

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой Курамбаев З.К.
профессор, к.ф.-м.н.
(Ф.И.О., ученая степень, звание)

« » 20 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка мобильного приложения «Правильное питание» для операционной системы Android

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Жангабекова С.Т. ПО-12-2
(Фамилия и инициалы) группа

Научный руководитель _____
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмусина А.Ч., к.э.н., доцент
(Фамилия и инициалы, ученая степень, звание)
АЧ «03» 05 2016 г.
(подпись)

по безопасности жизнедеятельности:

Тришкель Николай Георгиевич, д.т.н., проф.
(Фамилия и инициалы, ученая степень, звание)
ТН «28» 04 2016 г.
(подпись)

по применению вычислительной техники:

Шайхин Б.М., к.ф.-м.н., доц.
(Фамилия и инициалы, ученая степень, звание)
БМ «25» 05 2016 г.
(подпись)

Нормоконтролер: Шайхин Б.М., к.ф.-м.н., доцент
(Фамилия и инициалы, ученая степень, звание)
БМ «25» 05 2016 г.
(подпись)

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

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмических и информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Джангобкова Салтанат Талматбековна
(фамилия, имя, отчество)

Тема проекта Разработка мобильного приложения "Травильное питание" для операционной системы Android

утверждена приказом ректора № 21 от «10» марта 2016 г.

Срок сдачи законченной работы «__» _____ 20__ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Разработать мобильное приложение "Травильное питание" для ОС Android

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Перечень графического материала (с точным указанием обязательных чертежей)

1 рисунок в главе 1
 0 рисунков в главе 2
 12 рисунков в главе 3
 5 рисунков в главе 4
 1 рисунок в разделе 5

Рекомендуемая основная литература

Рето М. Android в профессиональном применении для планшетных компьютеров и смартфонов

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БЭИ	Трихубко Н.Г.	25.02 - 25.04.16	[Подпись]
Экспертная часть	Бокшичев А.Ч.	12.03 - 03.05.16	[Подпись]
Глава 1	Майхин Б.М.	05.01 - 29.02.	[Подпись]
Глава 2	Майхин Б.М.	01.03 - 29.02.	[Подпись]
Глава 3	Майхин Б.М.	01.04. - 30.04.	[Подпись]
Корректировки	Майхин Б.М.	25.05.2016	[Подпись]

Аннотация

В дипломной работе рассмотрены и изучены технологии и инструменты для разработки мобильного приложения для операционной системы Android. Разработанное мобильное приложение является помощником в поддержании здорового питания. Приложение отвечает всем предъявленным требованиям для публикации в магазине мобильных приложений PlayMarket.

Произведены расчеты затрат и анализ условий труда в соответствующем разделе дипломной работы.

Annotation

In the thesis work the technology and tools were considered for application development on the Android platform. The developed application is a helper for healthy diet maintenance. The application conform to all the requirements of publication in the mobile application store Google Play.

The relevant section of the thesis were made cost calculation and analysis of working environment with the reasons.

Андатпа

Дипломдық жобада Android операциялық жүйесіне арналған мобильдік бағдарламаның дамуы үшін технологиялары және құралдары қарастырылды. Дамылған мобильдік бағдарлама салауатты тамақтануның қолдаушысы болып табылады. Қолдану мобильді қосымша PlayMarket дүкенде жариялауы үшін барлық талаптарына сай келеді.

Дипломдық жұмыстың бөлек бөлімдерінде экономикалық есептеулер жүргізілген және еңбек жағдайының талдауы жасалған.

Содержание

АННОТАЦИЯ.....	1
ВВЕДЕНИЕ	7
1 ОСОБЕННОСТИ РАЗРАБОТКИ.....	9
1.1 ПРИНЦИП РАБОТЫ ANDROID OS И ANDROID ПРИЛОЖЕНИЙ.....	9
1.1.1 Основные характеристики среды разработки для платформы Android...	16
1.2 СТРУКТУРА ANDROID ПРИЛОЖЕНИЙ.....	11
2 ИНСТРУМЕНТЫ РАЗРАБОТКИ.....	16
3 ЭТАПЫ РАЗРАБОТКИ	32
3.1 ОБЗОР И АНАЛИЗ АНАЛОГОВ ПРИЛОЖЕНИЙ	32
3.2 ПРОЕКТИРОВАНИЕ	33
4.2 РАЗРАБОТКА ПРИЛОЖЕНИЯ	42
4.2.1 СОЗДАНИЕ И РЕАЛИЗАЦИЯ ФОРМ	42
5 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ	57
5.1 АНАЛИЗ УСЛОВИЙ ТРУДА ПРИ РАЗРАБОТКЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ	57
5.1.1 ОПАСНЫЕ И ВРЕДНЫЕ ФАКТОРЫ	58
5.1.2 ЭЛЕКТРОМАГНИТНОЕ ИЗЛУЧЕНИЕ	59
5.1.3 ЭРГОНОМИКА ДИСПЛЕЯ И УСТРОЙСТВ ВВОДА ИНФОРМАЦИИ	59
5.2. ОПТИМАЛЬНЫЙ РЕЖИМ РАБОТЫ	61
5.3 РАСЧЕТ ЕСТЕСТВЕННОГО ОСВЕЩЕНИЯ	62
4 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА	45
4.1 ТЕХНИЧЕСКОЕ ОБОРУДОВАНИЕ, ИСПОЛЬЗОВАННОЕ ПРИ РАЗРАБОТКЕ	46
4.2.1 ТЕХНИЧЕСКИ-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ИСПОЛЬЗУЕМОЕ В РАБОТЕ	46
4.3 РАСХОДЫ НА ОПЛАТУ ТРУДА ОСНОВНЫМ РАЗРАБОТЧИКАМ И ПРОГРАММИСТАМ	47
4.4 РАСЧЕТ ЗАТРАТ ПО СОЦИАЛЬНОМУ НАЛОГУ	50
4.6 РАСЧЕТ АМОРТИЗАЦИОННЫХ ОТЧИСЛЕНИЙ	51
4.7. РАСЧЕТ ЗАТРАТ НА ЭЛЕКТРОЭНЕРГИЮ	51
4.8 РАСЧЕТ НАКЛАДНЫХ РАСХОДЫ	52
4.9 ЦЕНА РЕАЛИЗАЦИИ ПРОЕКТА	55

ЗАКЛЮЧЕНИЕ	69
------------------	----

ПРИЛОЖЕНИЕ А.....	71
-------------------	----

Введение

В наше время во всем мире начинается повсеместное распространение различных технологий, к примеру, таких как смартфоны и планшетные компьютеры. Эти устройства позволяют решать многие проблемы и задачи, на выполнение которых много лет назад требовало наличие персонального компьютера. К подобным задачам относятся: поиск информации в интернете, общение людей через социальные сети, различные игры и т.д. С появлением мобильных устройств разработка и реализация мобильных приложений стала особенно актуальной. К современным распространенным системам для мобильных устройств относятся: Android, IOS, CyanogenMod, Cyanogen OS, Fire OS, Flyme OS, iOS, Windows Phone, BlackBerry OS, Firefox OS, Sailfish OS, Tizen, Ubuntu Touch. Выбор платформы является одной из спорных проблем для любого нового проекта. Последние наблюдения свидетельствуют, что на мировом рынке смартфонов в значительной степени идет гонка между iOS и Android. За четвертый календарный квартал 2015 года обе мобильные операционные системы вместе составляют 98,4% всего мирового рынка смартфонов. Android по-прежнему остается самой распространённой мобильной операционной системой в мире с показателем в 80,7% доли рынка, на счету iOS – 17,7%. 3 место занимает операционная система Windows Phone, она разработана всем известной компанией Microsoft. Во второй половине 2015 года Windows Phone составила на 2,6% смартфонов, на которые установлена эта ОС. Катализатором роста платформы являются успешные продукты от Sony, Samsung, HTC, LG, ZTE и Huawei. Общая сумма приложений для Android в магазине приложений Google Play превышает 1,44 млн. Samsung Galaxy S6 edge 32Gb – является победителем среди смартфонов 2015 года. Вместе с распространением мобильных платформ также наблюдается рост рынка мобильных приложений. Мобильные приложения разрабатываются для многих различных задач: GPS-навигация, работа с текстом, интернет обозреватели.

Раньше разработчики писали программу в текстовых редакторах, но с точки зрения удобства и производительности куда лучше использовать интегрированную среду разработки (IDE). На данный момент актуальной средой разработки под него является Android Studio. Android Studio позиционируется как средство для быстрой, эффективной разработки мобильных приложений и также является заменой Eclipse. В отличие от Eclipse среда разработки Android работает быстрее, удобнее и почти все разработчики одобрили эту среду. Программа позволяет проверить приложение на устройствах с различным размером экрана и даже с разными версиями API. Одними из «фишек» стоит отметить встроенную интеграцию с Github,

поддержку нескольких APK с различными функциями внутри одного проекта, модуль оптимизации с монитором оперативной памяти. Также можно отметить, что среда разработки Android Studio набирает своих сторонников за счет удобства данной SDK и больших функциональных возможностей.

Целью моей дипломной работы является разработка мобильного приложения «Правильное питание» для смартфонов под управлением операционной системы Android с поддержкой всех современных устройств на базе Android версии 2.2 и выше. Данный проект содержит в себе счетчик калорий, который будет высчитывать суточную норму для одного человека. Так же в данном приложении расписано меню питания на каждый день. Приложение «Правильное питание» позволит урегулировать питание пользователя, что в последующем поможет предотвратить множество болезней и нормировать свой режим дня. Питание и здоровье человека является одной из самых актуальных проблем. Таким образом, учитывая вышеуказанные особенности Android и дальнейшие перспективы развития программирования мобильных приложений, можно с уверенностью сказать, что изучение данной платформы имеет смысл быть. Хотелось бы отметить тот факт, что данное приложение является кроссплатформенным, что позволяет создавать приложения на любой платформе. С течением времени спрос на программистов-разработчиков различных приложений будет стабильно расти. Этим и обусловлена актуальность данной работы.

1 Особенности разработки

1.1 Принцип работы Android OS и Android приложений

Операционная система Android реализует идею адаптации программ различного назначения к мобильным устройствам. Это многофункциональная платформа, представляющая собой программный стек операционной системы на основе Linux, который предназначен для управления памятью и процессами устройств.

К достоинствам данной операционной системы по моему мнению относятся:

- скорость работы, то есть интерфейс системы практически не тормозит;
- интерфейс платформы достаточно удобен и понятен ;
- можно установить множество многофункциональных бесплатных программ. Каталог программ в GooglePlay очень обширный;
- регулярные обновления. С каждой новой версией исчезают недостатки и система становится функциональнее и удобнее.

Магазин приложений GooglePlay имеет несколько минусов, в частности если речь заходит о качестве самих приложений. Однако для того чтобы разработанное приложение стало доступно в магазине Google, необходимо внести плату всего один раз 25 долларов, где в Apple нужно ежегодно оплачивать по 99 долларов, что является ударом по карману. Более того, пройдет всего несколько часов с момента отправления приложения на одобрение Google до момента, когда оно станет доступно широким массам. В случае с App Store понадобятся недели.

В библиотеках Android есть много различных функций, связанных с видео, графикой, программированием пользовательских интерфейсов и многими возможностями мобильных устройств. Платформа Android ,сама по себе предназначена для разработки под мобильные устройства, у нее есть качества полноценного фреймворка для локального компьютера. Google дает возможность писать для этого фреймворка программы на Java, предоставляя разработчикам большой комплект инструментов (Software Development Kit) под названием Android SDK. При работе с этим инструментарием, почти не возникает ощущения, что программа создается для мобильного устройства, так как в нем открыт доступ к большинству библиотек классов, используемых на локальном компьютере или сервере, в том числе к реляционной базе данных.

Большинство функций платформы Java Standard Edition (Java SE), исключая абстрактного оконного интерфейса (Abstract Window Toolkit, AWT) и Swing поддерживаются комплектом Android SDK. Вместо AWT и Swing в Android имеется свой расширенный современный фреймворк пользовательского интерфейса. Поскольку многие приложения пишутся на Java, может появиться надобность виртуальной машины Java (Java Virtual Machine, JVM), она отвечает за интерпретацию исполняемого байт-кода Java. JVM осуществляет

необходимую оптимизацию, чтобы Java могла достичь уровней производительности, сравнимых с аналогичными уровнями транслируемых языков — таких как C и C++. Android предлагает собственный оптимизированный вариант JVM для исполнения скомпилированных файлов классов Java. Это делается, чтобы преодолеть ограничения, которые свойственны для мобильных устройств, а именно связанные с памятью, скоростью работы процессора и мощностью, все это позволяет ART VM виртуальная машина. Доступность и простота языка Java, усиленная обширной библиотекой классов Android, позволяет создавать многофункциональные приложения. Программный код (вместе со всеми файлами ресурсов и прочей необходимой информацией) запаковывается в специальный файл-архив, Android Package, который имеет расширение *.apk и компонуется утилитой aapt tool. Дальше он распространяется как программа и устанавливается на смартфоны. Один такой файл связан с кодом одного приложения. И каждое приложение в Android живет в своем собственном мире – в такой машине.

Для многих разработчиков главным преимуществом Android является открытость платформы. Платформа Android создана на основе открытого исходного кода и распространяется на свободной основе. Благодаря этому разработчики могут получить доступ к исходному коду Android и разобраться, каким образом реализованы особенности и функции приложений. Все пользователи и другие разработчики могут участвовать в совершенствовании и улучшении операционной системы Android. Для этого существует возможность отправить отчет об обнаруженных ошибках (на веб-странице source.android.com/source/report-bugs.html) и принять участие в одной из групп дискуссий Open Source Project (source.android.com/community/index.html). В интернете есть доступ на различные приложения Android с открытым исходным кодом, которые предлагает компания Google и рядом других производителей.

Главное преимущество Android-приложения это возможность использовать ресурсы других приложений, если они дают на это соответствующие права. Android-приложениях включают четыре типа компонентов: Activities, Services, Broadcast receivers и Content providers. Приложения для Android в своей работе использует окна, в отличие от системы Windows окна имеют иное название – Activity. У каждого окна есть свой жизненный цикл и особенности. Для того чтобы создать окно нужно использовать метод onCreate(), нем происходит инициализация приложения и его элементов. Методы onStart() и onResume() активизируются перед отображением окна при его создании, либо восстановлении. При сворачивании вызываются методы onPause() и onStop(). При закрытии приложения и окна вызывается onDestroy(), в этом методе есть возможность сохранить свои данные и параметры.

На платформе Android есть встроенная поддержка 2-D и 3-D графики, включая библиотеку OpenGL. Хранение данных осуществляется благодаря встроенной популярной базе данных SQLite с открытым исходным кодом. На

рисунке 1.1 показана упрощенная схема уровней программного обеспечения Android.

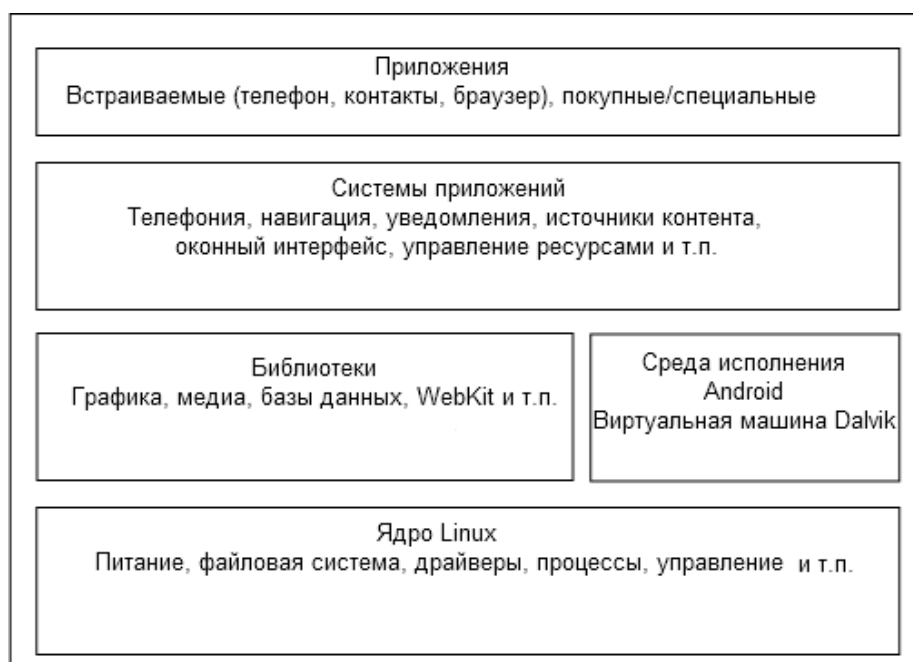


Рисунок 1.1 – Уровни программного обеспечения Android

1.2 Структура Android приложений

Можно выделить четыре типа компонентов: Activities, Services, Content providers, Broadcast receivers.

Activity является единицей пользовательского интерфейса, и представляет собой текущий активный экран. К примеру, в приложении для обмена сообщениями одно activity используется для отображения листа контактов, другое для того чтобы создать сообщение и так далее. Все activity в приложениях работают совместно и формируют один пользовательский интерфейс, при этом они независим друг от друга. Каждое из них наследуется от базового класса Activity и осуществляет создание окна. В данном окне разработчик может поместить свой интерфейс. Мобильные приложения могут состоять из одного или сразу нескольких activity. Переход от одного Activity к другому осуществляется с помощью вызова следующего из текущего Activity. Каждое Activity вызывает окно по умолчанию. Окно обычно создается в полноэкранном виде, также может находиться поверх других окон. Activity имеет возможность задействовать другие дополнительные окна. Визуальное содержание окна состоит из объектов, наследуемых от базового класса View. Каждый компонент является прямоугольным пространством внутри окна. Основные компоненты хранят в себе дочерние и определяют их место расположения. Иерархию элементов можно рассмотреть в виде дерева, а составляющие компоненты, которые расположены в самой нижней его части можно представить в виде “листьев”, они не имеют дочерних компонент и

ожидают указаний пользователя на этом участке. Таким образом организуется интерактивное взаимодействие с пользователем. Например, можно отобразить на экране маленькую иконку и присвоить ей какое-нибудь действие, то есть при нажатии пользователем на иконку, она будет исполнять присвоенную функцию. В операционной системе Android уже встроены наборы визуальных компонент, которые разработчик может свободно использовать. Эти наборы состоят из : кнопок, текстовых полей, полос прокрутки, меню и многие другие визуальные элементы. Для вызова иерархии элементов в окно используется метод `Activity setContentView()`. Экземпляр класса `View` является параметром метода `Activity setContentView()` и он лежит в корне иерархии.

Второй вид компонентов `Services` работают в фоновом режиме. Фоновым режимом пользуются для длительных операций и обеспечения работы удаленных процессов. Сервис это не пользовательский интерфейс, точнее он не взаимодействует с пользователем. Сервисы связаны с другими компонентами, такими как `Activity`. Сервис так же может запускаться вместе с системой.

`Content providers` распоряжается наборами данных, которые в базе данных `SQLite`, в сети, или в месте, где приложение будет иметь к ним доступ. С помощью `content provider` другое приложение может запросить данные и, если имеется разрешение, оно может изменять их. Например, система Android содержит `content provider`, который управляет пользовательской информацией о сообщениях. Он позволяет любому приложению, обладающему соответствующими правами вызвать составляющие этого компонента и считать, далее изменять информацию в данном сообщении. Общей функцией `content provider` считывание и запись данных, которые используют приложения. Приложение `Note Pad` использует этот компонент для сохранения сделанных записей. Для того, чтобы другие приложения могли использовать данные , им необходим стандартный набор API.

`Broadcast receivers` это компонент отвечающий за распространение общесистемных сообщений и отслеживание их действий. К таким сообщениям относится информация о том, садится заряд батареи или информирование что данные загружены на устройство и готовы к использованию. `Broadcast receivers` не предоставляет пользовательского интерфейса, но при этом, он создает уведомления в строке состояния, и предупреждает пользователя о произошедшем событии. `Broadcast receiver` взаимодействует с другими компонентами для того, чтобы уменьшить свой объем работы. Так, он может инициировать сервисы для выполнения действий, привязанных к какому-то событию.

1.1.1 Набор библиотек для функционала приложений

`FreeType` библиотеки нужна для работы с массивами битов и форматирования различных шрифтов. Эту библиотеку можно назвать движком для работы с шрифтами и для создания текстов.

LibWebCore - это библиотеки известного движка WebKit, который используется браузером Google Chrome.

SGL -это библиотека, которая является своего рода доступным движком для работы с графикой. SGL библиотека является продуктом компании Google, она применяется и в других различных программах.

Библиотека Libc –это библиотека программного языка C, и используется для работы на устройствах которые базируются на Linux.

Графические библиотеки поддерживают стандартные форматирования, они дают возможность редактировать ориентацию рисунков. Графическая библиотека хранит в себе объект Camera, с его помощью двухмерном реализовывать трехмерные эффекты. Всистему Android встроен OpenGL ES 1.0 стандарт, благодаря этому, приложения поддерживают трехмерную графику. Он является плоским API на базе языка C.

Интегрированный поиск Android (Integrated Android Search) - третья по счету идея, которая связана с домашней страницы. Этот вид поиска дает возможность производить как на самом устройстве, так же и в Интернете. Функция поиска Android так же позволяет с помощью команд управлять поиском. Более того, Android интерпретации движений пальцев пользователя, то есть управление жестами устройства с помощью сенсорного экрана. Android записывает различные последовательности движений пользователя по экрану и сохраняет их как жесты. Такие жесты используются приложениями, чтобы обозначить конкретные действия.

