основная заработная плата, тенге;
 Z_{di} – дополнительная заработная плата, тенге.

Затраты на оплату труда находятся от объема и трудоемкости создания программного обеспечения. Общее количество (V_0) программной части

дипломной работы находим на вычислениях массы и размера функций, реализуемых программой:

$$V_0 = \sum_{i=1}^n V_i \quad (4.3)$$

где V_i - объем отдельной функции ПО;

n – общее число функций.

Из приложения устанавливаем объем ПО (строки исходного кода, LOC) составляют 4660. Таким образом:

$$V_0 = 4660 \text{ строк кода}$$

Общая трудоемкость проекта находится по формуле:

$$T_0 = T_n \cdot K_c \cdot K_T \cdot K_n \quad (4.4)$$

где K_c – коэффициент, учитывающий трудность создания ПО;

K_T – коэффициент поправки, который берет в расчет степень эксплуатации при создании условных модулей;

K_n – коэффициент, берущий в расчет новизну ПО;

T_n – нормированная трудоемкость.

Коэффициент трудности вычисляется относительно данных, показанных в таблице 4.1 и имеет значение $K_c = 0,18$, потому что в проекте могут быть более 3-х характеристических параметров – программный функционал, интерактивное взаимодействие, предоставление недостижимости, управления и поиска данных в многоуровневых структурах.

Таблица 4.1 – Побочные значения трудности создания ПО

Характеристика ПО	Значения K_c
1 Работоспособность ПО в общей операционной сфере (связь с другими ПО)	0,09
2 Интерактивный допуск	0,05
3 Гарантия сохранности, ведения и розыска инфы в многосложных конструкциях	0,08
4 Содержание в ПО в одно время нескольких параметров по табл.Г4.1, приложение Г	
4.1 2 характеристики	0,11
4.2 3 характеристики	0,19
4.3 Свыше 3-х характеристик	0,27

Коэффициент поправки, принимающий во внимание частоту использования при работе над проектом дефолтных модулей (K_T),

высчитывается на основании данных, доступных в таблице 4.2 и имеет значение 0,7.

Таблица 4.2 – Показания уточняющего значения

Размер охвата используемых функций создаваемого ПО классическими модулями, типовыми продуктами и ПО	Значения K_T
1 От 62 % и выше	0,5
2 От 42 % до 61	0,8
3 От 18 % до 41 %	0,7
4 До 17 %	0,9
5 Стандартное ПО не применяемое для проектировки функционала создаваемого ПО	1,1

Примерное значение, берущий в расчет свежесть создаваемого продукта (K_H) находится на основе данных, представленных в таблице 4.3 и составляет 0,8.

Таблица 4.3 – Поправочные коэффициенты

Классификация свежести	Коэффициент свежести	Использование		Значение K_H
		На основе свежего типа ПК	В среде специфичной ОС	
А	Полностью новые ПО, у которых нет аналогов	+	+	1,86
		-	+	1,7
		+	-	1,1
		-	-	1,2
Б	ПО, которые считаются развитием конкретного характерного ряда ПО	+	+	1,2
		-	-	0,8
		+	-	0,7
В	ПО, считающиеся развитием конкретного характерного ряда ПО, созданных для заранее изученных видов конфигурации ПК и ОС	-	-	0,9

Базой для нахождения нормативной трудоемкости считаются, укрупненные нормы времени на разработку проекта в зависимости от уточненного объема ПО и группы сложности.

Нормативная трудоемкость проекта (T_n) определяется на основе принятого в расчет объема ПП и классификации трудности, которая уточняется с учетом сложности и новизны проекта и степени использования стандартных модулей при разработке.

Основываясь на информации из стандарта: для верхнего класса трудности ПО:

$$T_n = 162$$

Целый размер трудоемкости высчитывается по формуле (4.4)

$$T_o = 162 \cdot 0,18 \cdot 0,7 \cdot 0,9 = 19 \text{ чел./час.}$$

Принимая во внимание сложность работы, устанавливаются желательное количество проектировщиков ($Ч_p$) и желательные границы, выделенные для разработки проекта в итоге (T_p). Кроме этого поставляются определенные задания:

- подсчет числа разработчиков при установленных рамках выполнения работы;
- предположение срока проектирования при заданном числе разработчиков.

Количество разработчиков проекта ($Ч_p$) считается по формуле:

$$Ч = T_o / (T_p \cdot \Phi_{эф}) \quad (4.5)$$

где $\Phi_{эф}$ – значимый объем срока деятельности одного проектировщика на протяжении года (дн.);

T_o – целая трудоемкость разработки проекта (чел./дн.);

T_p – временные рамки создания проекта.

Длительность разработки продукта (T_p) находится по методу:

$$T_p = T_o / (Ч_p \cdot \Phi_{эф}) \quad (4.6)$$

где $Ч_p$ – рекомендуемое количество проектировщиков.

Эффективный фонд времени работы одного работника ($\Phi_{эф}$) вычисляется по формуле:

$$\Phi_{эф} = D_r - D_{п} - D_{в} - D_o \quad (4.7)$$

где D_r – количество дней в году;
 D_p – количество праздничных дней в году;
 D_b – количество выходных дней в году;
 D_o – количество дней отпуска.

В соответствии с общепринятым календарем на 2016 год:

- $D_r = 366$;
- $D_p = 16$;
- $D_b = 113$;
- $D_o = 15$, эффективный фонд времени одного работника составит:

$$\Phi_{эв} = 366 - 16 - 113 - 15 = 222 \text{ дня}$$

Установленное число разработчиков $Ч_p = 1$, следовательно, по формуле (4.6)

$$T_p = 18 / (1 \cdot 222) = 0,08 \text{ лет} = 30 \text{ дней}$$

Значит, учитывая вышепредоставленные вычисления и в соответствии с формулой (4.5)

$$Ч = 18 / (0,08 \cdot 225) = 1 \text{ чел}$$

Основная заработная плата разработчиков на конкретизированное ПО вычисляется по формуле:

$$З_{oi} = \sum_{i=1}^n T_{чи} \cdot T_{ч} \cdot K \quad (4.8)$$

где n – количество исполнителей, участвующих в разработке конкретного ПО; $T_{чи}$ – часовая тарифная ставка некоторого проектировщика программного продукта (тыс. тенге);

Φ_n – плановый фонд рабочего времени i -го исполнителя (дней), число рабочих дней в месяц;

$T_{ч}$ – количество часов работы в день (час), 8 часов; K – коэффициент премирования, составляет 1,38.

На разработку уйдет 1 месяц, из них рабочие дни составят:

$$\Phi_n = 24 \text{ раб. дня}$$

Учитывая специфику и трудность функционала образуется обязательный график работы программиста–проектировщика, принимающего участие в создании ПО, с классификацией знаний, специальности, уровня подготовки и статуса (таблица 4.4).

Таблица 4.4 - Данные по сотрудникам, задействованным в работе

Специалист-Исполнитель	Количество, чел-к	Зарботная плата в месяц, Тенге
Разработчик	1	150 000,00
Итого	1	150 000,00

Почасовая регулярная сумма определяется путем деления тарифной суммы, предложенной при еженедельном стандарте рабочего срока (Φ_p)

$$T_{\text{ч}} = \frac{T_{\text{м}}}{\Phi_p} \quad (4.9)$$

где $T_{\text{ч}}$ – часовая регулярная сумма (тыс.тенге); $T_{\text{м}}$ – месячная регулярная сумма (тыс.тенге).

Целый временной капитал:

$$\Phi_p = T_{\text{ч}} \cdot \Phi_{\text{п}} \quad (4.10)$$

Из этого следует:

$$\Phi_p = 8 \cdot 24 = 192 \text{ часов}$$

Регулярная сумма ставки проектировщика:

$$T_{\text{ч}} = 150000 / 192 = 781 \text{ тенге в час}$$

Согласно формуле (4.8) основная зарплата проектировщика будет равна

$$Z_{oi} = 781 \cdot 1,37 \cdot 192 = 178954,5 \text{ тенге}$$

Результаты главной зарплаты представлены в виде таблице 4.5.

Таблица 4.5 – Приведенные итоги вычисления издержек

Наименование содержания работ	Исполнитель	Трудоёмкость норма–час	Зарботная плата за час работы, т/час	Сумма заработной платы, тенге
ТЗ	Разработчик	45	781	35145
Моделирование	Разработчик	30	781	23430
Программирование	Разработчик	60	781	46860
Тестирование	Разработчик	29	781	22649
Внедрение	Разработчик	20	781	15620
Итого		184		143704

Дополнительная бонусная зарплата насчитывает 24% от ключевой зарплаты и ищется по формуле:

$$З_{дi} = З_{oi} \cdot Н_{д} / 100 \quad (4.11)$$

где $Н_{д}$ – коэффициент бонусной заработной платы разработчика

$$З_{дi} = 148978,5 \cdot 24 / 100 = 32863 \text{ тенге}$$

Итоговая сумма зарплаты деятельности проектировщика равен:

$$З_{фот} = 169700 + 35601 = 178590,5 \text{ тенге}$$

Социальные отчисления составляют 11% (ст. 358 п. 1 НК РК) от дохода работника, и вычисляются по формуле:

$$З_{сzi} = (ФОТ - ПО) \cdot 11\% \quad (4.12)$$

где ПО – пенсионный налог, который составляет 10% от ФОТ и социальными отчислениями не подлежит:

$$ПО = ФОТ \cdot 10\% \quad (4.13)$$

Таким образом:

$$ПО = 178590,5 \cdot 0,1 = 17859 \text{ тенге}$$

$$З_{сzi} = (178590,5 - 17859) \cdot 0,11 = 17789,47 \text{ тенге}$$

Затраты на ресурсы определяются по формуле:

$$M_i = (З_{oi} \cdot Н_{мз}) / 100\% \quad (4.14)$$

где $Н_{мз}$ – норма расхода материалов от основной заработной платы (3–5%)

$$M_i = 148978,5 \cdot 0,03 = 41469,35 \text{ тенге}$$

Расходы по статье «Аппаратное время» (P_{mi}) включают оплату техники, нужной для создания и отладки ПО, время использования которого определяется в соответствии с условными нормативами (в машино – часах) на 100 строк исходного кода ($Н_{мв}$) аппаратного времени в зависимости от классификации поставленных задач и типа ПК

$$P_{mi} = C_{mi} \cdot (V_{oi} / 100) \cdot H_{mb} \quad (4.15)$$

где C_{mi} – цена одного машино–часа (тыс.тенге);
 V_{oi} – общий объем ПО (строк исходного кода);
 H_{mb} – норматив расхода машинного времени на отладку 100 строк исходного кода (машино–часов).

Норматив расхода аппаратного времени на отладку 100 строк исходного кода вычисляется на основе таблицы Д.1 (Приложения Д) и составляет 13 ч/100 строк кода

$$P_{mi} = 781 \cdot (4770 / 100) \cdot 13 = 384984,2 \text{ тенге}$$

Издержки по пункту «Второстепенные расходы» (Π_{zi}) на конкретно ПО включают в себе расходы на закупку и установку определенной научно–технической информации и специальной литературы. Находим по стандарту, составленному в общем виде по компании, в процентах к главной зарплате:

$$\Pi_{zi} = Z_{oi} \cdot H_{pz} / 100 \quad (4.16)$$

где H_{pz} – норматив прочих затрат в целом по организации в (20%)
 Таким образом:

$$\Pi_{zi} = 189548,5 \cdot 0,2 = 39875,7 \text{ тенге}$$

Издержки по пункту «Накладные затраты» (P_{ni}), вычисляются по стандарту (H_{pn}) в нужном количестве к главной зарплате проектировщиков. Стандарт вычисляется обычно в компании

$$P_{ni} = Z_{oi} \cdot H_{pn} / 100\% \quad (4.17)$$

где P_{ni} – накладные расходы на конкретную ПО (тыс.тенге);
 H_{pn} – стандарт накладных затрат в общем по компании семьдесят процентов. После расчетов, имеем, что накладные затраты равны:

$$P_{ni} = 189548,5 \cdot 0,7 = 103514,95 \text{ тенге}$$

Итоговые затраты на проектировку базы знаний, формула (4.1)

$$C_{ni} = 174590,5 + 18989,47 + 5636,36 + 319713,2 + 29575,7 + 103514,95 = 754420 \text{ тенге}$$

Заключительные результаты расчета затрат на проектировку базы знаний и расходы в табличном и структурированном виде внесены в таблицу 4.6 и на рисунке 4.1.