В принципе существуют и другие самостоятельные инновации, которые облегчают сам процесс разработки. Примеры таких инноваций: PhoneGap , XML/VM и Titanium и другие.

Еще одним важным компонентом Android SDK являются сервисы. Этот компонент SDK предоставляет разработчикам приложений интерфейсы, которые предназначены для работы с картами, получения информации о местоположении устройства в реальном времени.

Так же Android имеются специальные интерфейсы API, которые нужны для работы с аудио и видео компонентами.

В старых версиях Android можно было записывать только аудио. Начиная с версии 1.5, при помощи MediaRecorder, была предоставлена возможность записывать как аудио, так и видео. В версиях выше Android 2.0 применяется движок Pico, он нужен чтобы преобразовать текст в речь.

Следует отметить, что в Android все эти компоненты объединяются созданием единого XML-файла, в нем определяется, каким будет пакет прикладных программ. Манифест описывает структуру, данные и требования создаваемого приложения. В манифесте находятся все узлы каждого компонента (Activity, Service и другие), из которых строится мобильное приложение, и с помощью Intent Filters определяет, способ их взаимодействия друг с другом и с внешними программами. В манифесте хранятся атрибуты указывают на метаданных. Дополнительные узлы верхнего уровня используются для настройки безопасности, юнит-тестов и системных

требований. Так же в нем есть корневой тег <manifest> с атрибутом package, и он ссылается на пакет проекта. Тег <manifest> включает в себя атрибут xmlns:android, который поддерживается системными узлами внутри файла.

1.1.2 Ресурсы

Ресурсы это файлы ,которые используются программой и встраиваются во время исполнения ,без изменений исходного кода,и встраиваются в него во время работы. В Android есть поддержка различных видов файлов ресурсов, таких как XML, PNG и JPEG. Файлы XML хранят различные форматы, и зависят от того, что они будут описывать. Есть несколько типов ресурсов:

- layout-файлы ,они находятся в “/res/layout/”;
- изображения находятся в “/res/drawable/”;
- анимации хранятся в “/res/anim/”;
- различные стили, строки ,массивы ,которые хранятся в “/res/values/”;
- arrays.xml файл служит для определения массивов;
- colors.xml используются для форматирования цветов;
- dimens.xml файл необходим для определения размеров (dimensions);
- strings.xml файл используется для определения строк;
- styles.xml файл работает со стилями объектов.

Чтобы использовать ресурс в программе необходим полный ID ресурса и тип объекта. Тип ресурса содержится в resource_type ,который является подклассом R. Resource_name –имя файла для ресурса, который определяется другими типами файла. Каждый тип ресурса добавляется в подкласс R.

Ресурсы, откомпилированные в приложении, могут использоваться без названия пакета . В Android есть свои стандартные ресурсы как стили экрана и фоновые кнопки. К этим ресурсам можно обратиться при написании программы через android.R.resource_type.resource_name.

Значения атрибуте и ресурсе может ссылаться на другой ресурс. Ссылки часто используется в layout файлах для хранения строк и изображения, при этом ссылка может быть и на другие типы ресурсов, такие как цвета и числа.

1.3 Назначение мобильного приложения

Цель, создаваемого мною мобильного приложения, состоит в том ,чтобы помочь людям следить за своим питанием и здоровьем. Потребности любого человека в энергии зависят от возраста, пола, роста и образа жизни. Если калорийность суточного рациона будет выше, чем потребность организма в энергии, то лишние калории, будут откладываться и это приведет к увеличению веса. А если же наоборот, то организм переходит в «режим экономии»,из-за недостатка калорий может появиться слабость, истощение и различные недомогания. В программе есть счетчик калорий, который высчитывает количество калорий необходимое для вашего организма по формуле Миффлина — Сан Жеора . Миффлина-Сан Жеора появилась недавно, но она считается одной из самых точных на сегодняшний день.

Формула 1.3 по которой высчитывается суточная норма калории выглядит так:

$$10 \times \text{вес (кг)} + 6,25 \times \text{рост (см)} - 5 \times \text{возраст (лет)} - 161 \quad (1.3)$$

Далее полученный результат умножается на коэффициент физической нагрузки:

- 1.2 – Низкая(минимум или отсутствие физической нагрузки);
- 1.4625 – Средняя(занятия фитнесом 5 раз в неделю);
- 1.6375 – Высокая(занятия фитнесом каждый день).

Так же мобильная программа содержит подготовленное меню, которое состоит только из полезных и богатых витаминами блюд. Нет необходимости соблюдать различные диеты, так как в приложении указаны все продукты, которые нежелательно употреблять. Еще его одним плюсом, в отличии от многих, является возможность поделиться информацией в различных социальных сетях. Так данное приложение рассчитано не только для женщин, но и для мужской аудитории, то не мало важным аспектом является разработка дизайна. Фишкой дизайна моего приложения является –выбор предпочитаемой расцветки(розовая, голубая и фиолетовая).

1.1.1 Основные характеристики среды разработки для платформы Android

Пару лет назад у Google не было своей среды разработки (Integrated Development Environment - IDE). Поэтому разработчики использовали другие доступными среда. Одними из самых распространенных являются: Eclipse, NetBeans и IntelliJ IDEA. Однако после долгого ожидания в декабре 2014 года Google выпустила первую стабильную версию своей интегрированной среды разработки Android Studio 1.0. Примерно в то же время остановилась поддержка Android Development Tools (ADT) для Eclipse. Android Studio - это пакет инструментов для разработки приложений под Android OS. Эта среда разработки имеет следующие функции:

- wysiwyg-редактор, возможность работать с UI;
- сборка проекта с использованием Gradle;
- создание нескольких .apk файлов;
- lint-анализатор кода позволяющий находить ошибки и дает подсказки для их исправления;
- proguard- встроенная утилита для оптимизации кода;
- содержит много готовых шаблонов (Navigation Drawer, LoginActivity, Master/Detail Flow и etc);
- возможность создания приложений для Android Wear и Android TV.
- встроенная поддержка Google Cloud Platform;
- эмулятор для наглядного просмотра работы приложения;
- встроенное подключение ресурсов Android;
- система контроля версии (Version Control System).

Среда разработки Android Studio построена на Java, с интеллектуальным редактором кода, шаблонами для сервисов Google, интеграцией с GitHub, виртуальными устройствами для всех типов экранов и возможностью создания APK. В целом среда Android Studio понятна и вместе с тем привлекательна. Компания Google убедила разработчиков, что эта среда поможет создавать приложения намного быстрее и эффективнее. Так же компания Google предоставила руководство по переходу на Android Studio с Eclipse. В него входит инструмент для анализа продуктивности Memory Monitor. В Android Studio есть возможность портировать уже существующие Android-приложения, создавая таким образом новый проект приложения. На рисунке 1.3 представлен вид интегрированной среды разработки Android Studio 1.0

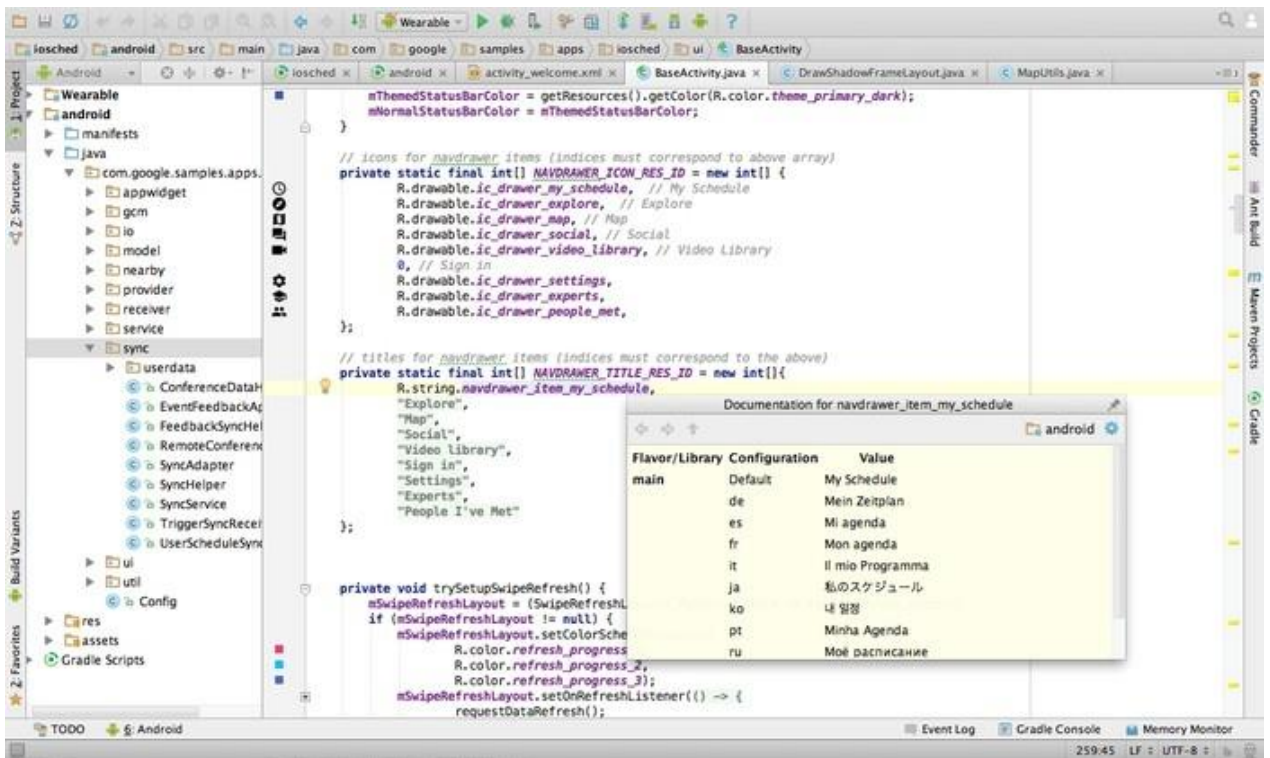


Рисунок 1.2 – Среда разработки Android Studio 1.0

Архитектура проекта в Android Studio значительно отличается от структуры которую использует Eclipse. Структура проекта представлена на рисунке 1.3.

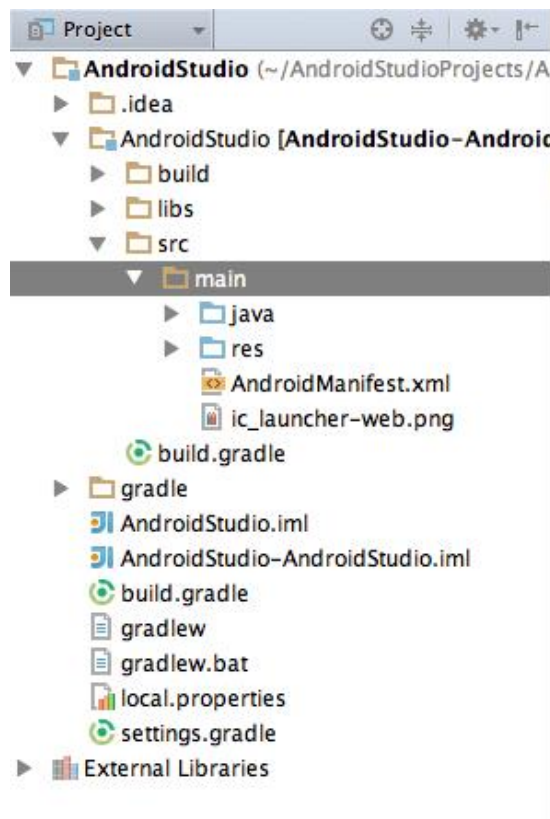


Рисунок 1.3-Структура Android studio

Однако близится финальный релиз Android N. Для разработчиков была выпущена уже 2 версии грядущей операционной системы. Android N содержит множество усовершенствований включая новых API и других различных возможностей. Обновленные функции не будут работать, чтобы они работали разработчикам нужно интегрировать их в приложения, для этого была выпущена Android Studio 2.1. Android Studio 2.1 — новейшая на сегодняшний день среда разработки приложений для Android, и она является доступной для загрузки для ОС Windows, Mac и Linux. В этой версии осуществлена поддержка Android N Preview SDK. Более того, есть возможность опробовать новый компилятор Jack (Java Android Compiler Kit), усовершенствованная поддержка Java 8 и функция Instant Run. Самым значимым новшеством стала поддержка Android N и Android N API.

1.1.2 Виртуальная машина ART

Виртуальная машина исполняет программу стороннего приложения в отдельной среде. Благодаря этому код программы не сможет навредить системным файлам.

В Google Play и других источниках хранятся APK-файлы разработчиков, которые являются не скомпилированными. Эти файлы перед выполнением собирает виртуальная машина. Это позволяет программе большую совместимость, так как в противном случае, программа скомпилированная для устройств на базе Snapdragon, не будет работать на Atom-аппаратах.

Android RunTime— это новая виртуальная машина, которая идет на смену виртуальной машины Dalvik. Она была представлена Google в версии Android 4.4. Основным отличием от старой среды выполнения приложений является применение технологии Ahead-Of-Time (вперед времени). ART компилирует инструкции приложения во время установки. Виртуальная машина не компилирует текущее приложение во время его исполнения, что обеспечивает удобство при использовании многозадачности. В отличие от Dalvik, ART загружает в оперативную память абсолютно все инструкции, поэтому необходим большой объем памяти, но при этом программы открываются быстрее и меньше используется процессор. Google собирается использовать технологию Ahead-Of-Time для борьбы с его главным конкурентом— iOS. Однако минусом этой технологии является долгая продолжительность установки и увеличение размера программы на 20%. ART пока еще является экспериментальным, однако Google предоставил возможность переключиться с Dalvik на ART через скрытые инструменты. Выбор виртуальной машины ART представлен на рисунке 1.4.

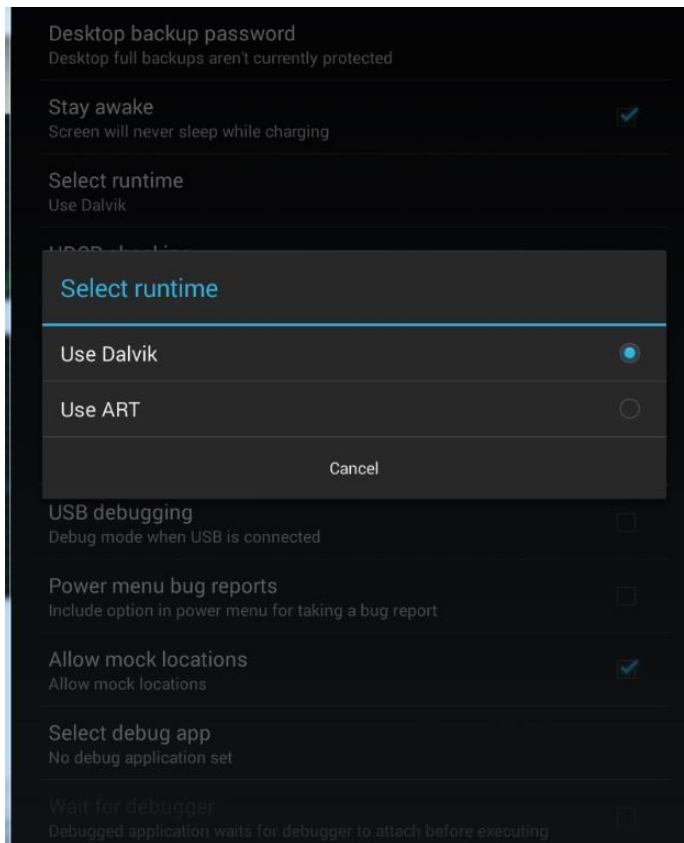


Рисунок 1.4 – Выбор виртуальной машины

2 Инструменты разработки

2.1 Правила и рекомендации для разработки приложений

Каждая компания выдвигает ряд требований разработчикам приложений к созданию приложения для обеспечения работоспособности продукта, а также публикации в Google Play. Помимо всего, имеются некоторые рекомендации по концепции дизайна и организации интерфейса приложений, для того чтобы получить лаконичный, удобный и не выбивающегося из общего стиле системы пользовательского интерфейса (User Interface, UI). Эти принципы проектирования были разработаны группой Android по взаимодействию приложений с пользователями с целью соблюдения интересов пользователей. Для создателей приложений под Android они лежат в основе более подробных рекомендаций по проектированию для конкретных типов устройств. Со всеми ими можно ознакомиться на профильных официальных сайтах компании Google.

2.2 Инструменты для разработки графической оболочки

Чтобы создать графические элементы приложения используется программное обеспечение, предоставленное компанией Adobe. Illustrator CS служит для работы с векторной графикой. Для того чтобы его приобрести нужно :

- 1) Пройти ссылке <http://www.adobe.com/ru/products/illustrator.html>.
- 2) Оформить подписку.

3) Загрузить и пройти пару простых пунктов установки программы. Также есть возможность купить DVD-диска в специализированных магазинах. Первый этап процесса установки изображен на рисунке 2.1.

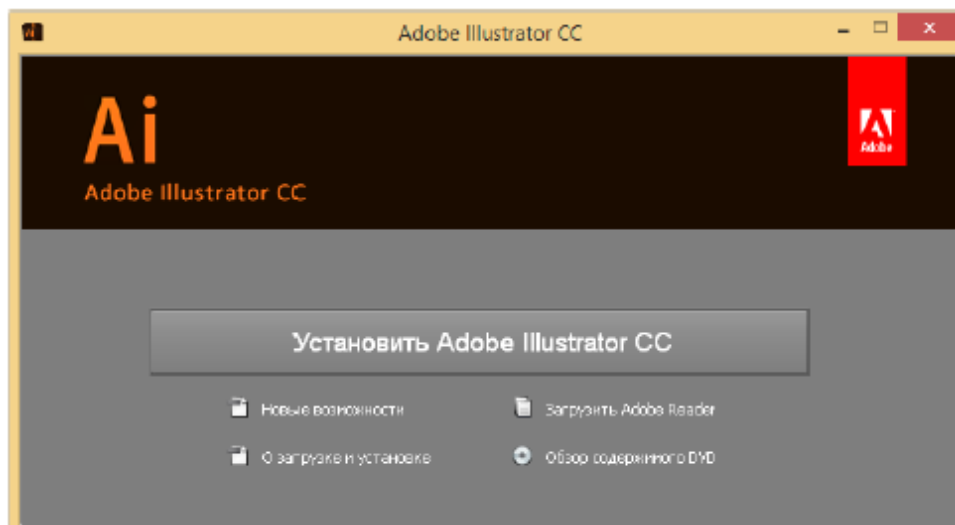


Рисунок 2.1– Первый этап установки ПО

Существует версии Adobe Illustrator CC для 32 и 64 разрядных ОС. При наличии 64х системы, достаточно лишь установка одной версии для ОС. Обычно 64 разрядные системы поддерживают обе версии. Установив одну версию программы сохраняется больше пространства на диске. После определения нужной версии необходимо нажать кнопку «установить». На рисунке 2.2 предствален выбор необходимой версии, следом за ним на рисунке изображен 2.3 интерфейс установленного программного обеспечения.