Таблица 4.6 Результаты расчета затрат на разработку базы знаний

Затраты на разработку	Краткое выражение	Количество, тенге	В процентах от итогового числа
Объем награждения работы	$Z_{\text{фот}}$	189590,5	28
Социальные издержки	$Z_{\text{сзи}}$	16599,47	3
Ресурсы	M_i	5336,36	1
Аппаратные сроки	$P_{\text{ми}}$	310878,2	49
Побочные издержки	$P_{\text{зи}}$	28475,7	4
Накладные затраты	$P_{\text{ни}}$	103514,95	15
Результат		754420	100

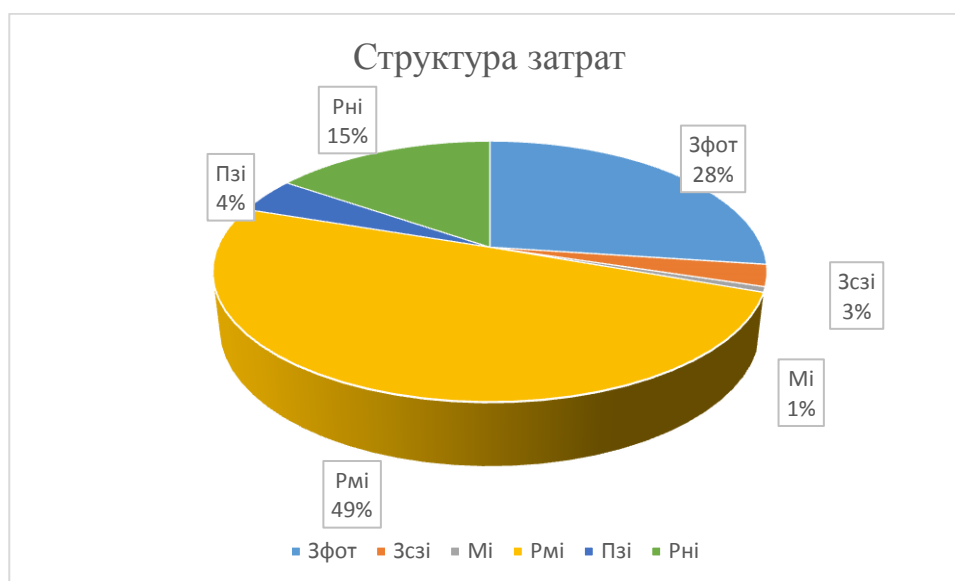


Рисунок 4.1 – Структура затрат на разработку экспертной системы

4.3 Расчет цены программного продукта

Расчет цены продукта, созданный самостоятельно проектировщиком и не собирается афишироваться, определяем с помощью:

$$C_{\text{пп}} = Z_{\text{пп}} + P_{\text{п}} + \text{НДС} \quad (4.18)$$

где $C_{пп}$ – цена программного продукта, тенге;
 $Z_{рпр}$ – затраты на создание продукта, в данном случае программного продукта, тенге;
 $\Pi_{п}$ – ожидаемый доход, тенге;
НДС – налог на дополнительную стоимость, тенге.

Запланированный доход равен (20%) от собственной стоимости проектировки

$$ПП = 754420 \cdot 0,2 = 150884 \text{ тенге}$$

НДС, начисленный на ПП, определяется так:

$$\text{НДС} = (Z_{рпр} + \Pi_{п}) \cdot k_{\text{НДС}} \quad (4.19)$$

где $k_{\text{НДС}}$ – ставка налога на добавленную стоимость.
Применив данные в формуле (4.19), вычисляем

$$\text{НДС} = (754420 + 167084) \cdot 0,12 = 102060,5 \text{ тенге}$$

Подставив данные в формулу (4.18) выводим результат

$$C_{пп} = 754420 + 167084 + 102060,5 = 936564,5 \text{ тенге}$$

Вывод. Разработка мобильного приложения – интересный и трудоемкий процесс, выполнение которого требует огромных затрат интеллектуального труда. Без логического мышления, определенных знаний и структурного подхода, это невозможно.

Была рассчитана общая стоимость продукта, на основе суммирования различных видов затрат, которые необходимы при производстве. Сделав анализ полученных расчетов, можно сделать вывод, что данный продукт является экономически прибыльным, потому что цена реализации продукта конкурентоспособна.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда

Тема данного дипломной работы – «Разработка мобильного приложения «City Info» на платформе Android

Цель данной работы - предоставить пользователям подробную информацию заведений и кинотеатров, возможность легко и быстро находить их по маршруту, построенным картой, а также возможность связываться с ними по телефонной связи.

Разработка мобильного приложения осуществляется с использованием компьютерной техники и электронного оборудования. В рассматриваемом помещении работают программист и дизайнер, у которых есть свое рабочее место. При разработке мобильного приложения ключевую роль играет правильная организация условий труда в рабочем помещении.

По причине того, что данная работа связана непосредственно с длительным нахождением за ПК, необходимо провести анализ пожарной безопасности и кондиционирования для обеспечения нормальных условий работы.

Основные задачи БЖД:

- распознавание любого вида опасности с определением количественных характеристик и координат;
- защита от угроз на основе сравнения затрат и выгод:- избежание возможных опасностей, следуя концентрации и остаточному риску, и ликвидация воздействия вреда на пользователя;
- опасность – это явления, процессы или объекты, способные в конкретных условиях принести вред здоровью пользователя напрямую или неявно.

Есть ряд вредоносных факторов, влияющих на людей работающих с видео_дисплейными терминалами (ВДТ) и персональными компьютерами (ПК):

- нехороший микроклимат комнаты: повышенная и пониженная температура воздуха; сильная запыленность и загазованность воздуха; повышенная и пониженная влажность воздуха;
- плохая освещенность рабочего места;
- превосходящие допустимые нормы шум;
- превосходящий уровень ионизирующего излучения;
- превосходящий уровень электромагнитных полей;
- превосходящий уровень статического электричества;
- угроза поражения электрическим током;
- недостаточная яркость экрана дисплея;
- несоблюдение эргономических норм при работе с ПК и др.

Следуя ГОСТу 12.1.005–88. ССБТ. Оптимальные и допустимые нормы микроклимата, в зависимости от типа работ, работа работников в помещении принадлежит к работе лёгкой сложности (1а), потому что управление оборудованием производится дистанционно с помощью ПК.

Для создания нормальных условий труда оператора установлены нормы производственного микроклимата. При работе с ПК должны соблюдаться нижеследующие правила:

В холодный период:

- Средняя температура 22-24 °С, допустимая – 18-26 °С;
- Относительная влажность воздуха 40-60%, допустимая 75%;

В теплый период:

- Средняя температура 23-25 °С, допустимая 20-30 °С;
- Относительная влажность воздуха 40-60%, допустимая влажность 55%.

Проанализируем помещение, в котором ведется разработка моб приложения:

- Размеры комнаты: длина 4.5 м; ширина 3.5 м, высота 3.5 м;
- остекление помещения – тройное (одно окно размером 2000 x 2000 мм);
- искусственное освещение – настольные светильники: 3 светильника, в каждом по 2 люминесцентные лампы (ПВЛМ–1x50);
- вид работы – разработка мобильного приложения «CityInfo»;
- количество рабочих мест – 2.

План помещения нарисован на рисунке 5.1.

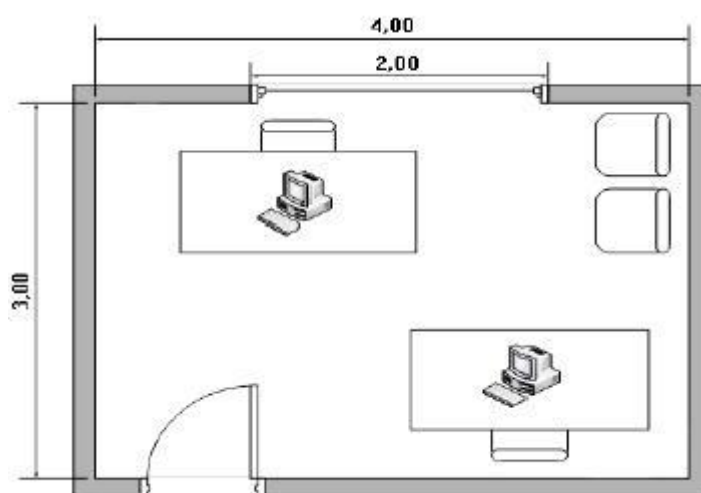


Рисунок 5.1 – План помещения

Рабочее место инженера-программиста оборудовано нижеперечисленным оборудованием:

- рабочий стол;
- стул-кресло;
- ноутбук;
- принтер.

Затем проанализирована обычная офисная оргтехника и степень ее вреда на работника.

1) Персональный компьютер.

К вредоносным излучениям компьютера можно отнести низкочастотные электромагнитные поля и ионизирующее (рентгеновское) излучение мониторов на электронно-лучевых трубках (ЭЛТ).

Теперь несколько слов о радиационном излучении ЭЛТ-мониторов. На самом деле, этот вид излучения есть в любом приборе на электронно-лучевых трубках. Но на расстоянии 50 см от экрана монитора уровень этого излучения не превышает уровня естественного фона. И еще мониторы на ЖКИ уже почти полностью вытеснили ЭЛТ, в которых нет радиационного излучения.

Есть наиболее опасные вредные факторы при работе с персональными компьютерами. Все они являются не прямыми опасностями, исходящими от этого вида техники, а эргономическими факторами, влияющими на здоровье человека.

Работа на персональных компьютерах принадлежит к зрительно напряженным работам. Это значит, что в первую очередь при работе с компьютером подвергаются воздействию наши глаза. Именно этот вредный фактор есть в большом количестве документов, регулирующий труд сотрудников с использованием ПЭВМ.

Согласно СанПиНам, есть три группы работ с видео--дисплейными терминалами и ПЭВМ, а также три категории сложности и напряженности работы. К первой группе работ (группа А) относятся работы по считыванию информации с дисплея, ко второй (группа Б) – работы по вводу информации, к третьей (группа В) – интерактивная работа с компьютером. Очевидно, что труд офисного сотрудника в большинстве случаев относится к группе В, потому что требует интерактивной работы с разным программным обеспечением.

На зрительную усталость особенно сильно действует также уровень освещенности рабочего места. А именно это видно при потребности одновременной работы с электронными и бумажными документами. Согласно СанПиН [3, п. 6.3.], уровень освещенности рабочего места при работе за компьютером должен составлять 309-509 лк. Однако при этом монитор и источники света должны располагаться так, чтобы не появлялись блики на поверхности экрана.

Следующей опасностью, поджидающей нас в офисе, является

постоянная поза при работе за компьютером. Постоянная напряженная поза при долгой работе на компьютере может привести к болям в мышцах, связках и сухожилиях спины и ног, заболеваниям позвоночника и суставов (остеохондроз, тендинит и пр.), а частое напряжение рук – к повреждениям запястья и сухожилий (так называемый синдром лучезапястного сустава или туннельный синдром). Эти заболевания вызываются так именуемыми травмами повторяющихся нагрузок и являются постепенно накапливающимся недомоганием, обусловленные продолжительными повторяющимися воздействиями и перетекающие в болезни нервов, мышц и сухожилий. Воздействие этого вредного фактора уменьшается при грамотной организации рабочего места – правильно подобранной мебели, удобном размещении элементов компьютера.

2) Принтер.

Если работа с персональным компьютером во многом регламентирована законодательно, то этого нельзя сказать об остальном оборудовании, используемом в офисе. Что касается принтера, то это понятно: его использование не требует постоянного присутствия работника, напряжения внимания и т.п. Вся работа с принтером приводит к его включению и выключению, добавлению бумаги и изыманию отпечатков. Даже замену картриджа чаще всего выполняет не работник, а обслуживающий персонал.

Но любой принтер является сложным электроприбором, поэтому работая с ним нужно выполнять определенные требования пожарной и электробезопасности. Так же в каждом принтере есть движущиеся части, а некоторые элементы могут нагреваться в процессе работы до высокой температуры (в матричных и струйных принтерах это печатные головки, в лазерных принтерах – специальные нагревательные валы).

Еще одним вредоносным фактором при работе с принтерами является шум. Наивысший уровень шума во время работы издают матричные принтеры, но в паспортах этих приборов уровень шума, как это является, не указан, а фактический уровень может быть измерен разве что при проведении аттестации рабочего места. Это единственный тип принтеров, уровень шума которых может быть сопоставим с максимально допустимым на рабочих местах, оборудованных ПЭВМ. Все остальные принтеры работают заведомо тише.

Шум ухудшает условия труда оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Эти нарушения в работе множеств органов и систем организма человека могут вызвать плохие изменения в эмоциональном состоянии человека прямо до стрессовых. Под действием шума уменьшается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением,

ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума [выше 80 дВ(А)] на слух человека приводит к его частичной или полной потере.

В табл. 1.1 указаны предельные уровни звука в зависимости от категории тяжести и напряженности труда, являющиеся безопасными в отношении сохранения здоровья и работоспособности.

Таблица 1.1 Предельные уровни звука, дБ, на рабочих местах.

Категория напряженности труда	Категория тяжести труда			
	I. Легкая	II. Средняя	III. Тяжелая	IV. Очень тяжелая
I. Мало напряженный	80	80	75	75
II. Умеренно напряженный	70	70	65	65
III. Напряженный	60	60	-	-
IV. Очень напряженный	50	50	-	-

Уровень шума на рабочем месте не должен быть больше 50 дБА. Для уменьшения уровня шума стены и потолок комнаты, где стоят компьютеры, могут быть облицованы звукопоглощающими материалами. Уровень вибрации в комнатах может быть снижен с помощью установки оборудования на специальные вибро изоляторы.