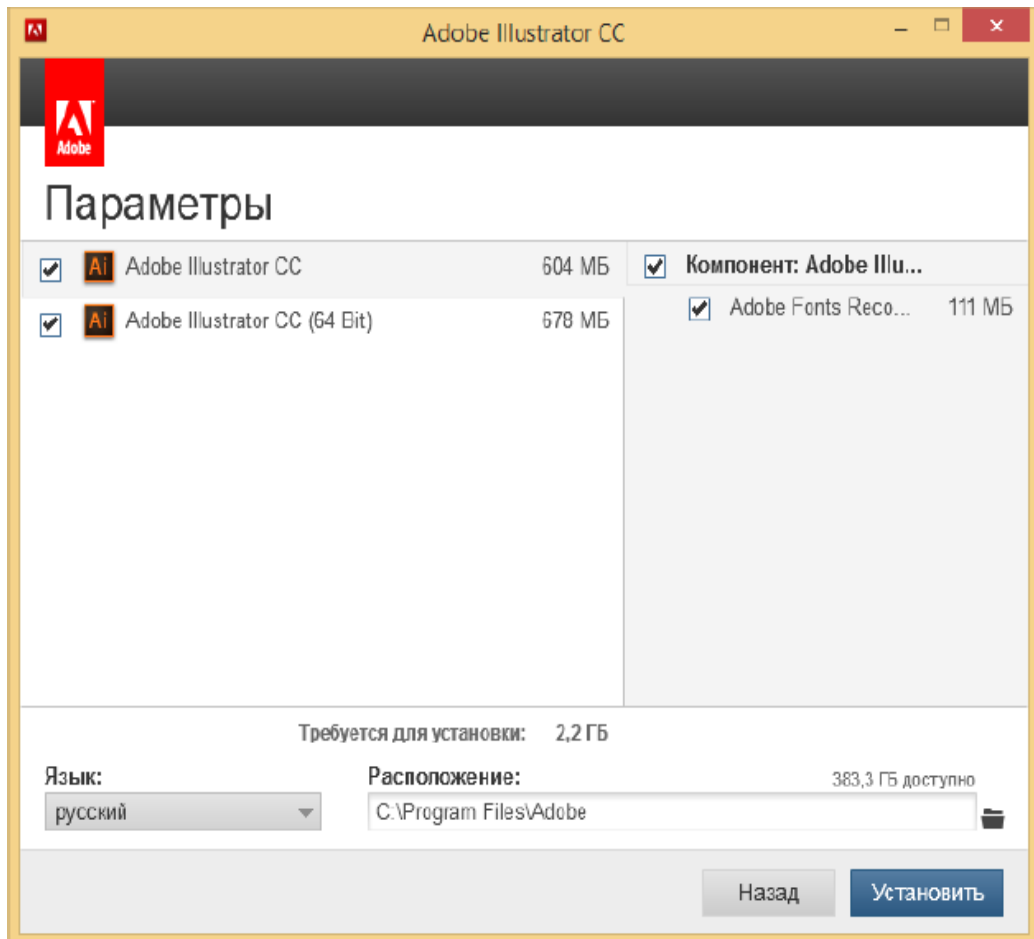


Рисунок 2.3 – Выбор версии программного обеспечения

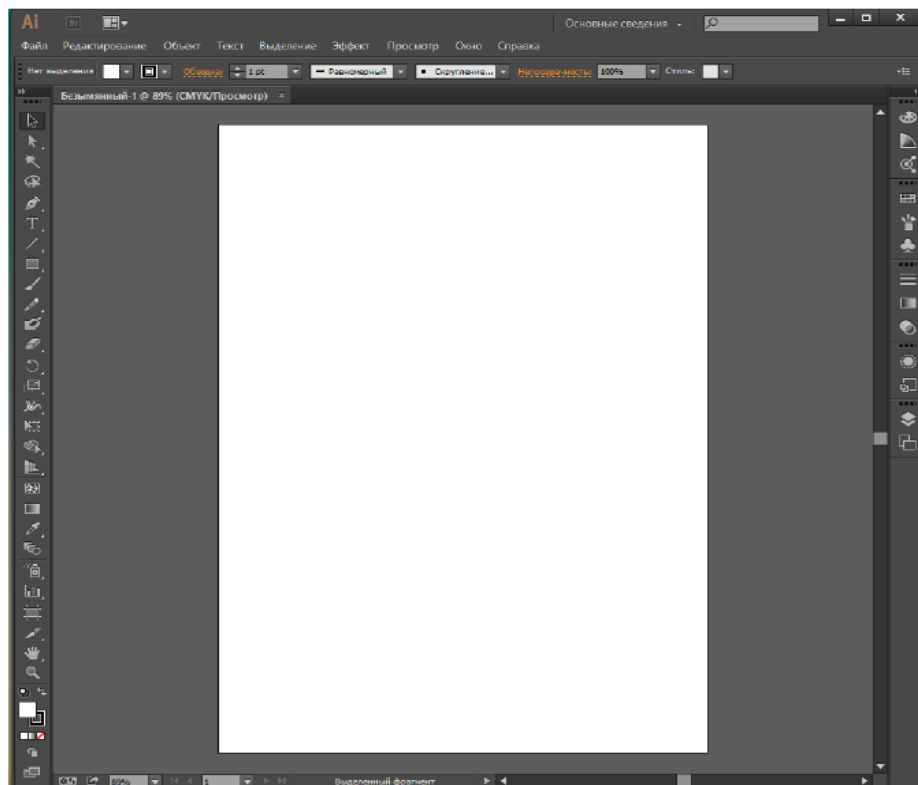


Рисунок 2.3-Интерфейс Adobe Illustrator CC

Adobe Illustrator CC – это одна из новейших редакторов векторных изображений, предоставляющая множество различных возможностей для разработки дизайна. Плюсы в использовании векторной графики:

- масштабируется без потери в разрешении;
- четкие и удобные линии;
- печатается на высоких разрешениях;
- экономный размер файлов;
- подходит для создания различных иллюстраций.

К недостаткам векторной графики относятся плоское отображение рисунков и затруднения с созданием фотореалистичных изображений. Инструментарий Adobe Illustrator CC изображен на рисунке 2.4.

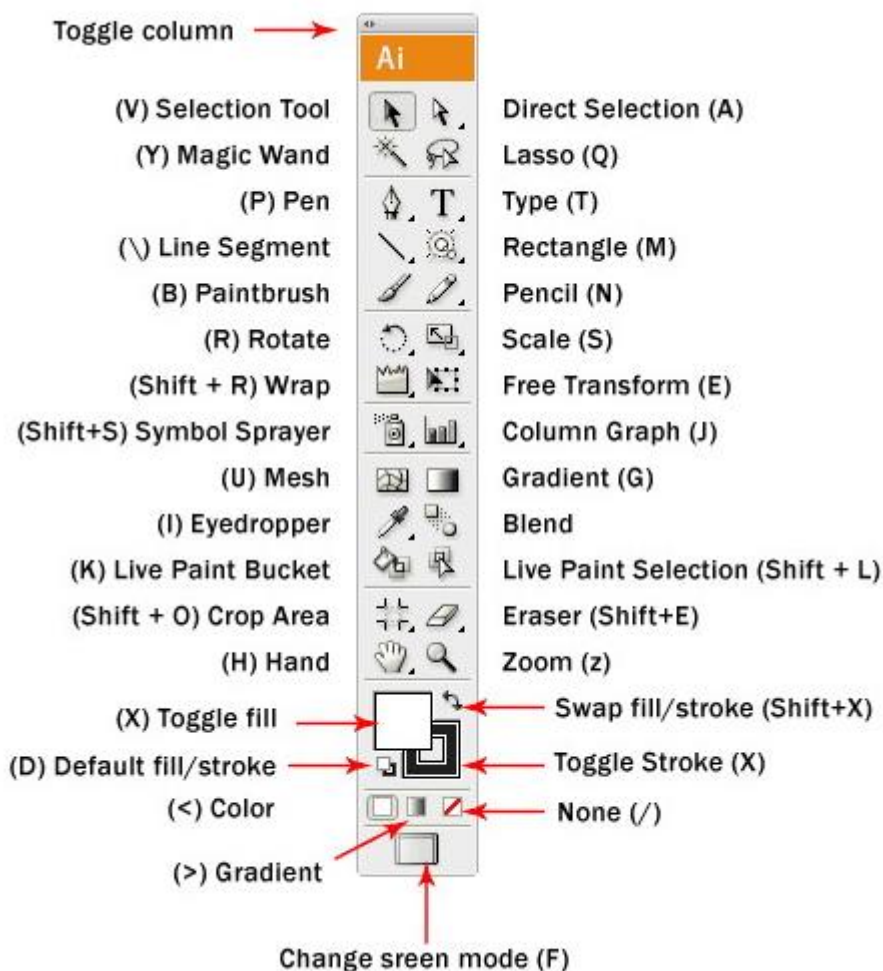


Рисунок 2.4 – Инструментарий Adobe Illustrator CC

Чтобы создать и управлять документами используются такие компоненты различные компоненты интерфейса, такие как палитры, панели и окна. Эти компоненты находятся на рабочем пространстве/среде программы. В принципе пользователь имеет возможность настроить программное обеспечение в соответствии со своими предпочтениями, выбрав себе рабочее пространство из

предложенных или создав свое. Несмотря что вид рабочих пространств разных приложениях немного отличается, процесс работы происходит схожим образом.

Окно «Панель приложения» находится в верхней части и содержит переключатель рабочих сред, меню (только Windows) и другие компоненты управления приложением. При работе Adobe на операционной системе Mac пользователь имеет возможность отобразить или скрыть эту панель, воспользовавшись меню «Окно».

Панель «Инструменты» содержит в себе инструменты создания и форматирования различных изображений, графических объектов, компонентов страниц и т.д. Связанные с этой панелью инструменты располагаются в группах.

На панели «Управление» хранятся настройки инструмента, который выбран в данный момент. В программе Illustrator на панели управления содержатся параметры для текущего выбранного объекта. В Adobe Flash®, Adobe Dreamweaver® и Adobe Fireworks® ее называют инспектором свойств.

На окне документа отображен файл находящийся в работе. На окно документа можно поместить закладки, а в других некоторых случаях, закладки можно объединять в группы и закреплять.

Палитры облегчают отслеживание и изменение обработки объекта. К примеру, панель «Слои», в Adobe Photoshop® временная шкала в Flash, панель «Кисть» в Illustrator и панель «Стили CSS» в Dreamweaver. Палитры тоже можно объединять в группы, собирать в подборки или закреплять.

Фрейм приложения объединяет все компоненты рабочей среды в одно интегрированное окно, позволяющее работать с приложением как с единым целым. При работе с размерами фрейма приложения или его компонентами все элементы внутри него взаимодействуют друг с другом и не перекрываются. Палитры в приложении не теряются во время переключения приложений или при щелчке за пределами приложения. Во время работы с другими приложениями параллельно, можно расположить их рядом на экране или на нескольких мониторах. Если пользователь предпочитает привычный традиционный произвольный интерфейс Mac, то он может отключить фрейм приложения. В Adobe Illustrator® для этого нужно щелкнуть на «Окно» и перейти на «Фрейм приложения», и там можно включить или выключить его.

Поле «Поиск по справке» находится на правой части панели приложения и необходим для поиска в разделе справки и по онлайн-содержимому. Если есть подключение к интернету, то есть возможность получить доступ к содержанию веб-узла справки сообщества. При выполнении поиска в справке без наличия подключения к интернету, в результатах поиска будет отображаться только содержимое справки, которая была установлена вместе с Illustrator. Для поиска нужно ввести имя элемента, о котором нужна информации нажать клавишу Enter. Все разделы, содержащиеся в центре справки сообщества сразу отображаются в отдельном окне браузера.

Строка состояния находится в левой нижней стороне окна иллюстрации. В ней выводится информация о текущем уровне масштаба, использованном инструменте, используемой монтажной области, количество возможных отмен и повторов, цветовой профиль, а также дата, время и состояние обрабатываемого файла.

Изменить тип сведений, отображаемых в строке состояния можно используя параметр в подменю «Показать». Если нужно отобразить текущий файл в Adobe Bridge, то нужно выбрать функцию «Открыть в Bridge».

С помощью Adobe Illustrator Draw можно создавать произвольные векторные дизайны для смартфонов под управлением операционной системы Android и отправлять их в настольное приложение Illustrator, чтобы в дальнейшем можно было его доработать.

В связи с последними обновлениями появилась возможность создания макетов в приложении Comp CC для устройств iPhone и iPad. Можно создать в Comp связанные ресурсы. Созданные ресурсы будут обновляться в Comp при форматировании их в Illustrator и других приложениях Creative Cloud.

2.3 Инструменты разработки кода приложения

Для написания программы в данном проекте будет использоваться язык Java, а среда разработки Android Studio. Для работы в Android Studio необходимо установить Java SE Development Kit (JDK) на систему. Рекомендуется установка новых версий JDK не ниже 7. Для установки JDK необходимо скачать установочный файл на ссылке: <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

Java Development Kit - это программный инструмент для производительной работы с языком Java, который хранит в обширную библиотеку классов Java. Он включает в себя: компилятор *javac*, интерпретатор *java*, программу просмотра апплетов *appletviewer*, отладчик *jdb*, отладчик *jdb* и множество других компонентов. После загрузки исполняемого файла его необходимо запустить и следуя инструкциям, установить данный инструментарий. Затем нужно загрузить файл установки Android Studio с официального сайта Android специально для разработчиков. Установочный файл можно загрузить пройдя по адресу: <http://developer.android.com/intl/ru/sdk/installing/studio.html>

Здесь нужно выбрать версию подходящую для операционной системы и затем после загрузки установочного файла, открыть программу следуя его инструкциям. Одним из преимуществ является то, что загрузка Android SDK для работы в Android Studio не нужна, потому что все необходимые компоненты уже включены в комплект установки приложения.

После установки при запуске приложения появится экран приветствия. Экран запуска представлен на рисунке 2.4.

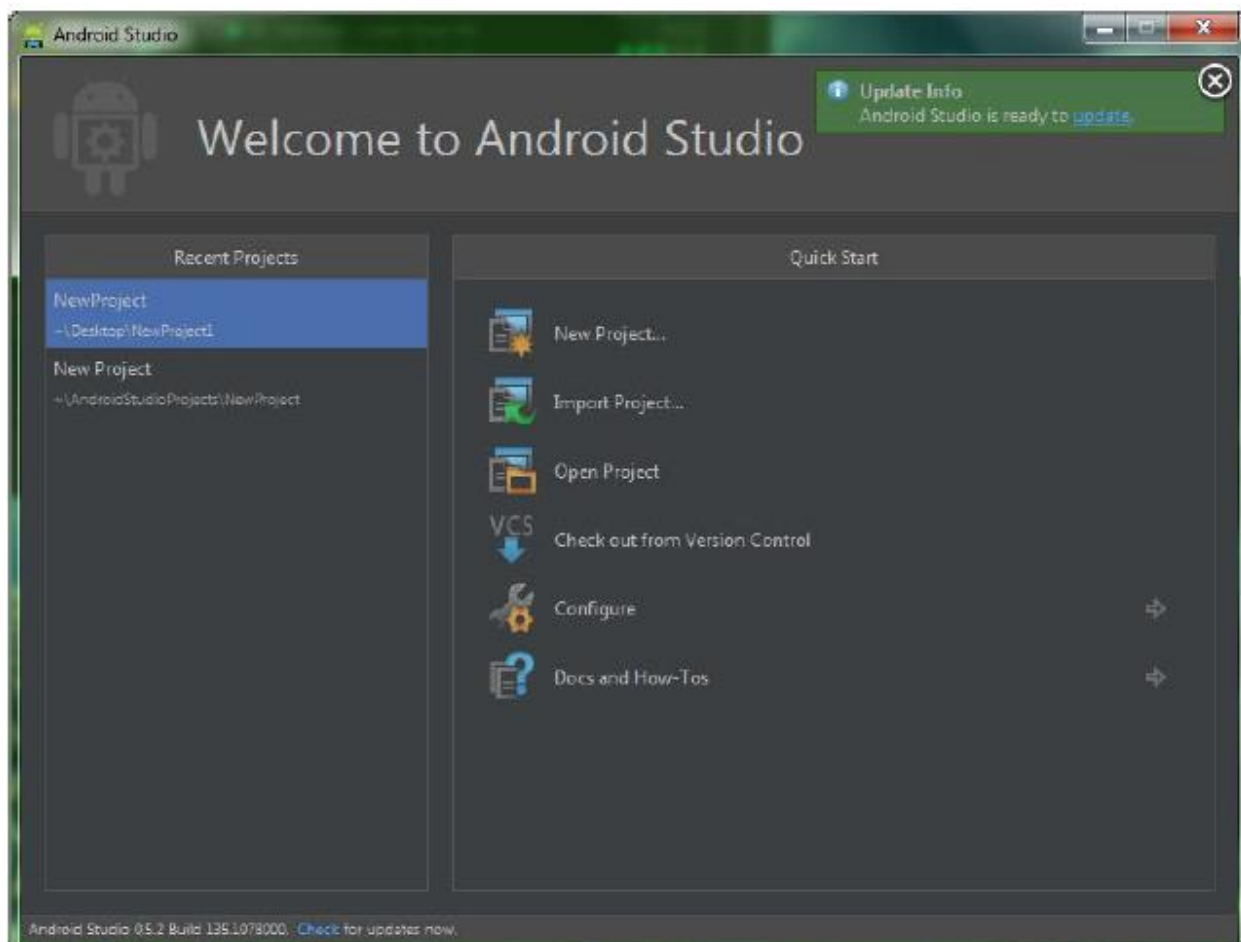


Рисунок 2.4 – Экран запуска Android Studio

Здесь предоставляется выбор для :

- Создания нового проекта.
- Импортирования готового проекта в программу.
- Открытия уже существующего проекта.
- Проверки версии ПО.
- Изменения конфигураций среды.
- Получения справки и помощи в использовании среды.

Чтобы начать работу нужно создать новый, для этого необходимо выбрать пункт «New Project...» и за ним последует окно ввода параметров, которые нужны для создания проекта. Окно создания проекта представлено на рисунке 2.5.

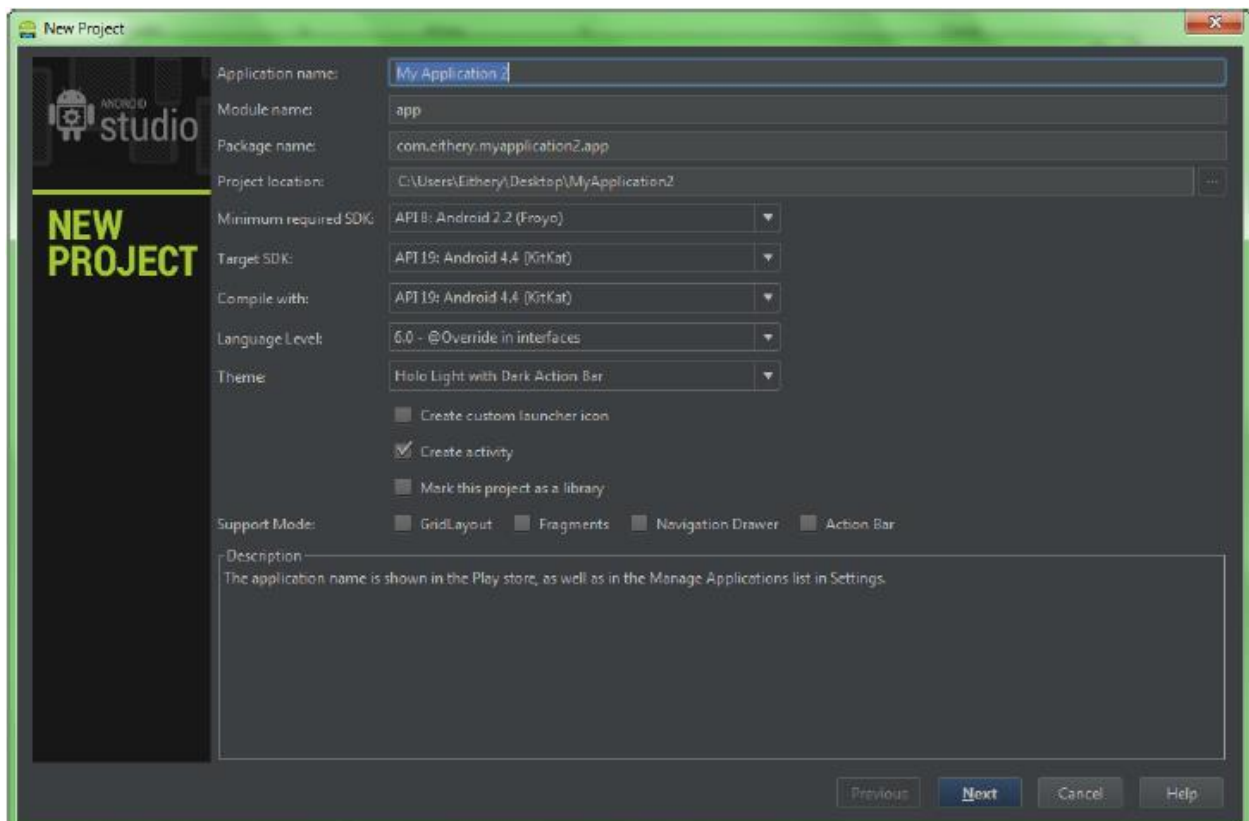


Рисунок 2.5 – Окно создания проекта Android Studio

На данном окне нужно ввести название приложения, модуля, назвать файлы для приложения, а также указать место хранения с данными проекта. Также можно выбрать минимальную версию операционной системы, которая поддерживается приложением и основной, на которую приложение ориентируется. Помимо этого, есть возможность выбора уровня языка. Так же доступен выбор удобной темы приложения для разработчиков, светлой или темной. Если потребуется, можно отметить бокс выбора «Create custom launcher icon»- оно служит для создания иконки создаваемого приложения непосредственно в Android Studio. И последним является возможность выбора «Support Mode».

После нажатия кнопки «next», так как было выбрано «Create custom launcher icon», появится окно для создания и редактирования иконки приложения. Окно для редактирования представлено на рисунке 2.6. Данный редактор удовлетворяет потребности всех разработчиков, так как с помощью него легко и быстро создать иконку для разных плотностей экрана и есть возможность сразу же увидеть результат.