Таблица 1 – Показатели оптимальных величин микроклимата в офисе

Период Года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, С	Температура поверхностей, С	Относительная влажность воздуха	Скорость движения воздуха, м/с
Холодный	Ia(до 139)	22 – 24	21 – 25	40 – 60	0,1
Теплый	Ia(до 139)	23 – 25	22 – 26	40 – 60	0,1

Место разработки по зрительным условиям работы относится к классу легких работ (легкая физическая, категория Ia, работа производится сидя и не требует физического напряжения). Тип работы – проектирование базы знаний на персональном компьютере. Оборудование малошумящее – вред от шума отсутствует.

Основным источником электронных полей считаются заряды на дисплее персонального компьютера. Заряды электризуют взвешенные пылинки,

повышая концентрацию электричества возле рабочего места. Это может являться фактором возникновения недугов и заболеваний дыхательной системы, может способствовать развитию аллергических болезней, что имеет отрицательное влияние на систему здоровья пользователей. При несоблюдении технической безопасности, рабочий кабинет может поддаться возгоранию. Причиной пожара может служить неполадки в электрической технике, перегрузка проводов, неисправности в переключателях и розетках. На случай чрезвычайной ситуации в коридоре перед рабочим офисом расположен в подвешенном состоянии щит пожарного инвентаря, включающий в себя из: лопаты, ведра, лом, топор, ящик с песком. На этаже в коридоре находится внутренний противопожарный кран. Для тушения незначительного возгорания используют ручные углекислотные огнетушители типа ОУ–8, емкостью 8л. Такие огнетушители чрезвычайно эффективны для борьбы с огнем в закрытых помещениях и используются для аппаратов, находящихся под напряжением.

Проанализировав условия труда, рассмотрим более подробно следующие моменты.

5.2 Расчет системы кондиционирования

В Таблице 5.2 написаны оптимальные нормы параметров микроклимата с учетом периода года согласно СНиП 2.04.05-91. для облегченной физической работы. Оборудование, установленное в рабочей комнате, не является источником выделения тепла (очень незначительное выделение тепла аппаратурой никаким образом не оказывает влияние на микроклимат рабочего помещения). Климатические условия эксплуатации оборудования абсолютно совпадают с климатическими условиями, нормируемыми для рабочего персонала. Для вентиляции офисного помещения используются каналы естественной вентиляции, прокладываемые при строительстве здания и открытые окна летом. В теплый период времени для поддержания оптимального микроклимата используется кондиционер. Нормальный микроклимат в офисе дает хорошее самочувствие сотрудникам в любое время года, и соответственно продуктивность работы улучшается. Поэтому, для поддержания условий микроклимата в помещении, важнее всего оборудовать его системой кондиционирования.

Избыточное тепло определяется по формуле:

$$Q_{\text{ИЗБ}} = (Q_{\text{ОБ}} + Q_{\text{ОСВ}} + Q_{\text{Л}} + Q_{\text{Р}}) - Q_{\text{ОТД}} \quad (5.1)$$

Где $Q_{\text{ОБ}}$, $Q_{\text{ОСВ}}$, $Q_{\text{Л}}$ – тепло, выделяемое производственным оборудованием, системой искусственного освещения помещения и работающим персоналом (людьми) соответственно, ккал/ч;

Q_p – тепло, вносимое в помещение солнцем (солнечная радиация), ккал/ч;

$Q_{отд}$ – теплоотдача естественным путём, ккал/ч.

Тепло, выделяемое производственным оборудованием определяется по формуле:

$$Q_{об} = 860 \cdot P_{об} \cdot \eta \quad (5.2)$$

где 860 – тепловой эквивалент 1 кВт/ч;

$P_{об}$ – мощность, потребляемая оборудованием, кВт/ч;

η – коэффициент перехода тепла в помещение. Значение $\eta=0,95$ – норма потерь потребляемой мощности на тепловыделения компьютерного оборудования.

Для 2 компьютеров имеем:

$$Q_{об}=860 \cdot (2 \cdot 0,25) \cdot 0,95 = 408,5 \text{ ккал/ч}$$

Тепло, выделяемое осветительными установками, рассчитывается по формуле:

$$Q_{осв} = 860 \cdot N \cdot \eta \quad (5.3)$$

где N – расходуемая мощность светильников, кВт;

$\eta = 0,55$ – норма потерь потребляемой мощности на тепловыделения люминесцентных ламп.

$$Q_{осв}=860 \cdot 0,55 \cdot 0,53 = 236 \text{ ккал/ч}$$

Тепло, выделяемое людьми, рассчитывается по формуле:

$$Q_l = K_l \cdot q \quad (5.4)$$

где K_l – количество работающих;

q – тепловыделения одного человека, ккал/ч.

$q = 60$.

$$Q_l=2 \cdot 60 = 120 \text{ ккал/ч}$$

Тепло, вносимое солнечной радиацией, рассчитывается по формуле:

$$Q_p = m \cdot F \cdot q_{ост} \quad (5.5)$$

где m – количество окон в помещении;

F – площадь одного окна, m^2 ;

$q_{\text{ост}}$ – солнечная радиация через остеклённую поверхность, т.е. количество тепла, вносимое за один час через остеклённую поверхность площадью в 1 м^2 .

Для окна с двойным остеклением с деревянными переплетами $q_{\text{ост}} = 105$. Количество окон равно 1. Площадь окна равна $1 \cdot 2 = 2 \text{ м}^2$.

$$Q_p = 1 \cdot 2 \cdot 105 = 210 \text{ ккал/ч}$$

Для тёплого периода года при расчётах можно принять $Q_{\text{отд}} = 0$.

$$Q_{\text{изб}} = 408,3 + 246 + 120 + 210 = 984,2 \text{ ккал/ч}$$

При наличии теплоизбытков, количество воздуха которое необходимо удалить из помещения рассчитывается по формуле:

$$L_b = \frac{Q_{\text{изб}}}{C_b \cdot \Delta t \cdot \gamma_b} \quad (5.6)$$

где $Q_{\text{изб}}$ – избыточное тепло, ккал/ч;
 C_b – теплоёмкость воздуха ($0,24 \text{ ккал/кг}^\circ\text{C}$);
 $\Delta t = t_{\text{вых}} - t_{\text{вх}}$;
 $\Delta t = 10^\circ\text{C}$;
 $t_{\text{вых}}$ – температура воздуха, выходящего из помещения, $^\circ\text{C}$;
 $t_{\text{вх}}$ – температура воздуха, поступающего в помещение, $^\circ\text{C}$;
 $\gamma_b = 1,206 \text{ кг/м}^3$ – удельная масса приточного воздуха.