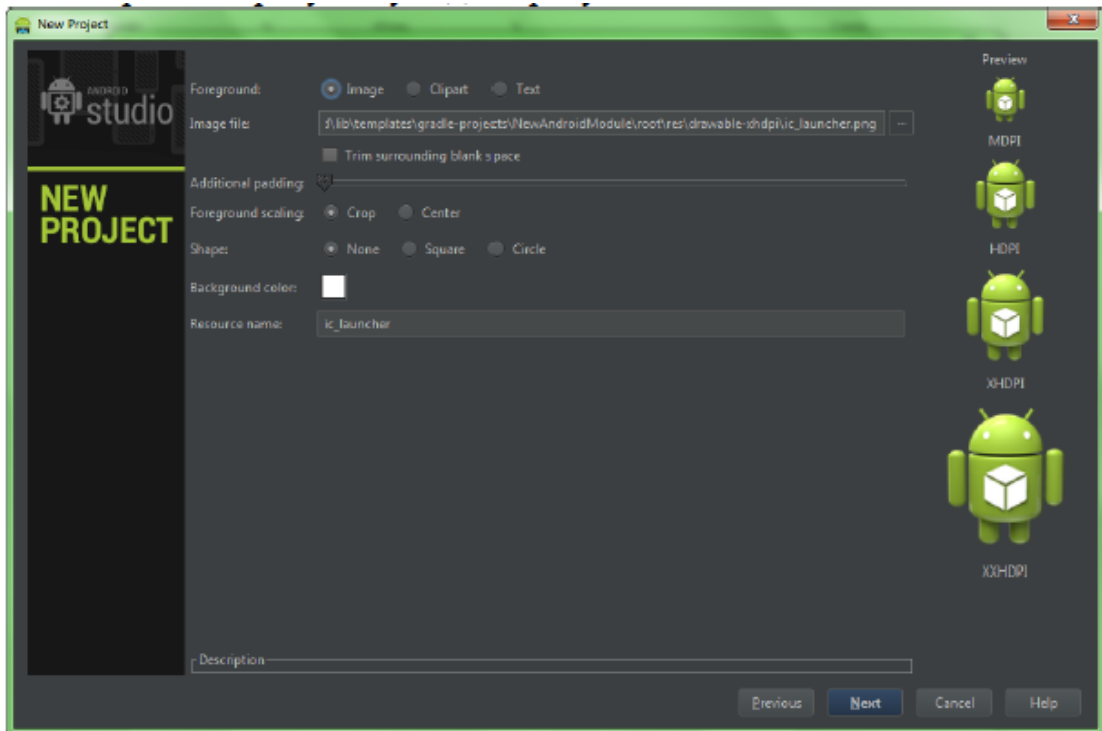


Рисунок 2.6 – Окно создания значка приложения

После завершения создания и редактирования иконки, идет переход на экран выбора одной из трех activity. Окно выбора activity изображено на рисунке 2.7.

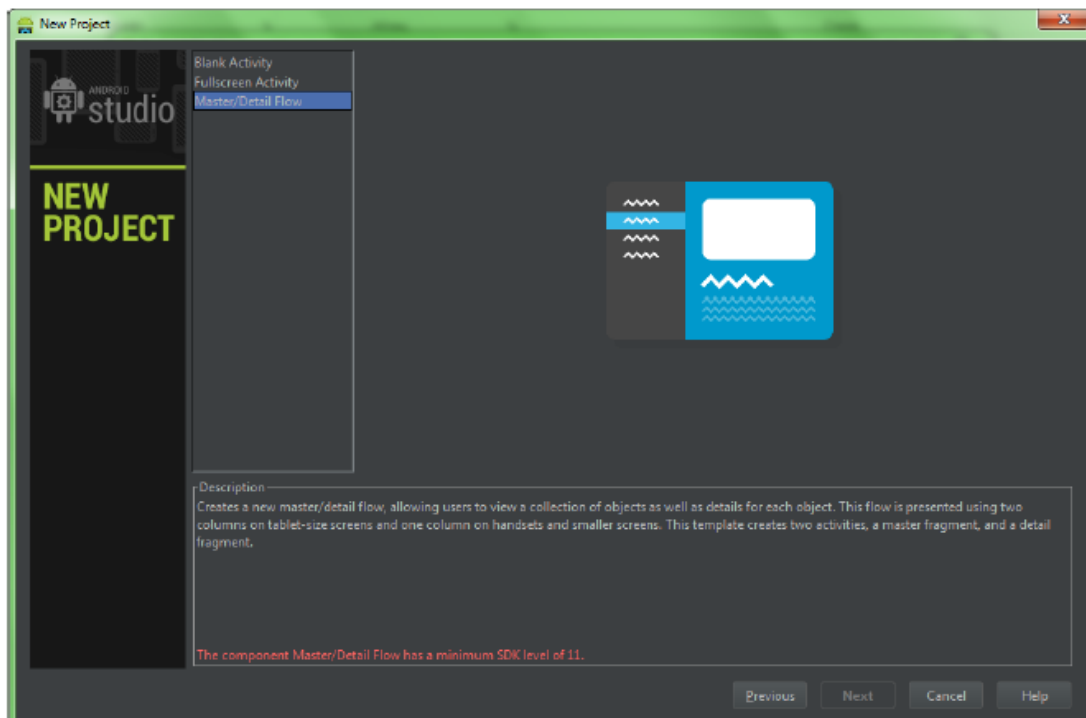


Рисунок 2.7 – Окно выбора Activity

На экране предоставлен выбор из трёх Activity:

- 1) Blank Activity.
- 2) Fullscreen Activity.
- 3) Master/Detail Activity.

После выбора базовой Activity, нужно указать ее имя и имя её, а также имя первого фрагмента, поскольку была выбрана поддержка фрагментов при создании проекта. Выбор типа навигации по Activity и ввода ее параметров изображено на рисунке 2.8.

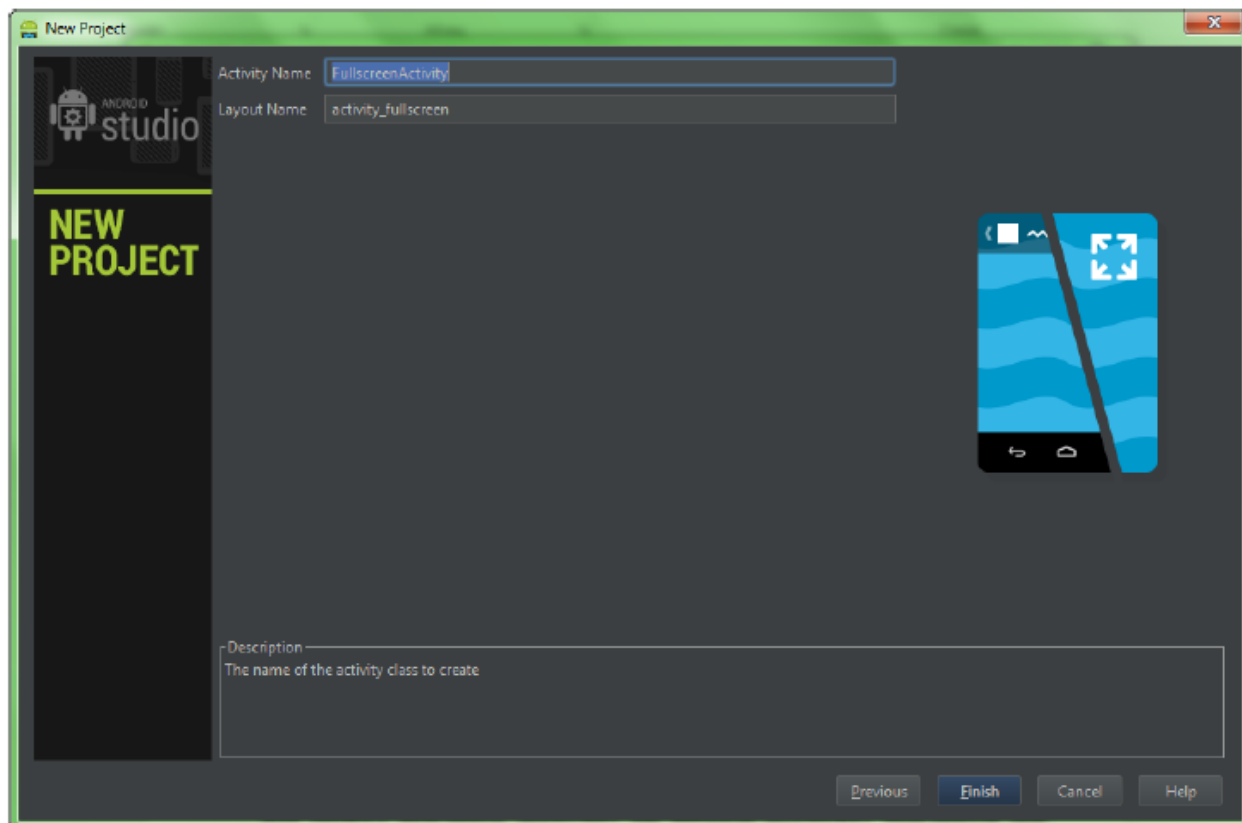


Рисунок 2.8 – Ввод параметров Activity

После чего открывается окно IDE. Первое на что бросается взгляд— это структура проекта, так как она сильно отличается от среды Eclipse.

Хотя остались привычные папки src и res, но при этом папка res теперь находится внутри папки src. А так же появилась новая папка java, и в ней теперь находятся пакеты и классы. Новая структура проекта Android Studio образовалась из-за новой системы сборки проекта – Gradle. Сборка позволяет управлять зависимостями в созданном проекте и добавляет внешние библиотеки, которые часто обновляются. Плюсом является то, что при импорте проектов с прежней или более сложной структурой – всё работает очень быстро и не тормозит. На рисунке 2.9 представлено рабочее окно среды разработки.

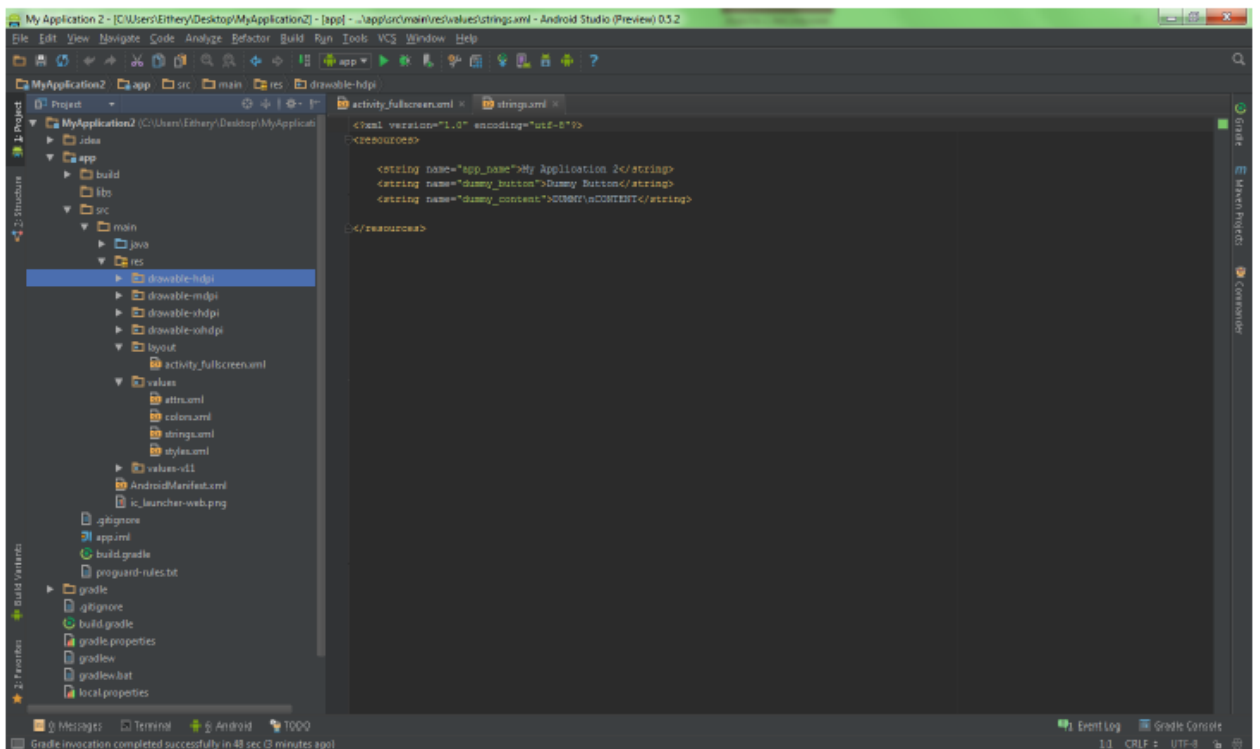


Рисунок 2.9-Рабочее окно Android Studio

В среде Android Studio есть специальный Xml редактор который изображен на рисунке 2.10

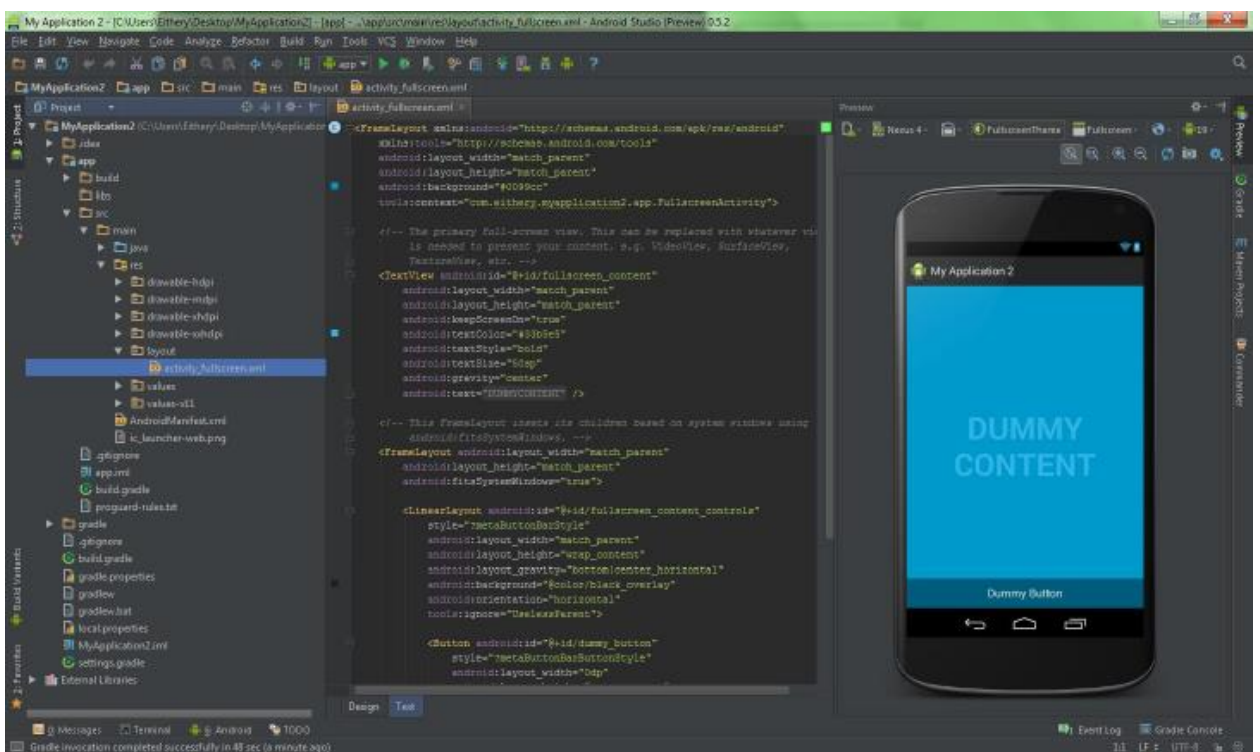


Рисунок 2.10 –Xml редактор

В описании xml редактора есть несколько особенностей:

- При редактировании xml в текстовом режиме можно просматривать превью;
- Редактирование цветов и рисунков, которые применяются в layout'e сразу меняются на макете, что помогает понять какой ресурс используется;
- При выборе конкретного ресурса, его содержимое отображается в JavaDoc'ах;
- Ресурсы взятые из dimens сразу показываются значениями, так же при наведении сразу можно определить вид используемого ресурса;
- Xml парсер позволяет приложению принимать события от парсера в отличие от SAX-парсеров, которые автоматом передают события обработчику.

Совершенствованная интеграция с Android компонентами. При создании нового класса ,переходим внутрь пакета, куда разработчик хочет разместить созданный класс, и можно нажать сочетание клавиш alt-insert. Следует заметить что для тех кто привык работать с Eclipse что hotkey's в Android studio иногда сложно запомнить, но он несет в себе гораздо более сложный ,но очень гибкий функционал.

Среда предлагает на выбор несколько объектов:

- Java Class –это Java Component. Он создает один из основных элементов Java:Interface, Class, Annotation, Enum, и Singleton.
- Module – создает модуль. Модуль – это вспомогательный проект в среде Android Studio. Модулями являются внешние библиотечные проекты;
- File – файл с различным разрешением ,к примеру txt.
- Package – пакетсоздаваемого приложения.

Рассмотрим как создаются компоненты в Android Studio. На рисунок 2.11 изображено окно для создания элемента программы.

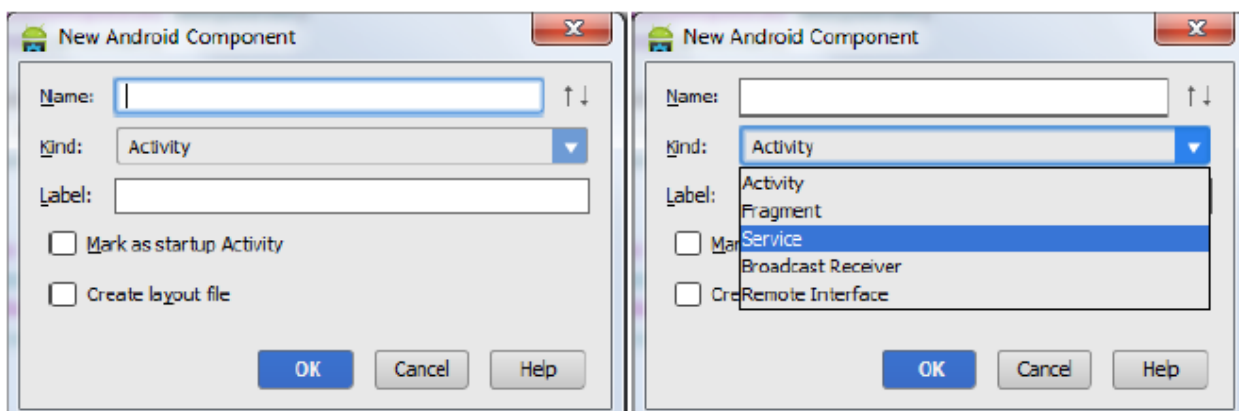


Рисунок 2.11 – Окно создания компонента

Activity можно сказать окно, которые используя готовые шаблоны, и регистрирует Activity в Манифесте. Android Component –это компонент, создает ключевые компоненты нашего Android приложения, желательно создать layout для него, и обозначить Activity стартовой. Если появляется необходимость то он по умолчанию регистрирует компонент в Манифесте. Package-info.java – это

файл содержит информацию про пакет. HTML File –нужен для создания html файла.

Следует отметить достоинства текстового редактора среды:

- Интеллектуальный анализ ошибок в коде;
- Содержит Android Sources;
- Создавать тесты на классы в 2 клика.

3 Этапы разработки

3.1 Обзор и анализ аналогов приложений

Тематикой приложения будет приложения для здоровья. В связи последних событий, люди очень остро нуждаются в информировании о здоровом питании, это обусловлено тем, что сейчас распространена вредная пища и фастфуд и из-за этого у людей часто возникают проблемы с желудком. Здоровое питание является одной из самых актуальных тем. Это важно не только для улучшения здоровья, но и для поддержки здорового состояния организма. Еще пару десятилетий тому назад никто и представить не мог, что меню и рекомендации для питания может помещаться в любом кармане и находится под рукой. Сегодня это стало реальностью, благодаря развитию мобильных технологий. Теперь в решении многих вопросов может помочь разработка приложения «Правильное питание».

На рынке мобильных приложений существует множество приложений на данную тематику, для этого нам необходимо проанализировать аналогичные приложения в GooglePlay. В магазине было найдено несколько аналогичных приложения, которые показаны на рисунке 3.1.

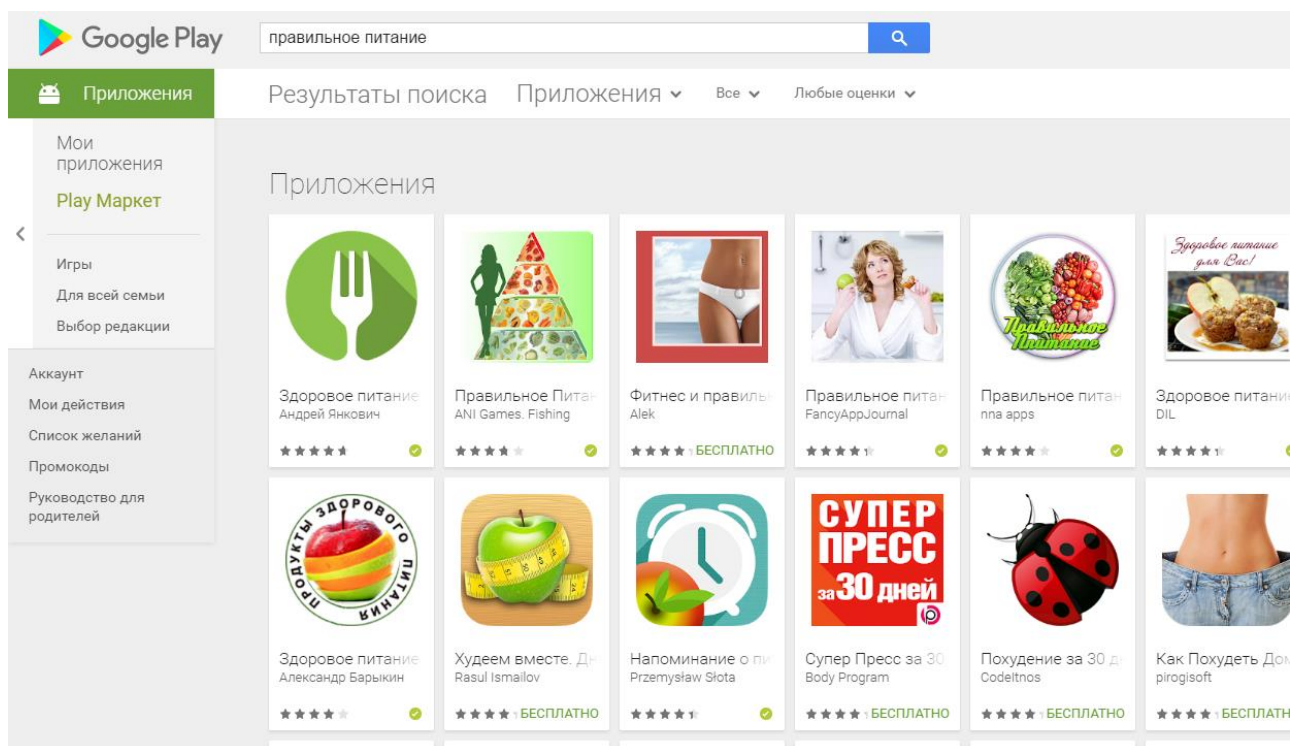


Рисунок 3.1–Аналоги

Основным минусом данных приложений является отсутствие совмещенности счетчика и меню. На рисунке 3.3 изображена страничка установки приложения «Правильное питание».