$$L_b = \frac{984,2}{0,24 \cdot 10 \cdot 1,206} = 340,6 \text{ м}^3/\text{ч}$$

Отношение количества воздуха, поступающего в помещение за один час, к объему помещения называется кратностью воздухообмена:

$$K = \frac{L}{V_n}$$

где V_n – объем помещения, м^3 ;
 $V_n = 4 \cdot 3 \cdot 3 = 36 \text{ м}^3$.

$$K = \frac{340,6}{36} \approx 9,15$$

Существующий оконный кондиционер имеет расход воздуха $459 \text{ м}^3/\text{ч}$. Теперь нужно определить требуемое количество таких кондиционеров:

$$N = \frac{340,1}{459} \approx 1 \text{ кондиционер}$$

Это соответствует действительности и является достаточным для обеспечения комфортного микроклимата.

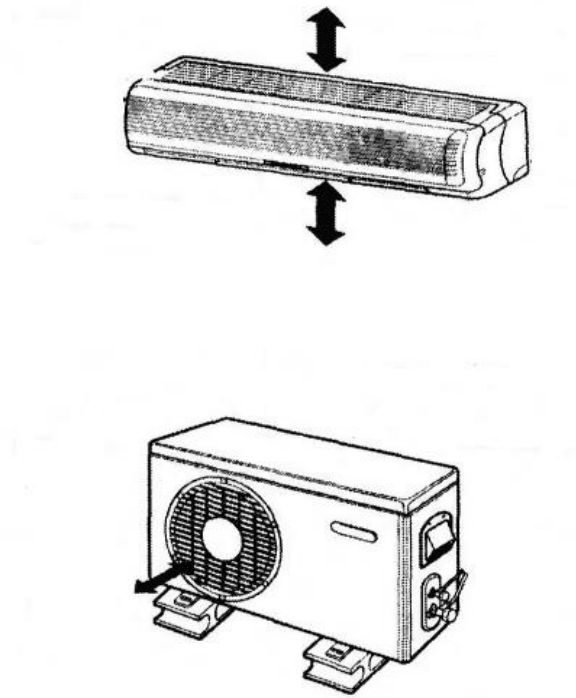


Рисунок 5.2 - Схема кондиционера сплит-системы

В нашем случае высота потолков не позволяет установить канальный кондиционер, поэтому можно выбрать установку сплит системы, показанная на рисунке 5.2. Конструкция кондиционера сплит-системы включает в себя наличие двух агрегатов - наружного и внутреннего блока. Как следует из названия, внутренний блок должен быть установлен внутри офиса, в то время как наружный блок будет располагаться снаружи здания. Кондиционирующую систему следует установить в соответствии с рисунком 5.3. Достоинством сплит-системы является низкий уровень шума, небольшие размеры и разнообразие функций.

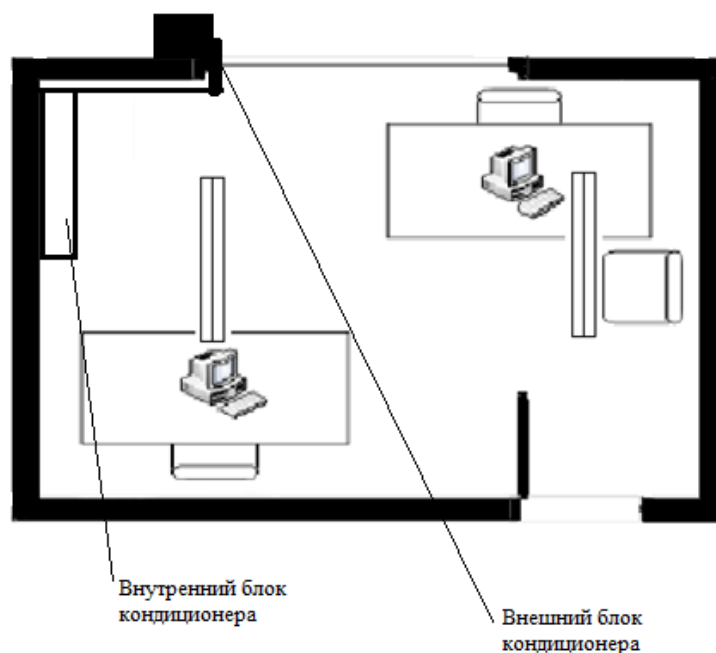


Рисунок 5.3 – Расположение кондиционера в помещении

5.3 Расчет системы автоматического пожаротушения

Здание по степени опасности развития пожара, от функционального назначения и пожарной нагрузки горючих материалов, относится к 1 – ой группе категории D.

Причинами возникновения пожара могут быть:

- возгорание элементов аппаратуры;
- возгорание отделочных материалов от неисправных выключателей, розеток;
- несоблюдение режимов эксплуатации оборудования, неправильное действие персонала.

При появлении пожара может пострадать не только помещение, также и дорогостоящие устройства и приборы, которые могут привести к человеческим жертвам. Поэтому, очень важно, принять меры по раннему выявлению и ликвидации пожаров. Источниками возгорания могут оказаться персональные компьютеры, ноутбуки, приборы, применяемые для технического обслуживания, электронные схемы ЭВМ, устройства электропитания, кондиционеры воздуха, где в результате различных нарушений, образуются перегретые элементы и прочее.

В соответствии с требованиями правил пожарной безопасности помещение оборудованы углекислотными огнетушителями ОУ – 5 с учетом – один огнетушитель на 100 м^2 . Общая площадь помещения составляет 15 м^2 таким образом устанавливаются 1 огнетушитель. В качестве огнетушащего вещества применяется комбинированный углекислотно

– хладоновый состав. Расчетная масса комбинированного углекислотно – хладонового состава m_d , кг, для объемного пожаротушения определяется по формуле

$$m_d = k \times g_n \times V \quad (5.8)$$

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

$g_n = 0,04$ – нормативная массовая концентрация углекислотно–хладонового состава;

V – объем помещения.

Объем помещения определяется по формуле

$$V = A \times B \times H \quad (5.9)$$

где $A = 5$ м – длина помещения;

$B = 3$ м – ширина помещения;

$H = 3$ м – высота помещения.