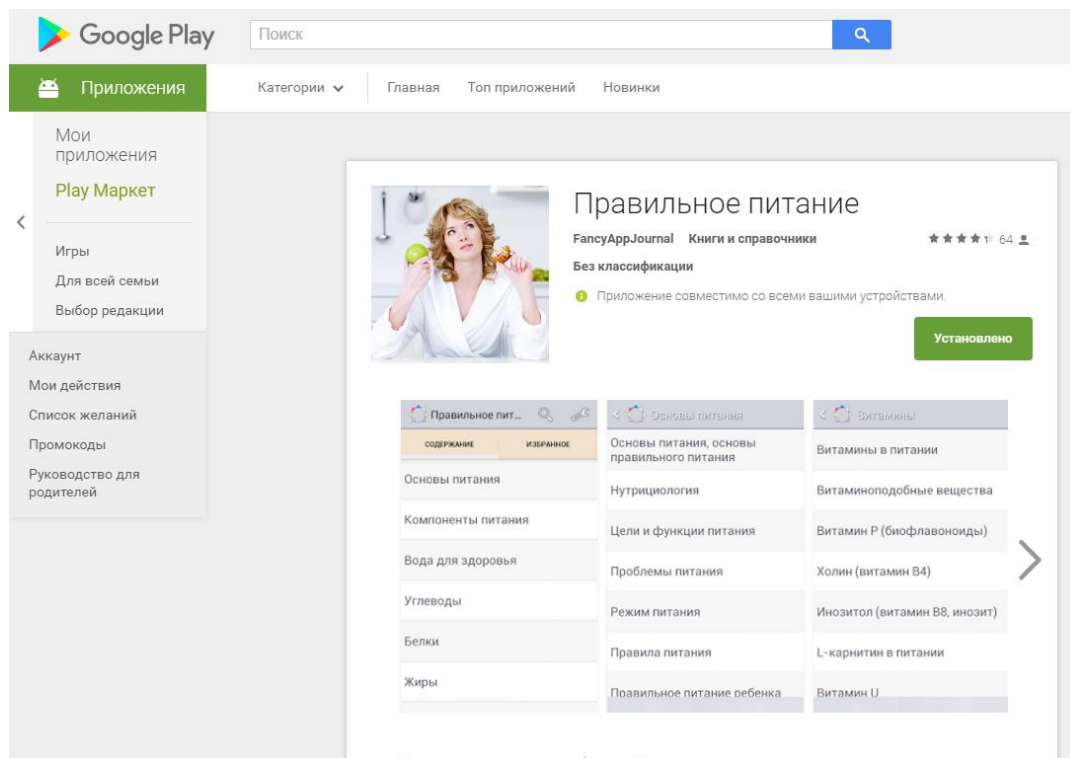


Рисунок 3.2 - Страница установки приложения «Правильное питание»

3.2 Проектирование

Приложение «Правильное питание» содержит в себе счетчик калорий, который будет высчитывать суточную норму для одного человека. Так же в данном приложении расписано меню питания на каждый день.

Данное приложение красочный и функциональный интерфейс с применением единого стиля Material Design. Material Design — единая логика построения приложения и его внешнего. Он унифицирует большинство продуктов Google с целью максимально удобного и интуитивного восприятия пользователей приложения.

Интерфейс приложения должен сочетаться с другими компонентами оформления внешнего вида операционной системы, чтобы у пользователей не было проблем с его пониманием. Дизайн приложения должен следовать установленным стандартам и быть гибким чтобы отвечать требованиям выпускаемых платформ.

Используя все требования и стандарты, реализуется интерфейс мобильного приложения «Правильное питание».

Интерфейс приложения состоит из шести частей:

- Главное меню, в котором находится счетчик.
- Меню здоровое питание.
- Пирамида правильного питания.
- Разрешенные продукты.
- Запрещенные продукты.
- Меню на день.

На главном окне находится счетчик для расчета суточной нормы калорий. В главном окне предоставлен интерфейс в котором пользователям нужно ввести рост, вес, выбрать активность и пол. При нажатии на кнопку «Суточная норма калорий» выдается результат о суточной норме потребления калорий, который высчитывается с помощью формулы. Так же в дополнении есть выдвижное меню, в котором хранятся настройки и информация о разработчике. В настройках можно выбрать оформление приложения. На выбор есть 3 цвета: фиолетовый, голубой и розовый. Главное окно для ввода персональных данных представлено на рисунке 3.1

Правильное питание

Я женщина Я мужчина

Ваш возраст
22

Ваш рост
160

Закреть

Суточная норма калорий
1498.8ккал/день

Рисунок 3.3 –Окно главного меню

Вывод высчитанной нормы потребления калорий изображен на рисунке 3.2.



Правильное питание

Я женщина Я мужчина

Ваш возраст
22

Ваш рост
160

Закреть

Суточная норма калорий
1498.8ккал/день

Рисунок 3.4-Вывод суточной нормы калорий

Формула для расчета нормы для мужчин и женщин немного отличается.
Формула для женщин:

$$10 \times \text{вес (кг)} + 6,25 \times \text{рост (см)} - 5 \times \text{возраст (лет)} - 161; \quad ()$$

Формула для мужчин:

$$10 \times \text{вес (кг)} + 6,25 \times \text{высоту (см)} - 5 \times \text{возраст (лет)} + 5;$$

При переходе в меню «Здоровое питание» предоставлена общая информация о питании и кнопки для перехода в другие разделы. Все управление в приложении производится с помощью пальцев руки. Меню «Здоровое питание» предоставлено на рисунке 3.2.

Питание человека –самый важный фактор, напрямую влияющих на здоровье человека. Неправильное питание приводит к нарушению функций как отдельных органов человека так и организма в целом. Пагубно влияет и неполноценная по составу пища, и недостаток пищи и ее переизбыток. Именно поэтому питанию необходимо уделять внимание и прикладывать необходимые усилия для того чтобы оно было полноценным!

Пирамида правильного питания

Разрешенные продукты

Запрещенные продукты

Меню на день



Рисунок 3.5 – Меню «Здоровое питание»

При переходе в Activity «Пирамида правильного питания» выдается текстовая информация. В ней пирамида в которой показана иерархия продуктов ,а так же описаны принципы здорового питания. Окно «Пирамида правильного питания» показана на рисунке 3.3.



Схематическое изображение основополагающих принципов здорового питания можно увидеть, посмотрев на пирамиду питания, разработанную зарубежными диетологами и одобренную российскими специалистами. В пирамиде питания представлены не конкретные продукты, а пять больших групп продуктов, что позволяет разнообразить свой рацион питания и выбирать те продукты, которые вам больше нравятся или подходят для вашего образа

Рисунок 3.6 –Activity «Пирамида правильного питания»

В Activity «Запрещенные продукты» изображенной на рисунке 3.7 так же выдает текстовую информацию. В ней описаны продукты, которые нежелательно употреблять часто, так как при избыточном потреблении они приводят к ожирению организма.

← Запрещенные продукты



Запрещенные продукты

- сахар и сладости, произведенные с его помощью; хлебобулочные изделия из белой муки (исключение мука твердых сортов пшеницы);
- алкогольные и газированные напитки;
- жареные блюда;
- фаст-фуд;
- пища, содержащая синтетические консерванты (копчености, консервы, колбасы и прочее);

Рисунок 3.7 –Activity «Запрещенные продукты»

Следующее Activity «Разрешенные продукты» содержит информацию о полезных продуктах, которые содержатся белки, углеводы и жиры. Activity «Разрешенные продукты» представлена на рисунке 3.8.

← Разрешенные продукты

Белки



Молоко, Изолированный соевый белок, Яйца, Говядина, Гороховая мука, Фасоль консервированная, Чечевица, Арахис, Пшеница, Куриная грудка.

Углеводы



Рисунок 3.5–Activity «Разрешенные продукты»

В разделе «Меню на день» предоставлено готовое меню, в котором описаны рецепты блюд.

← Меню на день

Яичница или омлет.



Приготовленные из 2 яиц на сковороде с антипригарным покрытием, смазанной растительным маслом. Дополнение – овощи: например, помидоры и болгарский перец. Также подойдет горошек или фасоль – около 2 ст. л. на порцию.

Творог



Рисунок 3.6 –Activity «Меню на день»

В главном меню содержится MenuBar, который изображен на рисунке 3.4. При нажатии на иконку MenuBar, появляется выдвигаемое меню с правой стороны, в котором есть Settings и About.

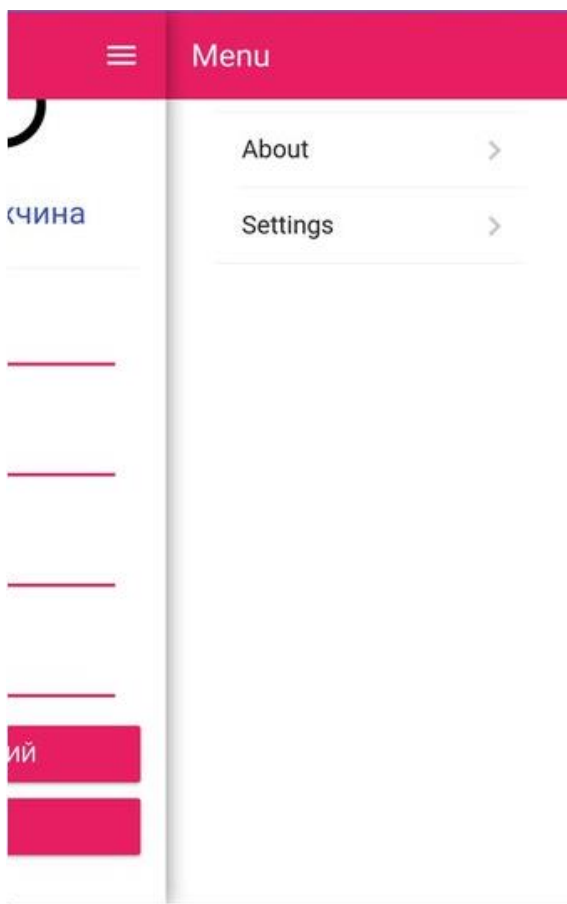


Рисунок 3.7 –MenuBar

При переходе в раздел About появляется информация о разработчике приложения и цели разработки приложения. А так же советы по применению данного приложения. Окно About представлен на рисунке 3.6.

Здравствуйте!
Я Джаныбекова Салтанат Талгатбековна. Студентка Алматинского Университета энергетики и связи. Кругом все завалено всяческими предложениями «быстро похудеть», но, пожалуй, таких «диет» просто не существует. Я являюсь яростным противником диет, голодания и всяческих разгрузочных дней, потому что это может очень отрицательно сказаться на Вашем здоровье. Ваша задача состоит лишь в коррекции пищевых привычек и контроль объема потребляемой пищи. Также значительно сократят путь к желанным цифрам на весах физические упражнения (даже 5 минут в день) и позитивный настрой. Я нашла программы для своего смартфона, которые помогли мне в этом процессе. Но в целом я осталась недовольна этими программами, потому что в основном подход был чисто формальным и математическим, а я за время своей борьбы, получила огромный опыт, которым мне захотелось поделиться. И поэтому я решила

Рисунок 3.8-Окно About

В разделе Settings можно изменить тему интерфейса приложения. Здесь предоставлен выбор из трех цветов: фиолетовый, голубой и розовый. Под пунктом «По умолчанию» подразумевается фиолетовый цвет. Возможность редактирования темы оформления предоставлена на рисунке 3.9.

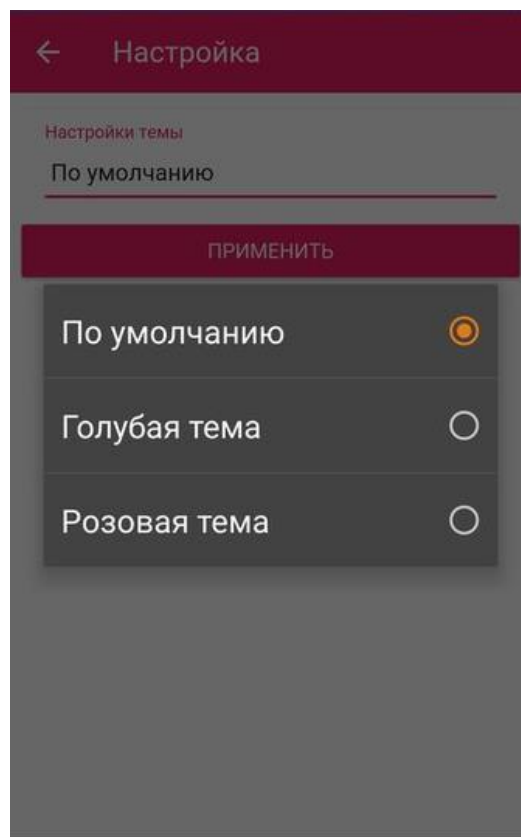


Рисунок 3.9-Настройки

4.2 Разработка приложения

Структура проекта приложения состоит из несколько папок:

- src – хранится исходный код на языке Java, в нем находится главный файл для работы и созданные классы;

- gen – сгенерированные файлы в ходе компиляции;

- res – ресурсы. В нем содержится несколько подкаталогов:

1. res/drawable-*dpi – хранятся ресурсы, которые необходимы для различных расширений экрана. Внутри этой содержится файл ic_launcher.png, который является иконкой приложения.

2. res/layout – в нем хранятся xml-файлы, которые файлы в нем определяют внешний вид форм и компонентов форм. После завершения создания проекта, появляются файлы fragment_main.xml. res/menu , activity_main.xml.

3. res/values – в ней содержатся различные строковые ресурсы, ресурсы цветов, стили, которые используются а разработке приложения. К тому же есть папки values- w820dp, values-v11 которые предназначены для планшетов Android 3.0, values-14 предназначенные для разных девайсов.

4.2.1 Создание и реализация форм

Для разработки форм в меню был создан файл activity_main.xml в каталоге res/layout.

```
<?xml version="1.0" encoding="utf-8"?>
<widget xmlns:="null" xmlns:cdv="http://cordova.apache.org/ns/1.0"
id="com.foodcalc.kz" version="0.0.1">
<preference name="loglevel" value="DEBUG" />
<feature name="Whitelist">
<param name="android-package"
value="org.apache.cordova.whitelist.WhitelistPlugin" />
<param name="onload" value="true" />
</feature>
<preference name="webView" value="org.crosswalk.engine.XWalkWebViewEngine" />
<preference default="17+" name="xwalkVersion" />
<preference default="--disable-pull-to-refresh-effect" name="xwalkCommandLine"
/>
<preference default="embedded" name="xwalkMode" />
<preference default="true" name="xwalkMultipleApk" />
<feature name="SocialSharing">
<param name="android-package" value="nl.xservices.plugins.SocialSharing" />
</feature>
<allow-intent href="market:*" />
<name />
<description />
<author email="dev@cordova.apache.org" href="http://cordova.io" />
<content src="index.html" />
<access origin="*" />
<allow-intent href="http://*/*" />
<allow-intent href="https://*/*" />
<allow-intent href="tel:*" />
<allow-intent href="sms:*" />
<allow-intent href="mailto:*" />
<allow-intent href="geo:*" />
<preference name="xwalkVersion" value="17+" />
<preference name="xwalkCommandLine" value="--disable-pull-to-refresh-effect" />
<preference name="xwalkMode" value="embedded" />
```

```

<preference name="xwalkMultipleApk" value="true" />
</widget>
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:background="res/drawable/autofill_chip_inset.xml"
    android:paddingLeft="8dp" android:paddingRight="8dp" android:layout_width="-2"
    android:layout_height="48dp">
    <TextView android:textSize="14sp" android:textColor="333333"
        android:ellipsize="0" android:gravity="16" android:id="false"
        android:paddingLeft="4dp" android:paddingRight="4dp"
        android:fadingEdgeLength="16dp" android:layout_width="-2"
        android:layout_height="-1" android:minWidth="32dp" android:singleLine="true"
        android:drawablePadding="4dp" android:requiresFadingEdge="4096"
        android:fontFamily="sans-serif-medium" />
    <TextView android:textSize="14sp" android:textColor="646464"
        android:ellipsize="0" android:gravity="16" android:id="false"
        android:paddingRight="4dp" android:visibility="2"
        android:fadingEdgeLength="16dp" android:layout_width="-2"
        android:layout_height="-1" android:singleLine="true"
        android:requiresFadingEdge="4096" />
</LinearLayout>