Тогда

$$V = 5 \times 3 \times 3 = 45 \text{ м}^3.$$

Следовательно

$$m_d = 1.2 \times 0.04 \times 45 \approx 2,16 \text{ кг}.$$

Расчётное число баллонов ξ определяется из расчета вместимости в 20-литровый баллон 12 кг углекислотно–хладонового состава.

Внутренний диаметр магистрального трубопровода d_i , мм равен

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

Эквивалентная длина магистрального трубопровода l_2 , м определяется

по формуле

$$l_2 = k_1 \times l \quad (6.0)$$

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

$l = 3$ м – длина трубопровода по проекту тогда

$$l_2 = 1.2 \times 3 = 3,6 \text{ м}.$$

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

Расчетное время подачи углекислотно–хладонового состава t , мин, определяется по формуле

$$t = \frac{m_d}{60 \times Q} \quad (6.1)$$

Таким образом

$$t = \frac{2,16}{60 \times 1,4} = 0,026.$$

Масса основного запаса углекислотно–хладонового состава m , кг, определяется по формуле

$$m = 1,1 \times m_d \times \left(1 + \frac{k_2}{k}\right) \quad (6.2)$$

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

$$m = 1,1 \times 2,16 \times \left(1 + \frac{0,2}{1,2}\right) = 2,77 \text{ кг.}$$

Это значит, из полученных результатов можем сделать вывод, что для обеспечения нормального функционирования системы автоматического пожаротушения нужно будет 1 баллон углекислотно–хладонового состава вместимостью 20 литров, с массой смеси 2,77 кг. Автоматические установки газового пожаротушения имеют устройства для автоматического пуска в соответствии с ГОСТ 12,4.009–83. Нахождение огнетушителя представлено на рисунке 5.2.

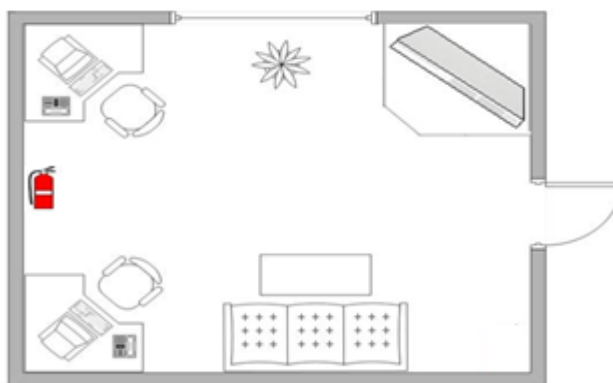


Рисунок 5.2 – Размещение огнетушителя в помещении

В разделе безопасность жизнедеятельности был проведен анализ условий труда и была произведена оценка рабочего помещения. Также были рассмотрены все факторы, негативно влияющие на деятельность разработчика и были произведены расчеты системы кондиционирования и системы авто пожаротушения.

Заключение

Разработка мобильного приложения – интересный и трудоемкий процесс, выполнение которого требует огромных затрат интеллектуального труда. Без логического мышления, определенных знаний и структурного подхода, это было бы невозможно.

При выполнении данной дипломной работы были рассмотрены и изучены основные принципы и требования разработки мобильных приложений. В рамках дипломной работы были достигнуты нижеперечисленные результаты:

1. Был выполнен анализ соответствующих приложений, имеющихся на Google Play;
2. Было создано приложение, выполняющее следующие функции:
 - позволяет туристам и жителям города Алматы выбирать заведения на свой вкус, просмотрев их подробное описание;
 - позволяет пользователям выбрать кинотеатр и посмотреть подробное расписание фильмов;
 - имеется возможность связываться по телефонной связи;
 - имеется навигация до выбранного объекта с использованием GoogleMap навигатора, а также построение кратчайшего пути.

Приложение «CityInfo» было протестировано и оно продемонстрировало свою работоспособность не только на стандартных эмуляторах, взятых из SDK Android, но и на реальных устройствах на платформе Android.

Была рассчитана общая стоимость продукта, на основе суммирования различных видов затрат, которые необходимы при производстве. Сделав анализ полученных расчетов, можно сделать вывод, что данный продукт является экономически прибыльным, потому что цена реализации продукта конкурентоспособна.

В разделе «Безопасность жизнедеятельности» были разработаны системы пожаробезопасности и кондиционирования, которые позволят предотвратить возникновение очага возгорания.

В процессе выполнения дипломной работы была достигнута цель разработки и выполнены все поставленные задачи.

Список используемой литературы

1. Медникс З., Дорнин Л. Программирование под Android. Издательство Питер, 2012.
2. Амелин К. С., Граничин О. Н., Кияев В. И., Корявко А. В.. Введение в разработку приложений для мобильных платформ. Издательство ВВМ, 2011
3. Голощапов, А.Л. Google Android: программирование для мобильных устройств / А.Л. Голощапов - СПб.: БХВ-Петербург, 2011. - 448 с.
4. СанПиН 9-131 РБ 2000 "Гигиенические требования к видеодисплейным терминалам, электронно-вычислительным машинам и организации работы", Мн., 2000.
5. Техника безопасности, №1, 2005. Голиков П.Е. Компьютерный зрительный синдром - основная проблема операторов ПК.
6. Хашими, С. Разработка приложений для Android / С. Хашими, С. Коматинени, Д. Маклин - СПб.: Питер, 2011 - 736 с.
7. Ретабоуил Сильвен. Android NDK. Создание приложений под Android на C/C++ – Москва: ДМК Пресс, 2012. – 296 с.
8. Эккель, Б. Философия Java. Библиотека программиста / Б. Эккель - СПб.: Питер, 2009. - 640 с.
9. ЕСКД. Общие требования к текстовым документам" (ГОСТ 2.105-95)
10. Официальный сайт обеспечения Android SDK [Электронный ресурс] - режим доступа: <http://developer.android.com>
11. Сатимова Е.Г. Проектирование баз данных. Методические указания к выполнению лабораторных работ (для студентов всех специальностей) – Алматы: АИЭС, 2009. - 37с.
12. Медникс Зигард, Дорнин Лаирд, Мик Блэйк, Накамура Масуми Программирование под Android – Питер:Москва, 2013. – 651 с.
13. Android A Programmers Guide - М. ИФРАН, 2011 – 399 с
14. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. З.Д. Еркешева, Г.Ш. Боканова Алматы: АУЭС, 2013 – 40 с.
15. ГОСТ 12.1.005-88 ОБЩИЕ САНИТАРНО-ГИГИЕНИЧЕСКИЕ ТРЕБОВАНИЯ К ВОЗДУХУ РАБОЧЕЙ ЗОНЫ
16. ГОСТ 30494-96 ПАРАМЕТРЫ МИКРОКЛИМАТА В ПОМЕЩЕНИЯХ