```

Форма отображения элемента управления выдвигаемого меню реализовывалась в файле select_action_menu.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:icon="{2:16843646}" android:id="false"
        android:title="resourceId:0x104000d" android:alphabeticShortcut="a"
        android:showAsAction="6" />
    <item android:icon="{2:16843537}" android:id="false"
        android:title="resourceId:0x1040003" android:alphabeticShortcut="x"
        android:showAsAction="6" />
    <item android:icon="{2:16843538}" android:id="false"
        android:title="resourceId:0x1040001" android:alphabeticShortcut="c"
        android:showAsAction="6" />
    <item android:icon="{2:16843539}" android:id="false"
        android:title="resourceId:0x104000b" android:alphabeticShortcut="v"
        android:showAsAction="6" />
    <item android:id="false" android:title="Share" android:showAsAction="6"
        style="@style/SelectActionMenuShare" />
    <item android:id="false" android:title="Web search" android:showAsAction="6"
        style="@style/SelectActionMenuWebSearch" />
</menu>

```

При создании интерфейса приложения используют графический редактор дизайна. Пример XML файла в графическом редакторе изображён на рисунке 4.1

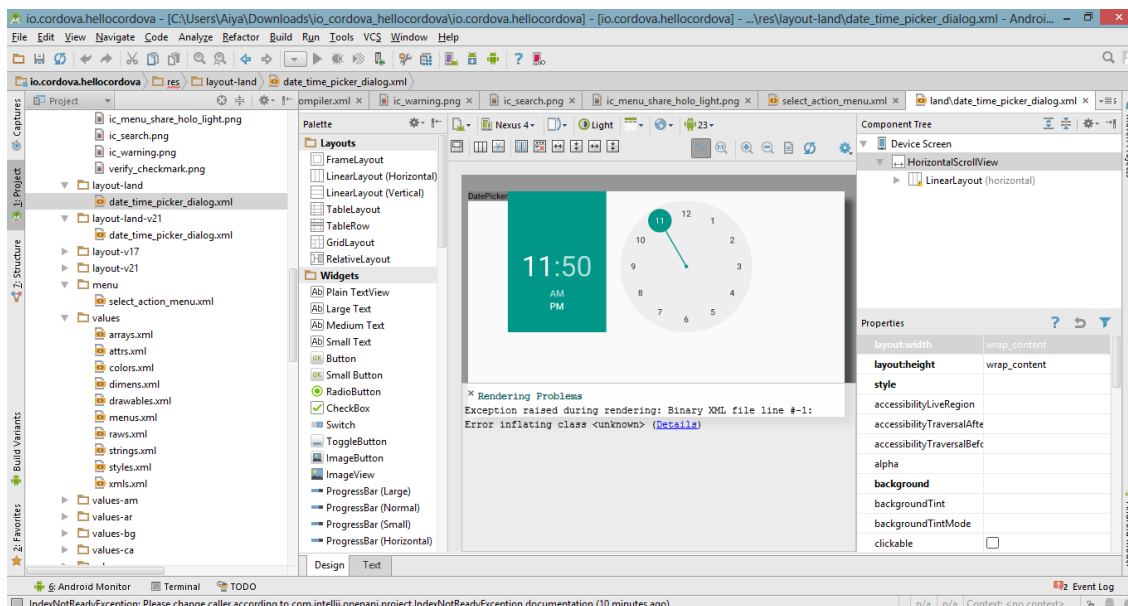


Рисунок 3.6 – Графическое изображение XML файла

3.5 Создание и реализация классов

В процессе разработки были созданы и реализованы множество классов:

- MainActivity – класс в котором описано главное окно;
- SocialSharing – в нем реализуется функция «Поделиться»;
- NavigationParams – класс для навигации;
- BuildConfig – класс для функции создание конфигураций ;
- CommandLine-класс для описания функций;
- ViewConfiguration – класс для функции «Настройки»;
- ColorPickerDialog – класс для настройки темы;
- Style-класс для описания стилей.

4.2 Трудовые ресурсы, используемые в работе

В работе над данным проектом задействованы самые необходимые сотрудники:

- дизайнер –Создание красочного и удобного интерфейса;
- инженер-разработчик – создание алгоритмов, сопровождение и инструктаж персонала сопровождения;
- программист-реализация функционала и самого приложения.

Количество сотрудников, задействованных в проекте, и их месячная заработная плата представлено в таблице 4.2.

Таблица 4.2 – Информация о сотрудниках

Должность	Количество	Зарботная плата в месяц
Дизайнер	1	25 000
Инженер-разработчик	1	50 000
Программист	1	50 000
Итого	3	120 000

4.2 Техническое оборудование, использованное при разработке

Характеристики оборудования, используемого в работе, а также его стоимость описана в таблице 4.3

Таблица 4.3 – Перечень оборудования

Название оборудования	Характеристики	Количество	Стоимость за единицу
Ноутбук Acer TravelMate 7750G-32314G50Mnss	Intel Core i3 3230M, 6 Gb DDR3, 1000 Gb HDD, Radeon HD 7670M	1	119000
Ноутбук Acer v5-552g-10578g50akk	Intel Core i5-4200U, 8 Gb DDR3, 1000 Gb HDD, GeForce GT 740M	1	125000
Смартфон Sony Xperia Z5	ОЗУ: 3 ГБ Флеш-память: до 16 ГБ Четырехъядерный процессор Qualcomm MSM8974AB с частотой 2,3 ГГц	1	250990
Итого		2	494990

4.2.1 Технически-программное обеспечение, используемое в работе

При разработке приложения используется следующее программное

обеспечение:

- OS Windows 8.1 – операционная система;
 - Android Studio – интегрированная среда для разработки;
- Программное обеспечение, использованное в работе и его стоимость приведены в таблице 4.4.

Таблица 4.4 – Перечень программного обеспечения приведены в таблице

Наименование	Количество копии	Стоимость за единицу
Windows 8.1	1	18394
Adobe Illustrator CC	1	Бесплатно (подписка на месяц)
MockingBot	1	Бесплатно (30 дней)
Android Studio	1	Бесплатно
Java Development Kit	1	Бесплатно
Итого		18 394

4.3 Расходы на оплату труда основным разработчикам и программистам

Основные затраты на реализацию программного продукта ложатся на выплату заработной платы работникам, которые рассчитываются по формуле:

$$\text{ФОТ} = \text{З}_{\text{осн}} + \text{З}_{\text{доп}}, \quad (4.2)$$

где $\text{З}_{\text{осн}}$ – основная заработная плата;
 $\text{З}_{\text{доп}}$ – дополнительная плата.

Для расчета затрат на основную заработную плату используются данные о средневзвешенной заработной плате и фактическом времени работы каждого сотрудника.

Средневзвешенной заработок

$$D = \frac{\text{ЗП}_m}{D_p} \quad (4.3)$$

где ЗП_m – ежемесячный размер заработной платы;
 D_p – количество рабочих дней в месяце (21 день).

Дизайнер:

$$D = \frac{25\,000}{21} = 1190 \text{ тенге/день};$$

Инженер-разработчик:

$$D = \frac{50\,000}{21} = 2381 \text{ тенге/день};$$

Программист:

$$D = \frac{50\,000}{21} = 2381 \text{ тенге/день};$$

Один час работы рассчитывается по формуле:

$$H = \frac{D}{\text{Ч}_p} \quad (4.4)$$

где D – средний дневной заработок работника; Ч_p – количество часов рабочего дня (8 часов).

Дизайнер

$$H = \frac{1190}{8} = 149 \text{ тенге/час};$$

Инженер-разработчик:

$$H = \frac{2381}{8} = 298 \text{ тенге/час};$$

Программист

$$H = \frac{2381}{8} = 298 \text{ тенге/час}.$$

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = \frac{T}{q_n * z * K} \quad (4.5)$$

где T – трудоемкость этапа, норма-час;

q_n – количество исполнителей поэтапу;

z – продолжительность рабочего дня, $z = 8$ часов;

$K = 952K$ – коэффициент выполнения норм времени, $K = 1,1$

Полученную величину t_n округляем в большую сторону до целых дней.

$$t_1 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} - \text{Программист, постановка задачи};$$

$$t_2 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} \text{ Дизайнер, изучение литературы};$$

$$t_3 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} \text{ Программист, изучение литературы};$$

$$t_4 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} \text{ Программист, выбор среды разработки};$$

$$t_5 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2 \text{ дня} \text{ Дизайнер, изучение «разделов питания»};$$

$$t_6 = \frac{44}{1 \cdot 8 \cdot 1,1} \approx 5 \text{ дней} \text{ Дизайнер, создание всех необходимых}$$

графических элементов;

$$t_7 = \frac{48}{1 \cdot 8 \cdot 1,1} \approx 6 \text{ дней} \text{ Дизайнер, создание всех необходимых}$$

графических элементов;

$$t_8 = \frac{132}{1 \cdot 8 \cdot 1,1} \approx 15 \text{ дней} \text{ Инженер - разработчик, разработка форм}$$

приложения;

$$t_9 = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1 \text{ дней} \text{ Программист, тестирование приложения};$$

$$t_{10} = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1 \text{ дней} \text{ Программист, отладка приложения};$$

$$t_{11} = \frac{24}{1 \cdot 8 \cdot 1,1} \approx 3 \text{ дня} \text{ Инженер-разработчик; ввод ПП в эксплуатацию.}$$

Сводные результаты расчета затрат на основную заработную плату работников, задействованных тенге/день, задействованных в разработке ПП представлены в таблице 4.5

в разработке Таблица 4.5 - Сводные результаты расчета затрат на основную заработную плату.

№	Наименование работ	Исполнитель	Трудоемкость		Длительность цикла, дни	Заработная плата за час работы, тенге	Сумма заработной платы, тенге
			Нормы-час	% отобшей			
1	Постановка задачи	Программист	16	4,8	2	298	4770
	Изучение литературы	Дизайнер	16	4,8	2	149	2386
	Изучение литературы	Инженер-разработчик	16	4,8	2	298	4770

	Выбор среды разработки	Программист	8	2,4	1	298	2385
	Изучение «гайдлайнов»	Инженер-разработчик	16	4,8	2	298	4770
2	Разработка интерфейса	Дизайнер	44	13	5	149	6561
	Создание необходимых графических элементов	Дизайнер	48	14	6	149	7158
3	Разработка форм приложения	Инженер-разработчик	132	39	15	298	39351
4	Тестирование приложения	Программист	8	2,4	1	298	2385
5	Отладка приложения	Программист	8	2,4	1	298	2385
	Оформление и сдача отчета	Инженер-разработчик	24	7	3	298	7155

4.4 Расчет затрат по социальному налогу

Социальный налог составляет 11% от дохода сотрудника и рассчитывается по формуле:

$$C_n = (\text{ФОТ} - \text{ПО}) * 0,11 \quad (4.7)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются и рассчитываются по формуле:

$$\text{ПО} = \text{ФОТ} * 0,1 \quad (4.8)$$

$$\text{ПО} = 92483,6 * 0,1 = 9248,36 \text{ тенге.}$$

Таким образом социальный налог составит:

$$C_n = (92483,6 - 9248,36) * 0,11 = 9155,88 \text{ тенге}$$

4.6 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле:

$$A_j = \frac{H_A * C_{\text{ПЕР}} * N}{100 * 12 * n} \quad (4.9)$$

где H_A – норма амортизации;

$C_{\text{ПЕР}}$ – первоначальная стоимость оборудования;

N – количество дней на выполнении работ;

n – количество рабочих дней в месяце.

Следовательно, амортизационные отчисления по используемому оборудованию и ПО, в соответствии с формулой 4.9 составят:

На оборудование

$$A_1 = \frac{40 * 494990 * 40}{100 * 12 * 21} = 31428 \text{ тенге}$$

$$A_1 = \frac{40 * 18394 * 40}{100 * 12 * 21} = 1168 \text{ тенге}$$

Суммарные затраты на амортизацию рассчитываются по формуле:

$$A = A_1 + A_2 \quad (4.10)$$

Амортизационные отчисления составляют:

$$A = 31428 + 1168 = 32596 \text{ тенге}$$

4.7. Расчет затрат на электроэнергию

Так как в процессе производства используется электрооборудование необходимо рассчитать затраты на электроэнергию. Затраты на электроэнергию для производственных нужд включают в себя расходы электроэнергии на оборудование и дополнительные нужды, рассчитываются по формуле 4.11.

$$\mathcal{E} = \mathcal{E}_{\text{эл.эн.обо}} + \mathcal{E}_{\text{доп.нуж}} \quad (4.11)$$

где $Z_{эл.эн.обор}$ – затраты на

электроэнергию оборудования;

$Z_{доп.нуж}$ – затраты электроэнергии на дополнительные нужды.

Расходы электроэнергии на оборудование рассчитываются по формуле:

$$Z_{эл.эн.обор} = W * T * S * K_{исп} \quad (4.12)$$

где W - потребляемая мощность, Вт;

T – количество часов работы оборудования;

S – стоимость киловатт-часа электроэнергии (1кВтч = 15,52 тенге);

$K_{исп}$ - коэффициент использования ($K_{исп} = 0,9$).

Потребляемая мощность HP Compaq 6710b составляет 90 Вт.

Потребляемая мощность HP Compaq 6710b составляет 65 Вт.

Потребляемой мощностью адаптеров для зарядки смартфона и планшета пренебрегаем, т.к. они используются исключительно для тестирования и отладки приложения, при этом постоянно подключены к ноутбуку HP Compaq 6710b.

Время высчитывается на основе количества рабочих дней и рабочих часов в день.

Сумма затрат на электроэнергию основного оборудования составляет:

$$Z_{эл.эн.обор} = (0,09 + 0,065) * (40 * 8) * 15,52 * 0,9 = 692,8 \text{ тенге}$$

Затраты на дополнительные нужды берутся по укрупненному показателю в размере 5% от затрат на оборудование:

$$Z_{доп.нуж} = 0,05 * Z_{эл.эн.обор} \quad (4.13)$$

$$Z_{доп.нуж} = 0,05 * 692,8 = 34,64 \text{ тенге}$$

Суммарные затраты на электроэнергию составляют:

$$\Sigma = 692,8 + 34,64 = 727,44 \text{ тенге}$$

4.8 Расчет накладных расходы

Накладные расходы рассчитываются как 50% от всех затрат и определяются по формуле:

$$H_p = (\text{ФОТ} + C_n + A + \text{Э}) * 0,5 \quad (4.14)$$

$$H_p = (92483,6 + 9155,88 + 32596 + 727,44) * 0,5 = 67481,46 \text{ тенге}$$

Таким образом, себестоимость разработки ПП, согласно формуле 4.1 составила:

$$C = 92483,6 + 9155,88 + 32596 + 727,44 + 67481,46 = 202\,444,38 \text{ тенге}$$

Сводные результаты расчета стоимости разработки приложения "Правильное питание" с применением кроссплатформенных технологий представлены в таблице 4.6 и на рисунке 4.1.

Наименование статей затрат	Сумма, тенге	В процентах от общей суммы, %
ФОТ	92483,6	46
Затраты по социальному налогу	9155,88	4,5
Амортизационные отчисления	32596	16
Затраты на электроэнергию	727,44	0,4
Накладные расходы	67481,46	33
Итого	202444,38	

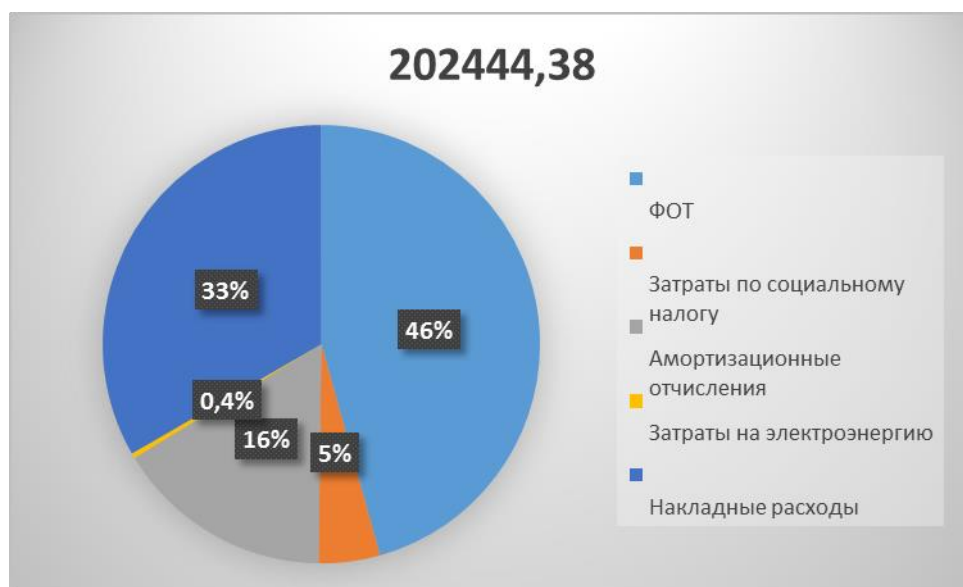


Рисунок 4.1 - Структура затрат на разработку приложения

4.9 Цена реализации проекта

Цена проекта складывается из себестоимости и желаемого чистого дохода. Минимальная цена программного продукта рассчитывается по следующей формуле:

$$Ц_{п} = C + П \quad (4.15)$$

где С – себестоимость разработки приложения;

П – чистый доход от программного продукта.

Для определения начальной цены используется желаемый уровень рентабельности. Для данной отрасли он составляет 25%.

$$Ц_{п} = C * \left(1 + \frac{P}{100}\right), \quad (4.16)$$

где P - рентабельность.

Тогда по формуле 4.16,

$$Ц_{п} = 202444,38 * \left(1 + \frac{25}{100}\right) = 253055,48 \text{ тенге}$$

Далее определяется цена реализации с учётом налога на добавленную стоимость (НДС) (формула 4.17):

$$Ц_{р} = Ц_{п} + \text{НДС} \quad (4.17)$$

где НДС – налог на добавленную стоимость.

Ставка НДС на 2016 год, согласно пункту 1 статье 268 « Ставки налога на добавленную стоимость» Налогового кодекса РК составляет 12 %.

В итоге получаем цену реализации проекта равной

$$Ц_{р} = 253055,48 + (253055,48 * 0,12) = 283422 \text{ тенге}$$

Выводы

Стоимость разработки приложения «Правильное Питание» для операционной системы Android составила 283422 тенге, в которую заложены все возможные затраты при разработки программного продукта.

По анализу расходов на реализацию проекта основной статьей расходов является затраты на оплату труда, более 45%.

Стоимость разработки мобильного приложения включает больших интеллектуальных и финансовых затрат. Расчеты показали, что производство и реализация данного программного продукта можно считать экономически выгодным.

Проблема здорового питания — одна из самых актуальных в наши дни. Полноценное питание предусматривает потребление достаточного количество белков, жиров, углеводов, витаминов, макро- и микроэлементов для нормального функционирования организма в целом. Многие болезни желудочно-кишечного тракта «молодеют» — это гастриты, язвенная болезнь желудка и различные нарушения обмена веществ.

Приложение «Правильное Питание» позволит урегулировать питание , что позволит предотвратить множество болезней и упорядочить свой режим дня.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда при разработке мобильного приложения

Для разработки мобильного приложения используются компьютерные технологии и электронное оборудование.

В рассматриваемом помещении работает 2 сотрудника.

При работе за компьютером для стабильной работы нужно соблюдать правильную организацию условий труда на месте работы. Длительное нахождение за компьютером негативно влияет на трудоспособность и здоровье рабочего. Чтобы избежать негативных последствий необходимо учесть нормы естественного и искусственного освещения, для обеспечения благоприятных условий работы.

Удобная и безопасная работа программиста во многом зависит от того, в каких условиях он работает и соответствуют ли они оптимальным. Условия работы подразумевают комплекс физических, химических, биологических и психофизических факторов, установленных стандартами по безопасности труда (ГОСТ 26883-86 «Внешние воздействующие факторы. Термины и определения»).

Характеристики рабочего помещения

Рассматриваемое рабочее помещение, расположено в здании, находящемся вдали от железнодорожных путей или нагруженных автомагистралей, аэропорта и так далее, поэтому внешних источников шума, влияющих на процесс работы – нет.

План помещения представлен на рисунке 5.1.

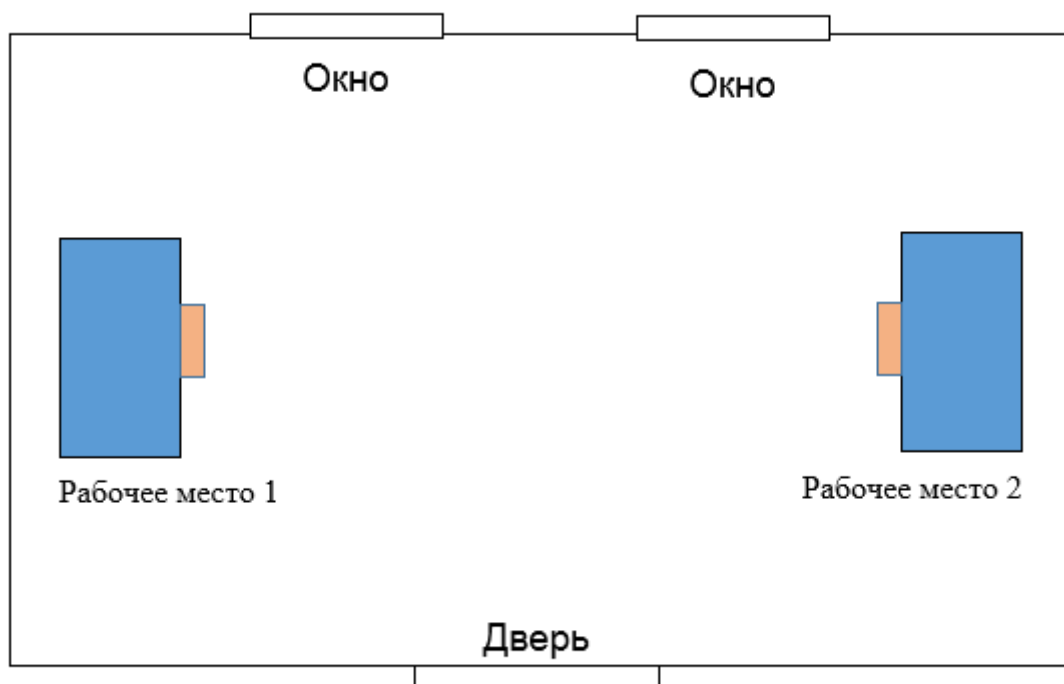


Рисунок 5.1 – План помещения

Параметры рабочего помещения:

- Рабочий кабинет находится на втором этаже.
- Размеры рабочего кабинета): длина 4,5 м, ширина 4 м, высота 2 м.
- Остекление помещения - двойное (одно окно размером 2400x1700мм) без стального переплетения.
- Солнцезащитные устройства –регулируемые жалюзи.
- Внутренняя отделка стен – светлая.
- Помещение по зрительным условиям работы относится категорией легких работ (категория Ia), выполняемых в сидячем положении (ГОСТ 12.2.032-78);
- Искусственное освещение – 4 светильника с двумя люминесцентными лампами.

Характеристики используемого в работе оборудования:

- Intel Pentium 4TravelMate 7720 Intel Core 2 Duo T8300 (2.4 ГГц, 35 Вт);
- Ноутбук HP Pavilion 15-n060er, Intel Core i5-4200U, 8 Gb DDR3, 1000 Gb HDD, GeForce GT 740M.
- Электропитание: переменное напряжение 240– 250 В, частотой 50 Гц и мощностью 90 Вт и 65 Вт соответственно.
- 4 люминесцентные лампы;
- Электропитание: переменное напряжение 220 – 250 В, частотой 50 Гц, мощность светильника 2x28 Вт.

Электротехническое оборудование является потенциальным источником возникновения пожарной опасности. Оборудование маломощное – вредность в качестве повышенного шума отсутствует.

5.1.1 Опасные и вредные факторы

При работе на компьютере на пользователя действуют следующие опасные физиологические и вредные факторы:

- недостаточность освещения может привести к ухудшению зрения, ослабляет внимание. Чрезмерно яркое освещение приводит к ослеплению, раздражению. Неправильное направление света может создавать тени, блики, и дезориентировать работающего. на экране монитора возникают статические заряды, которые заставляют частички пыли двигаться к ближайшему заземленному предмету ,а частицы пыли, оказавшиеся на коже, приводят к различным кожным заболеваниям;
- загрязнение воздуха вредными веществами, микроорганизмами и аэроионами;
- повышенный уровень шума на рабочем месте (создается вентиляторами местного охлаждения и электромеханическими устройствами);
- отсутствие или недостаточность естественного освещения;

– шум и вибрация от техники вызывают головную боль, понижение аппетита, негативные изменения в эмоциональном состоянии человека возникновение пожароопасной обстановки.

5.1.2 Электромагнитное излучение

Электромагнитного излучения провоцируют серьезные заболевания, к примеру болезнь Альцгеймера или Паркинсона, рак, а также всевозможные нарушения сна и памяти. Монитор создает серьезные электромагнитные возмущения. Производители техники знают об этой особенности, поэтому передняя часть монитора нередко получает защитное покрытие. А вот боковые и задние стенки – нет.

Для снижения воздействия этих видов излучения рекомендуется применять мониторы с пониженным уровнем излучения, устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

Таблица 5.1 – Допустимые значения параметров неионизирующих электромагнитных излучений (в соответствии с СанПиН 2.2.2.542-96)

Наименование параметра	Допустимые значения
Напряженность электрической составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	10В/м
Напряженность магнитной составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	0,3А/м
Напряженность электростатического поля не должна превышать: для взрослых пользователей для детей дошкольных учреждений и учащихся средних специальных и высших учебных заведений	20кВ/м 15кВ/м

5.1.3 Эргономика дисплея и устройств ввода информации

Для дисплея, вредными факторами, которые угрожают здоровью человека, являются:

- производимое им электромагнитное излучение;
- не качественность создаваемого на экране изображения.

Первый фактор является неотъемлемой частью дисплеев с электронно-лучевыми трубками (ЭЛТ). Дисплеи формируют как ионизирующее, так и не ионизирующее излучение. Ионизирующее излучение включает ультрафиолетовые лучи и способно нарушить химические связи в клетках человеческого организма, однако большинство ученых признает, что этот вид электромагнитного излучения не представляет для человека опасности, поскольку его уровень достаточно невелик и в основном поглощается покрытием экрана.

Не ионизирующим излучением являются электромагнитные волны сверхнизких частот.

Второй фактор присутствует у любых типов дисплеев, будь то дисплеи с ЭЛТ или с основой на жидких кристаллах.

Качество формируемого изображения можно оценивать по различным параметрам:

- размер видимого изображения на экране монитора. Эргономические стандарты предписывают, чтобы размер видимого изображения на экране был не менее 14" по диагонали. Производители мониторов предлагают на сегодняшний день выбор мониторов с диагоналями 14", 15", 17", 21" и более;

- частота обновления изображения на экране. Низкая частота обновления может привести к мерцанию изображения, что в свою очередь приводит к раздражению и быстрому утомлению глаз. На сегодняшний день эргономические стандарты требуют, чтобы частота обновления была не менее 70 Гц;

- инертность дисплея. Для дисплеев с ЭЛТ инертность определяется свойствами люминесцентного покрытия. Для таких дисплеев показатель инертности определяет необходимую частоту обновления изображения, и чем он меньше, тем более высокая должна быть частота обновления. Для дисплеев на жидких кристаллах (ЖКД) характерна очень высокая инертность, особенно для ЖКД с пассивной матрицей, поэтому быстро меняющееся изображение становится смазанным, что сильно затрудняет работу и требует повышенного напряжения зрительной системы. Решением проблемы смазанного изображения является использование более совершенных и более дорогих ЖКД с активной матрицей.

Кроме указанных выше требований к физическим характеристикам дисплея, эргономические стандарты ISO 9241, ANSI/HFS 100-1988 и «Правила об охране здоровья и безопасности труда при работе с графическими дисплеями» накладывают дополнительные ограничения и требования к дисплеям, окружающей обстановке и организации труда работающего:

- символы на экране должны быть четкими и легко распознаваемыми;
- яркость и контраст между символами и фоном должны без труда корректироваться пользователем;
- экран должен легко поворачиваться и наклоняться, чтобы принять удобное для пользователя положение;
- пользователю должна быть предоставлена возможность использовать отдельную подставку под дисплей или регулируемый стол;
- на экране дисплея должны отсутствовать блики и отражения, вызывающих неудобства для пользователя.

Все эти меры позволят снизить риск развития синдрома длительных зрительных нагрузок (СДЗН).

Устройства ввода информации, такие как клавиатура и мышь, также являются источником серьезной опасности. С работой на клавиатуре связана целая группа заболеваний, объединенная под общим названием - повреждения, вызванные повторяющимися нагрузками (RSI).

Проблема заболеваний связанных с СЗКП действительно является актуальной, особенно в развитых странах, где компьютеры получили широкое распространение в различных сферах человеческой деятельности.

Пользователи компьютеров могут уменьшить риск заболевания, предприняв некоторые шаги по улучшению эргономичности своего рабочего места и переходу на особый, здоровый стиль работы, который включает правильное положение работающего и тщательное планирование своего рабочего времени.

Рабочее место должно удовлетворять всем требованиям, изложенным в разделе «Эргономика монитора», кроме того эргономические стандарты накладывают дополнительные требования к положению работающего и геометрическим параметрам рабочего места :

- угол между плечом и предплечьем должен быть от 70° до 135° ;
- угол между торсом и бедром должен быть от 90° до 100° ;
- угол между верхней и нижней частью ноги должен быть от 60° до 100° ;
- ступни должны полностью стоять на полу;
- кресло должно быть регулируемое по высоте и наклону спинки;
- кресло должно быть на колесах, чтобы можно было легко сменить позицию перед терминалом;
- для обычного человека, высота стола, в среднем, 65-75см;
- при работе с клавиатурой, кисти рук должны быть на весу и не должны быть согнуты вверх, вниз или в другие стороны. Руки должны двигать кисти по клавиатуре, вместо того, чтобы держать кисти неподвижно и ударять по клавишам пальцами;
- при работе с клавиатурой не надо сильно ударять по клавишам, надо просто плавно нажимать на них;
- если необходимо нажать две клавиши одновременно, лучше использовать две руки.

По результатам многих исследований, при активной работе, рекомендуется каждый час делать перерыв на 10-15 минут. Это время надо просто не работать с клавиатурой. Во время перерыва желательно выполнить ряд упражнений, которые в дальнейшем облегчат работу и снизят риск развития СЗКП.

5.2. Оптимальный режим работы

Во время работы с персональным компьютером нельзя упускать соблюдение правильного режима труда и отдыха. Для обеспечения удобных условий труда предусматривается анализ рациональной организации умственного труда и выбор рационального труда и отдыха. Чтобы создать оптимальные условия для умственного труда нужна правильная организация. Важным условием для высокой работоспособности является определенный ритм работы. Определенный ритм работы является важным условием для высокой работоспособности. Ритмичный труд формируют определенный

стереотип, который помогает организму человека экономно расходовать энергию. Разным людям присущ различный ритм работы. Он отображен на рисунке 5.2.



Рисунок 5.2 - График оптимальной организации умственного труда

Очень важно соблюдать правильное (прямое) положение тела при сидении за столом во время работы.

Для снижения вредного воздействия на пользователя рекомендуется включить в восьми часовой рабочий день перерывы для отдыха общей продолжительностью 50 минут. Через 2 часа работы, утомление во время работы начинает резко возрастать, поэтому предлагается каждые 2 часа делать перерывы. Расписание перерывов на отдых:

- 2 часа работы – 15-20 минут отдыха и желательно проделать дыхательные упражнения на свежем воздухе;
- 2 часа работы - 30 минут перерыв на обед и отвлечься от работы;
- 2 часа работы – 15-20 минут отдыха и проделать упражнения для глаз;
- 2 часа работы - окончание работы.

Во время перерывов рекомендуется выполнять упражнения, включающие в себя шаги на месте, приседания, вращения и наклоны головой, сгибания и разгибание рук, наклоны и повороты туловища, для снятия утомления с туловища и ног, для улучшения кровообращения в области спины и живота, для предотвращения отечности, застоя крови и лимфообращения.

5.3 Расчет естественного освещения

На естественное освещение влияет множество факторов, таких как: позднее рабочее время, погода и так далее. Для этого предусмотрена система обеспечения искусственного освещения, которая состоит из светильников с люминесцентных ламп.

Нормативы на источники света приведены в таблице 5.2.

Таблица 5.2 – Рекомендуемые источники света при системе общего освещения.

Характеристика зрительной работы по требованиям к цветоразличению	Освещенность, лк	Индекс цветопередачи источников света, R_a	Диапазон цветовой температуры источников света T_c , К	Примерные источники света
1	2	3	4	5
Контроль цвета с очень высокими требованиями к цветоразличению, качество цветопередачи отличное	300 и более	90-100	5000-6500	ЛЛ типа: ЛДЦ, 950 ^{*)} , 965 ^{*)} ; СД
Сопоставление цветов с высокими требованиями к цветоразличению, качество цветопередачи отличное.	300 и более	85-89	3000-6500	ЛЛ типа: ЛТБЦЦ, ЛДЦ, 930 ^{*)} , 940 ^{*)} , 950 ^{*)} , 965 ^{*)} ; СД
Различение цветных объектов при невысоких требованиях к цветоразличению, качество цветопередачи хорошее	500 и более	80-84	4000-6000	ЛЛ типа: ЛЕЦ, 840 ^{*)} , 865 ^{*)} ; МГЛ, СД
	от 300 до 500	80-84	3500-5500	ЛЛ типа: ЛЕЦ 840 ^{*)} , 865 ^{*)} ; МГЛ, СД
	от 200 до 300	80-84	2700-4500	ЛЛ типа: ЛТБЦ827 ^{*)} , 830 ^{*)} ; МГЛ; СД, КЛЛ
	менее 200	80-84	2700-3500	ЛЛ типа: ЛТБЦ 827 ^{*)} , 830 ^{*)} ; МГЛ, НЛВД+МГЛ, СД, КЛЛ
Требования к цветоразличению отсутствуют, качество цветопередачи стандартное	500 и более	70-79	4000-6500	ЛЛ типа: ЛД 740 ^{*)} , 765 ^{*)} ; МГЛ, СД
	от 300 до 500	70-79	3500-5000	ЛЛ типа: ЛХБ 740 ^{*)} , 765 ^{*)} ; МГЛ, НЛВД+МГЛ, СД

Освещённость, необходимая для нормального выполнения работ в данном помещении: 400 лк. Для этого в помещении используются 4 люминесцентные лампы белого цвета. Этот выбор обусловлен тем, что люминесцентные лампы более экономичны, чем обычные лампы накаливания. Известно, что различный спектральный состав по-разному влияет на настроение человека. Лампы “теплого” цвета способствуют расслаблению, их лучше использовать в спальнях, местах для отдыха. Лампы “холодного” света способствуют зрительному и психоэмоциональному комфорту, их лучше использовать в гостиных, местах приема гостей и кухнях. Лампы “дневного” света способствуют концентрации внимания и увеличения работоспособности, их лучше использовать в кабинетах, местах, предназначенных для умственной работы. Так как лампы будут использоваться в рабочем помещении, были выбраны люминесцентные лампы ЛД-40.

Рассчитаем площадь боковых световых проемов помещения, необходимую для создания нормируемой освещенности на рабочем месте.

Помещение имеет размеры: длина $L = 6$ м, ширина $B = 4$ м, высота $H = 3$ м. Высота рабочей поверхности над уровнем пола $h_{рп} = 0,725$ м, окна начинаются с высоты $h_{рп} = 0,725$ м, высота окна $h_o = 1,5$ м, м. Рабочее помещение находится в IV часовом поясе – в г. Алматы (пояс светового климата – IV северной широты и южнее).

Рабочие места расположены в $l_{рт} = 0,5$ м, от наружной стены помещения, где проектируем оконные проемы. Минимальная освещенность будет в точке, отстоящей на расстояние 4 м от оконного проема.

Общую площадь окон $S_0, м^2$, определим по формулам 5.1 и 5.2

$$100 \frac{S_0}{S_n} = \frac{e_n \cdot \eta_0}{\tau_0 \cdot r_1} * k_{зд} * k_з, \quad (5.1)$$

$$S_0 = \frac{S_n \cdot e_n \cdot \eta_0}{\tau_0 \cdot r_1} \cdot k_{зд} \cdot k_з \quad (5.2)$$

где: S_n – площадь помещения ;
 $S_n = 24$ м²;

e_n – нормированное значение КЕО, выбираемое по таблице 5.1.

Для высокой точности зрительных работ принимаем $e_n = 1,2$

m_N – коэффициент светового климата, который находится по таблице 5.2.

Учитывая заданный световой пояс, приняв ориентацию световых проемов на Север, определим по формуле 5.3, при $m_N = 0,9$.

$$e_x^{IV} = e_x \cdot m \cdot c \quad (5.3)$$

где: $m = 0,7$;

$c = 0,75$ (в наружных стенах зданий);

$e_n = 1,2$ для работ высокой точности III разряда зрительной работы;

$$e_n^{IV} = 1,2 \cdot 0,7 \cdot 0,75 = 0,63.$$

Учитывая тип помещения, найдем коэффициент $k_3 = 1,2$ (учебные помещения, лаборатории, конструкторские бюро).

τ_0 – общий коэффициент светопропускания равный
 $\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$;

$\tau_1 = 0,5$ (пустотелые стеклянные блоки);

$\tau_2 = 0,6$ (деревянные двойные раздельные переплеты);

$\tau_3 = 0,8$ (железобетонные фермы и арки);

$\tau_4 = 1$ (убирающиеся регулируемые жалюзи и шторы);

η_0 – световая характеристика окон.

$$\tau_0 = 0,5 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,24.$$

Определяем по формуле 5.4

$$L = B - 1,$$

$$h_{расч} = h_{но} + h_0 - h_{рп} \quad (5.4)$$

$$L = 4 - 1 = 3 \text{ м}$$

$$L/1 = L/(B - 1) = 3/3 = 1,$$

$$h_{расч} = 0,8 + 1,5 - 0,75 = 1,57,$$

$$\frac{B}{h_{расч}} = \frac{4}{2,3} = 1,7.$$

По таблице 4.2 определим $\eta_0 = 10$.

τ_1 – коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя.

Средний коэффициент отражения в помещении $\rho_{ср} = 0,5$, принимаем одностороннее боковое освещение.

$$B/h_{расч} = 0,5/4 = 0,125,$$

тогда $r_1 = 1,05$.

$k_{зд}$ – коэффициент, учитывающий затенение окон противостоящими зданиями.

Поскольку затеняющих зданий поблизости нет, $k_{зд} = 1$.

Вычислим общую площадь окон:

$$S_0 = \frac{24 \cdot 1,35 \cdot 10 \cdot 1 \cdot 0,63}{100 \cdot 0,24 \cdot 1,05} = 8,16 \text{ м}^2.$$

Площадь световых проемов равна $S_{\text{сп}} = 8,16 \text{ м}^2$

Таким образом площадь световых проемов ($1,5 \cdot 1,2 \cdot 2 = 3,5 < 8,16$) не обеспечивает необходимых условий труда на рабочих местах. С целью создания наиболее благоприятных условий труда в помещении с параметрами $6 \times 4 \times 3$ для обеспечения нормированного значения КЕО, $e_N = 0,85$ при III характеристике зрительных работ, совместно с естественным освещением используется искусственное освещение.

5.3 Расчет искусственного освещения

Разряд зрительной работы – V. Нормируемая освещённость – 400 лк.

В качестве светильника возьмем ЛСП64-2. Длина светильника 1540 мм, ширина 276 мм (таблица 5.4).

Схема освещенности представлена на рисунке 5.3.

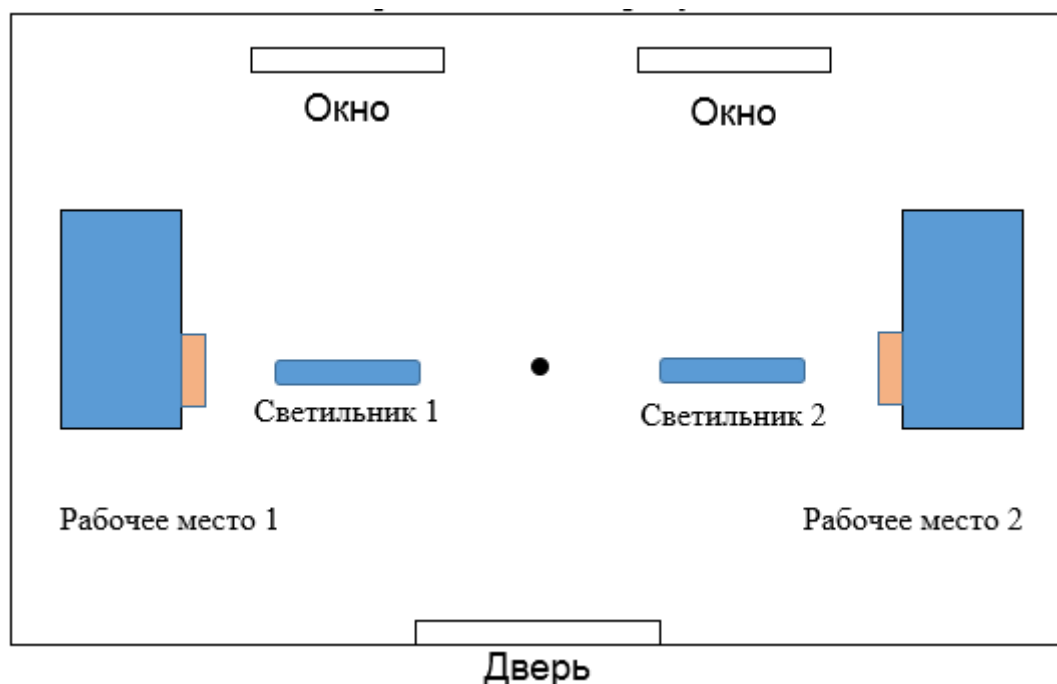


Рисунок 5.3 – Схема расчета освещенности

Таблица 5.3 – Технические характеристики газоразрядных ламп ЛБ

Номинальная мощность, Вт	Номинальный световой поток ламп типа ЛБ, лм	Размеры ламп, мм	
		Диаметр	Длина по штырькам
65	3570	40	1514,2

Коэффициенты отражения от потолка стен и пола соответственно равны:

$$\begin{aligned}\rho_{\text{пот}} &= 70\%, \\ \rho_{\text{ст}} &= 50\%, \\ \rho_{\text{пол}} &= 30\%.\end{aligned}$$

Вычислим высоту подвеса светильника над рабочей поверхностью по формуле 5.7:

$$H = h - h_p - h_c \quad (5.7)$$

где: h_c – расстояние от светильника до перекрытия, $h_c=0,05$ м;
 h_p – высота рабочей поверхности над полом, $h_p=0,7$ м;
 h – высота помещения, $h=3$ м.

$$H = 3 - 0,7 - 0,05 = 2,275 \text{ м.}$$

Лучшее расстояние от окна до светильника определяется по формуле.

$$L = \lambda \cdot H \quad (5.8)$$

где: $\lambda = 1,2 \div 1,4$

$$L = 1,25 \cdot 2,275 = 2,84 \text{ м}$$

Расстояние от стены до ближайшего светильника, когда работа у стены не проводится, определяем по формуле 5.9

$$l_1 = (0,4 \div 0,5) \cdot L \quad (5.9)$$

$$l_1 = 0,4 \cdot 2,73 = 1,138 \text{ м}$$

Определяем индекс помещения по формуле 5.10

$$i = \frac{l \cdot s}{H \cdot (l + s)}, \quad (5.10)$$

$$i = \frac{4 \cdot 2}{2,175 \cdot (4 + 2)} = 1,319$$

Коэффициент использования в данном случае равен $\eta = 65\%$, коэффициент запаса равен $k_z = 1,2$

Определим количество люминесцентных ламп по формуле 5.11

$$N = \frac{E \cdot k_z \cdot S_{\text{ос}} \cdot Z}{n \cdot \Phi_{\text{л}} \cdot \eta} \quad (5.11)$$

где: $S_{\text{ос}}$ – площадь помещения;
 k_z – коэффициент запаса;

E – заданная минимальная освещённость, $E=400$ лк.;
 Z – коэффициент неравномерности освещения, $Z=1,1$;
 n – количество ламп в светильнике;
 $\Phi_{л}$ – световой поток выбранной лампы, $\Phi_{л}=3570$ лм;
 η – коэффициент использования, $\eta=65\%$.

$$N = \frac{400 \cdot 1,2 \cdot 24 \cdot 1,1}{4 \cdot 3570 \cdot 0,65} \approx 4$$

Всего для создания нормируемой освещенности 400 лк необходимо 4 люминесцентных лампы серии ЛД, мощность каждой лампы должна быть не меньше 65 Вт, что соответствует действительности, а значит имеющегося в наличии освещения достаточно для соответствия санитарным нормам.

Заключение

В ходе выполнения данного дипломного проекта были усвоены основные принципы разработки приложений, а также предъявляемых к ним требований и стандартов. Изучение таких принципов, как проектирование и построение работы приложения для операционной системы Android стало наибольшей степени актуальной темой в современных реалиях, диктующего мобильного рынка.

На основе проведенного анализа магазина приложений GooglePlay было разработано мобильное приложение для изучения курса языка программирования C++ для соответствующей платформы. Приложение проектировалась на AndroidStudio, так как это наиболее удовлетворяющая по функциональным возможностям среда проектирования, которая имеет высокое качество безопасности и надежность в использовании большим числом пользователей. В ходе разработки внедрены уникальные функции, которые не были доступны в стандартных приложениях, а также в аналогичных проектируемых продуктах сторонних разработчиков. Данное приложение отвечает всем стандартам и требованиям платформ держателя. В ней предоставляется более удобный способ представление материала, что позволяет легко использовать интерфейс под нужны пользователя. Самым главным в реализации мобильного приложения стало получение опыта и участие в полном жизненном цикле в разработке ПО.

В экономической части дипломной работы была рассчитана общая стоимость продукта, в основе суммирования различных видов затрат, которые возможны при реализации. Данный расчет соответствует любому виду производства продукта, как на разработка для заказчика, так и самостоятельная разработка, для последующей публикации в интернет-магазине приложений. В процессе исследования аналогичных продуктов на рынке, установлено, что стоимость разработки полностью соответствует качеству разрабатываемого приложения и позволяет получить хороший объем прибыли.

В разделе безопасность жизнедеятельности были рассчитаны необходимые условия для организации, соответствующего всем установленным нормам рабочего места для сотрудников, занятых в реализации такого проекта. Рассчитано необходимое искусственное освещение в кабинете сотрудника, требующееся для комфортной работы без

нанесения ему вреда здоровью и произведен расчет системы автоматического пожаротушения.

Список используемой литературы

1. Сайт <https://www.youtube.com/user/freshgamer10/videos>
2. Сайт <http://startandroid.ru/ru/>
3. Сайт <http://developer.android.com/intl/ru/index.html>
4. Бадагуев Б.Т. Документация по охране труда в организации. – М.: Альфа-пресс, 2010.
5. Рето М., Android 2. Программирование приложений для планшетных компьютеров и смартфонов. – М. Эксмо, 2011.
6. Дюсебаев М.К., Бегимбетова А.С. Методические указания к выпускной работе (для студентов всех форм обучения специальностей 050719 – Радиотехника электроника и телекоммуникации, 050704 – Вычислительная техника и программное обеспечение). – Алматы: АИЭС, 2008.
7. СНиП РК 2.04-05-2002 «Естественное и искусственное освещение. Общие требования. – Астана, 2002.
8. Коробко, В.И. Охрана труда: Учебное пособие для студентов вузов / В.И. Коробко. – М.: ЮНИТИ-ДАНА, 2013. – 239 с.
9. Абдимуратов Ж.С, Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат. – Алматы: АИЭС, 2009.
10. Абдимуратов Ж. С., Маманбаева С. Е. Расчет производственного помещения. – Алматы: АУЭС, 2009.
11. Айдарханова М., Основы экономической теории. – М.: Фолиант, 2010.

12. Махотина М.В., Симоненко В.И. Экономика в схемах: Учебное пособие. – М.: Эксмо, 2011.
13. Носова С.С., Экономическая теория. – М.: Кнорус, 2010.
14. Артамонова В.С., Иванова С.А. Экономическая теория. – СПб.: Питер, 2010.
15. Камаева В.Д., Лобачевой Е.И. Экономическая теория: Учебник.– М.: Юрайт, 2010.

Приложение А

Листинг программы