Приложение А

Листинг программной части

```
package com.example.home_pc.cityinfo.activities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;

import com.example.home_pc.cityinfo.R;
import com.example.home_pc.cityinfo.adapters.CoffeeHouseAdapter;
import com.example.home_pc.cityinfo.models.CoffeeHouse;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.model.LatLng;

import java.util.ArrayList;
import java.util.List;

public class RestaurantActivity extends AppCompatActivity {

    String name, description, phone, address, averageCost;
    Float lat, lng;
    int kitchen;
    TextView tvRestaurantName, tvRestaurantDescription, tvRestaurantPhone,
    tvRestaurantAddress;
    ImageView iVRestaurantPicture;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_restaurant);
        tvRestaurantName = (TextView) findViewById(R.id.tvRestaurantName);
        tvRestaurantDescription =
            (TextView) findViewById(R.id.tvRestaurantDescription);
        iVRestaurantPicture =
            (ImageView) findViewById(R.id.iVRestaurantPicture);
        tvRestaurantPhone = (TextView) findViewById(R.id.tvRestaurantPhone);
        tvRestaurantAddress =
            (TextView) findViewById(R.id.tvRestaurantAddress);

        getSupportActionBar().setHomeButtonEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        name = getIntent().getStringExtra("name");
        description = getIntent().getStringExtra("description");
        address = getIntent().getStringExtra("address");
        phone = getIntent().getStringExtra("phone");
        averageCost = getIntent().getStringExtra("average_cost");
    }
}
```

```

        lat = getIntent().getFloatExtra("lat", 0f);
        lng = getIntent().getFloatExtra("lng", 0f);
        kitchen = getIntent().getIntExtra("kitchen", 0);
        int image_url = getIntent().getIntExtra("image_url", 0);
        tvRestaurantName.setText(name);
        tvRestaurantDescription.setText(description);
        tvRestaurantPhone.setText(phone);
        tvRestaurantAddress.setText(address);

        if (image_url != 0) {
            iVRestaurantPicture.setImageResource(image_url);
        }

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_map, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.action_map) {
            Intent intent = new Intent(RestaurantActivity.this,
MapActivity.class);
            intent.putExtra("name", name);
            intent.putExtra("lat", lat);
            intent.putExtra("lng", lng);
            startActivity(intent);
            return true;
        }
        else if (item.getItemId() == android.R.id.home) {
            onBackPressed();
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
}

```

```

public class RestaurantActivity extends AppCompatActivity {

    String name, description, phone, address, averageCost;
    Float lat, lng;
    int kitchen;
    TextView tvRestaurantName, tvRestaurantDescription, tvRestaurantPhone,
tvRestaurantAddress;
    ImageView iVRestaurantPicture;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_restaurant);
        tvRestaurantName = (TextView) findViewById(R.id.tvRestaurantName);
        tvRestaurantDescription =
(TextView) findViewById(R.id.tvRestaurantDescription);
    }
}

```

```

        iVRestaurantPicture =
(ImageView) findViewById(R.id.iVRestaurantPicture);
        tvRestaurantPhone = (TextView) findViewById(R.id.tvRestaurantPhone);
        tvRestaurantAddress =
(TextView) findViewById(R.id.tvRestaurantAddress);

        getSupportActionBar().setHomeButtonEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        name = getIntent().getStringExtra("name");
        description = getIntent().getStringExtra("description");
        address = getIntent().getStringExtra("address");
        phone = getIntent().getStringExtra("phone");
        averageCost = getIntent().getStringExtra("average_cost");

        lat = getIntent().getFloatExtra("lat", 0f);
        lng = getIntent().getFloatExtra("lng", 0f);
        kitchen = getIntent().getIntExtra("kitchen", 0);
        int image_url = getIntent().getIntExtra("image_url", 0);
        tvRestaurantName.setText(name);
        tvRestaurantDescription.setText(description);
        tvRestaurantPhone.setText(phone);
        tvRestaurantAddress.setText(address);

        if (image_url != 0){
            iVRestaurantPicture.setImageResource(image_url);
        }

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_map, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.action_map) {
            Intent intent = new Intent(RestaurantActivity.this,
MapActivity.class);
            intent.putExtra("name", name);
            intent.putExtra("lat", lat);
            intent.putExtra("lng", lng);
            startActivity(intent);
            return true;
        }
        else if (item.getItemId() == android.R.id.home) {
            onBackPressed();
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
}

```

```

public class RestaurantFragment extends Fragment {

    private ListView listViewRestaurant;

    private RestaurantAdapter restaurantAdapter;

    private List<Restaurant> restaurantList;

    public RestaurantFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_restaurant, container,
false);

        listViewRestaurant = (ListView)
view.findViewById(R.id.listViewRestaurant);

        restaurantList = new ArrayList<>();

        new GetAllRestaurant().execute();

        listViewRestaurant.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                Restaurant restaurant = restaurantList.get(position);
                Intent intent = new Intent(getActivity(),
RestaurantActivity.class);
                intent.putExtra("name", restaurant.getName());
                intent.putExtra("description", restaurant.getDescription());
                intent.putExtra("phone", restaurant.getPhone());
                intent.putExtra("address", restaurant.getAddress());
                intent.putExtra("average_cost", restaurant.getAverageCost());
                intent.putExtra("lat", restaurant.getLat());
                intent.putExtra("lng", restaurant.getLng());
                intent.putExtra("kitchen", restaurant.getKitchen());
                intent.putExtra("image_url", restaurant.getImageURL());
                startActivity(intent);
            }
        });

        listViewRestaurant.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
            @Override
            public boolean onItemLongClick(AdapterView<?> parent, View view,
int position, long id) {
                Restaurant restaurant = restaurantList.get(position);
                Intent intent = new Intent(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:" + restaurant.getPhone()));
                intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
                return true;
            }
        });
}
}

```

```

        return view;
    }

    private class GetAllRestaurant extends AsyncTask<Void, Void, Void> {

        private DatabaseConnector connector = new
DatabaseConnector(getActivity());

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
        }

        @Override
        protected void onPostExecute(Void aVoid) {
            connector.close();
            restaurantAdapter = new RestaurantAdapter(getActivity(),
restaurantList);
            listViewRestaurant.setAdapter(restaurantAdapter);
            super.onPostExecute(aVoid);
        }

        @Override
        protected Void doInBackground(Void... params) {
            connector.open();
            restaurantList.addAll(connector.getAllRestaurant());
            Log.d("restaurant", restaurantList.toString());
            return null;
        }
    }
}

```