```
package com.foodcalc.kz;

public final class BuildConfig {
public static final String APPLICATION_ID = "com.foodcalc.kz";
public static final String BUILD_TYPE = "debug";
public static final boolean DEBUG = Boolean.parseBoolean("true");
public static final String FLAVOR = "armv7";
public static final int VERSION_CODE = 12;
public static final String VERSION_NAME = "0.0.1";
}

package com.foodcalc.kz;

import android.os.Bundle;
import org.apache.cordova.CordovaActivity;

public class MainActivity extends CordovaActivity {
public void onCreate(Bundle var1) {
super.onCreate(var1);
this.loadUrl(this.launchUrl);
}
}

private static final int MAX_FINGERPRINT_LENGTH = 128;
private static final String TAG = "BuildInfo";

    @CalledByNative
public static String getAndroidBuildFingerprint() {
return Build.FINGERPRINT.substring(0, Math.min(Build.FINGERPRINT.length(),
128));
}

    @CalledByNative
public static String getAndroidBuildId() {
return Build.ID;
}

    @CalledByNative
public static String getBrand() {
return Build.BRAND;
}

    @CalledByNative
public static String getBuildType() {
return Build.TYPE;
}
```

```

    @CalledByNative
    public static String getDevice() {
    return Build.DEVICE;
    }

    @CalledByNative
    public static String getDeviceManufacturer() {
    return Build.MANUFACTURER;
    }

    @CalledByNative
    public static String getDeviceModel() {
    return Build.MODEL;
    }

    @CalledByNative
    public static String getPackageLabel(Context param0) {
    // $FF: Couldn't be decompiled
    }

    @CalledByNative
    public static String getPackageName(Context var0) {
    String var1;
    if(var0 != null) {
        var1 = var0.getPackageName();
    } else {
        var1 = null;
    }

    return var1 != null?var1:"";
    }

    @CalledByNative
    public static String getPackageVersionCode(Context param0) {
    // $FF: Couldn't be decompiled
    }

    @CalledByNative
    public static String getPackageVersionName(Context var0) {
    try {
        String var3 =
var0.getPackageManager().getPackageInfo(var0.getPackageName(),
0).versionName;
    return var3;
    } catch (NameNotFoundException var4) {
        android.util.Log.d("BuildInfo", "versionName not available");
    return "versionName not available";
    }
    }

    @CalledByNative
    public static int getSdkInt() {
    return VERSION.SDK_INT;
    }

    @TargetApi(21)
    @CalledByNative
    public static boolean hasLanguageApkSplits(Context var0) {
    if(VERSION.SDK_INT >= 21) {
        PackageInfo var1 = PackageUtils.getOwnPackageInfo(var0);
    if(var1.splitNames != null) {
    for(int var2 = 0; var2 < var1.splitNames.length; ++var2) {
    if(isLanguageSplit(var1.splitNames[var2])) {
    return true;
    }
    }
    }
    }
    }

```



```

    }
    }
}

return false;
}

private static boolean isLanguageSplit(String var0) {
return var0.length() == 9 && var0.startsWith("config.");
}

public static boolean isMncOrLater() {
return VERSION.SDK_INT >22 || TextUtils.equals("MNC", VERSION.CODENAME);
}
}

public class ViewConfiguration{
// $FF: synthetic field
static final boolean $assertionsDisabled = false;
private static final int ANDROID_LOW_MEMORY_DEVICE_THRESHOLD_MB = 512;
private static final int KBS_IN_MB = 1024;
private static final String TAG = "ViewConfiguration";
private static Boolean sLowEndDevice;

static {
boolean var0;
if(!SysUtils.class.desiredAssertionStatus()) {
var0 = true;
} else {
var0 = false;
}

$assertionsDisabled = var0;
}

private static int amountOfPhysicalMemoryMB() {
// $FF: Couldn't be decompiled
}

@TargetApi(19)
private static boolean detectLowEndDevice() {
if(!$assertionsDisabled && !CommandLine.isInitialized()) {
throw new AssertionError();
} else {
if(!CommandLine.getInstance().hasSwitch("enable-low-end-device-mode")) {
if(CommandLine.getInstance().hasSwitch("disable-low-end-device-mode")) {
return false;
}
}

if(VERSION.SDK_INT <19) {
return false;
}

Context var0 = ApplicationStatus.getApplicationContext();
if(var0 != null) {
if(((ActivityManager)var0.getSystemService("activity")).isLowRamDevice()) {
return true;
}
} else {
android.util.Log.e("SysUtils", "ApplicationContext is null in
ApplicationStatus");
}

int var2 = amountOfPhysicalMemoryMB();

```

```

    if(var2 <= 0) {
        return false;
    }

    if(var2 >512) {
        return false;
    }

    }

    return true;
}

    @CalledByNative
    public static boolean isLowEndDevice() {
        if(sLowEndDevice == null) {
            sLowEndDevice = Boolean.valueOf(detectLowEndDevice());
        }

        return sLowEndDevice.booleanValue();
    }
}

package org.chromium.base;

import android.text.TextUtils;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.concurrent.atomic.AtomicReference;
import org.chromium.base.ThreadUtils;
import org.chromium.base.VisibleForTesting;

public abstract class CommandLine {
    // $FF: synthetic field
    static final boolean $assertionsDisabled = false;
    private static final String SWITCH_PREFIX = "--";
    private static final String SWITCH_TERMINATOR = "--";
    private static final String SWITCH_VALUE_SEPARATOR = "=";
    private static final String TAG = "CommandLine";
    private static final AtomicReference sCommandLine;
    private static final List sResetListeners;

    static {
        boolean var0;
        if(!CommandLine.class.desiredAssertionStatus()) {
            var0 = true;
        } else {
            var0 = false;
        }

        $assertionsDisabled = var0;
        sResetListeners = new ArrayList();
        sCommandLine = new AtomicReference();
    }

    private CommandLine() {
}

```

```

// $FF: synthetic method
CommandLine(Object var1) {
    this();
}

public static void addResetListener(CommandLine.ResetListener var0) {
    sResetListeners.add(var0);
}

public static void enableNativeProxy() {
    sCommandLine.set(new CommandLine.NativeCommandLine(null));
}

@VisibleForTesting
public static CommandLine getInstance() {
    CommandLine var0 = (CommandLine)sCommandLine.get();
    if(!$assertionsDisabled && var0 == null) {
        throw new AssertionError();
    } else {
        return var0;
    }
}

public static String[] getJavaSwitchesOrNull() {
    CommandLine var0 = (CommandLine)sCommandLine.get();
    if(var0 != null) {
        if(!$assertionsDisabled && var0.isNativeImplementation()) {
            throw new AssertionError();
        } else {
            return ((CommandLine.JavaCommandLine)var0).getCommandLineArguments();
        }
    } else {
        return null;
    }
}

public static void init(String[] var0) {
    setInstance(new CommandLine.JavaCommandLine(var0));
}

public static void initFromFile(String var0) {
    char[] var1 = readUtf8FileFully(var0, 8192);
    String[] var2;
    if(var1 == null) {
        var2 = null;
    } else {
        var2 = tokenizeQuotedAruments(var1);
    }

    init(var2);
}

public static boolean isInitialized() {
    return sCommandLine.get() != null;
}

private static native void nativeAppendSwitch(String var0);

private static native void nativeAppendSwitchWithValue(String var0, String
var1);

private static native void nativeAppendSwitchesAndArguments(String[] var0);

```

```

private static native String nativeGetSwitchValue(String var0);

private static native boolean nativeHasSwitch(String var0);

private static native void nativeReset();

private static char[] readUtf8FileFully(String var0, int var1) {
    InputStreamReader var2 = null;
    File var3 = new File(var0);
    long var4 = var3.length();
    if(var4 == 0L) {
        return null;
    } else if(var4 > (long)var1) {
        android.util.Log.w("CommandLine", "File " + var0 + " length " + var4
+ " exceeds limit " + var1);
        return null;
    } else {
        int var6 = (int)var4;
        boolean var45 = false;

        InputStreamReader var18;
        char[] var17;
        label269: {
            char[] var24;
            label276: {
                InputStreamReader var8;
                label267: {
                    label277: {
                        label278: {
                            try {
                                var45 = true;
                                var17 = new char[var6];
                                var18 = new InputStreamReader(new
FileInputStream(var3), "UTF-8");
                                var45 = false;
                            } catch (FileNotFoundException var53) {
                                var45 = false;
                            }
                            break label277;
                            } catch (IOException var54) {
                                var45 = false;
                            }
                            break label278;
                            } finally {
                                if(var45) {
                                    if(var2 != null) {
                                        try {
                                            var2.close();
                                        } catch (IOException var48) {
                                            android.util.Log.e("CommandLine", "Unable
to close file reader.", var48);
                                        }
                                    }
                                }
                            }
                            label259: {
                                try {
                                    int var21 = var18.read(var17);
                                    if(!$assertionsDisabled && var18.ready()) {
                                        throw new AssertionError();
                                    }
                                }

                                if(var21 >= var17.length) {
                                    break label269;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
        var24 = Arrays.copyOfRange(var17, 0, var21);
break label276;
        } catch (FileNotFoundException var50) {
        ;
break label259;
        } catch (IOException var51) {
        } finally {
        ;
        }
        var8 = var18;
break label267;
    }
    var2 = var18;
}

if(var2 != null) {
try {
        var2.close();
    } catch (IOException var47) {
        android.util.Log.e("CommandLine", "Unable to close
file reader.", var47);
    }
}

return null;
}

var8 = null;
}

if(var8 != null) {
try {
        var8.close();
    } catch (IOException var46) {
        android.util.Log.e("CommandLine", "Unable to close file
reader.", var46);
    }
}

return null;
}

var17 = var24;
}

if(var18 != null) {
try {
        var18.close();
    } catch (IOException var49) {
        android.util.Log.e("CommandLine", "Unable to close file
reader.", var49);
    }
}

return var17;
}
}

public static void removeResetListener(CommandLine.ResetListener var0) {
    sResetListeners.remove(var0);
}

```

```

    }

    @VisibleForTesting
    public static void reset() {
        setInstance((CommandLine) null);
        ThreadUtils.postOnUiThread(new Runnable() {
    public void run() {
        Iterator var1 = CommandLine.sResetListeners.iterator();

        while(var1.hasNext()) {
            ((CommandLine.ResetListener) var1.next()).onCommandLineReset();
        }

        }
    });
}

private static void setInstance(CommandLine var0) {
    CommandLine var1 = (CommandLine) sCommandLine.getAndSet(var0);
    if(var1 != null && var1.isNativeImplementation()) {
        nativeReset();
    }
}

public static String[] tokenizeQuotedAruments(char[] var0) {
    ArrayList var1 = new ArrayList();
    StringBuilder var2 = null;
    char var3 = 0;
    int var4 = var0.length;

    for(int var5 = 0; var5 < var4; ++var5) {
        char var8 = var0[var5];
        if((var3 != 0 || var8 != 39 && var8 != 34) && var8 != var3) {
            if(var3 == 0 && Character.isWhitespace(var8)) {
                if(var2 != null) {
                    var1.add(var2.toString());
                    var2 = null;
                }
            } else {
                if(var2 == null) {
                    var2 = new StringBuilder();
                }

                var2.append(var8);
            }
        } else if(var2 != null && var2.length() > 0 && var2.charAt(-1 +
var2.length()) == 92) {
            var2.setCharAt(-1 + var2.length(), var8);
        } else if(var3 == 0) {
            var3 = var8;
        } else {
            var3 = 0;
        }
    }

    if(var2 != null) {
        if(var3 != 0) {
            android.util.Log.w("CommandLine", "Unterminated quoted string: "
+ var2);
        }

        var1.add(var2.toString());
    }
}

```

```

return (String[])var1.toArray(new String[var1.size()]);
    }

    @VisibleForTesting
    public abstract void appendSwitch(String var1);

    public abstract void appendSwitchWithValue(String var1, String var2);

    public abstract void appendSwitchesAndArguments(String[] var1);

    public abstract String getSwitchValue(String var1);

    public String getSwitchValue(String var1, String var2) {
        String var3 = this.getSwitchValue(var1);
        return TextUtils.isEmpty(var3)?var2:var3;
    }

    @VisibleForTesting
    public abstract boolean hasSwitch(String var1);

    public boolean isNativeImplementation() {
        return false;
    }

    private static class JavaCommandLine extends CommandLine {
        // $FF: synthetic field
        static final boolean $assertionsDisabled;
        private ArrayList mArgs = new ArrayList();
        private int mArgsBegin = 1;
        private HashMap mSwitches = new HashMap();

        static {
            boolean var0;
            if(!CommandLine.class.desiredAssertionStatus()) {
                var0 = true;
            } else {
                var0 = false;
            }

            $assertionsDisabled = var0;
        }

        JavaCommandLine(String[] var1) {
            super(null);
            if(var1 != null && var1.length != 0 && var1[0] != null) {
                this.mArgs.add(var1[0]);
                this.appendSwitchesInternal(var1, 1);
            } else {
                this.mArgs.add("");
            }

            if(!$assertionsDisabled &&this.mArgs.size() <= 0) {
                throw new AssertionError();
            }
        }

        private void appendSwitchesInternal(String[] var1, int var2) {
            boolean var3 = true;
            int var4 = var1.length;

            for(int var5 = 0; var5 < var4; ++var5) {
                String var6 = var1[var5];
                if(var2 >0) {

```

```

        --var2;
    } else {
if(var6.equals("--")) {
    var3 = false;
}

if(var3 && var6.startsWith("--")) {
    String[] var8 = var6.split("=", 2);
    String var9;
if(var8.length >1) {
    var9 = var8[1];
} else {
    var9 = null;
}

this.appendSwitchWithValue(var8[0].substring("--".length()), var9);
} else {
this.mArgs.add(var6);
}
}
}

private String[] getCommandLineArguments() {
return (String[])this.mArgs.toArray(new String[this.mArgs.size()]);
}

public void appendSwitch(String var1) {
this.appendSwitchWithValue(var1, (String)null);
}

public void appendSwitchWithValue(String var1, String var2) {
    HashMap var3 = this.mSwitches;
    String var4;
if(var2 == null) {
    var4 = "";
} else {
    var4 = var2;
}

    var3.put(var1, var4);
    String var6 = "--" + var1;
if(var2 != null && !var2.isEmpty()) {
    var6 = var6 + "=" + var2;
}

    ArrayList var7 = this.mArgs;
int var8 = this.mArgsBegin;
this.mArgsBegin = var8 + 1;
    var7.add(var8, var6);
}

public void appendSwitchesAndArguments(String[] var1) {
this.appendSwitchesInternal(var1, 0);
}

public String getSwitchValue(String var1) {
    String var2 = (String)this.mSwitches.get(var1);
if(var2 == null || var2.isEmpty()) {
    var2 = null;
}

return var2;

```



```

    }

    public boolean hasSwitch(String var1) {
    return this.mSwitches.containsKey(var1);
    }
}

private static class NativeCommandLine extends CommandLine {
private NativeCommandLine() {
super(null);
}

// $FF: synthetic method
NativeCommandLine(Object var1) {
this();
}

public void appendSwitch(String var1) {
    CommandLine.nativeAppendSwitch(var1);
}

public void appendSwitchWithValue(String var1, String var2) {
    CommandLine.nativeAppendSwitchWithValue(var1, var2);
}

public void appendSwitchesAndArguments(String[] var1) {
    CommandLine.nativeAppendSwitchesAndArguments(var1);
}

public String getSwitchValue(String var1) {
return CommandLine.nativeGetSwitchValue(var1);
}

public boolean hasSwitch(String var1) {
return CommandLine.nativeHasSwitch(var1);
}

public boolean isNativeImplementation() {
return true;
}
}

public interface ResetListener {
void onCommandLineReset();
}
}

public class ColorPickerDialog extends AlertDialog implements
OnColorChangedListener {

private final ColorPickerAdvanced mAdvancedColorPicker;

private int mCurrentColor = this.mInitialColor;

private final View mCurrentColorView;

private final int mInitialColor;

private final OnColorChangedListener mListener;

private final Button mMoreButton;

```



```

        View content = inflater.inflate(R.layout.color_picker_dialog_content,
null);

        setContentView(content);

        this.mMoreButton = (Button) content.findViewById(R.id.more_colors_button);
        this.mMoreButton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                ColorPickerDialog.this.showAdvancedView();
            }
        });

        this.mAdvancedColorPicker = (ColorPickerAdvanced)
content.findViewById(R.id.color_picker_advanced);

        this.mAdvancedColorPicker.setVisibility(8);

        this.mSimpleColorPicker = (ColorPickerSimple)
content.findViewById(R.id.color_picker_simple);

        this.mSimpleColorPicker.init(suggestions, this);
        updateCurrentColor(this.mInitialColor);
    }

    public void onColorChanged(int color) {
        updateCurrentColor(color);
    }

    private void showAdvancedView() {
        findViewById(R.id.more_colors_button_border).setVisibility(8);
        findViewById(R.id.color_picker_simple).setVisibility(8);

        this.mAdvancedColorPicker.setVisibility(0);
        this.mAdvancedColorPicker.setListener(this);
        this.mAdvancedColorPicker.setColor(this.mCurrentColor);
    }

    private void tryNotifyColorSet(int color) {
        if (this.mListener != null) {
            this.mListener.onColorChanged(color);
        }
    }

```

```

    }
}

private void updateCurrentColor(int color) {

    this.mCurrentColor = color;

    if (this.mCurrentColorView != null) {

        this.mCurrentColorView.setBackgroundColor(color);

    }

}

}

package com.foodcalc.kz;

public final class NavigationParams {
    public static final class array {
        public static final int xwalk_resources_list = 2131230720;
    }

    public static final class attr {
        public static final int buttonColor = 2130771970;
        public static final int select_dialog_multichoice = 2130771968;
        public static final int select_dialog_singlechoice = 2130771969;
    }

    public static final class color {
        public static final int color_picker_background_color = 2131296256;
        public static final int color_picker_border_color = 2131296257;
        public static final int dropdown_dark_divider_color = 2131296258;
        public static final int dropdown_divider_color = 2131296259;
    }

    public static final class dimen {
        public static final int color_button_height = 2131361792;
        public static final int color_picker_gradient_margin = 2131361793;
        public static final int config_min_scaling_span = 2131361794;
        public static final int config_min_scaling_touch_major = 2131361795;
        public static final int dropdown_item_divider_height = 2131361796;
        public static final int dropdown_item_height = 2131361797;
        public static final int keyboard_accessory_chip_height = 2131361798;
        public static final int keyboard_accessory_fading_edge_length = 2131361799;
        public static final int keyboard_accessory_half_padding = 2131361800;
        public static final int keyboard_accessory_height = 2131361801;
        public static final int keyboard_accessory_padding = 2131361802;
        public static final int keyboard_accessory_text_size = 2131361803;
        public static final int link_preview_overlay_radius = 2131361804;
    }

    public static final class drawable {
        public static final int autofill_chip_inset = 2130837504;
        public static final int bubble = 2130837505;
        public static final int bubble_arrow_up = 2130837506;
        public static final int button_borderless_compat = 2130837507;
        public static final int button_compat = 2130837508;
        public static final int button_compat_shape = 2130837509;
    }
}

```

```

public static final int color_button_background = 2130837510;
public static final int color_picker_advanced_select_handle = 2130837511;
public static final int color_picker_border = 2130837512;
public static final int dropdown_popup_background = 2130837513;
public static final int dropdown_popup_background_down = 2130837514;
public static final int dropdown_popup_background_up = 2130837515;
public static final int ic_menu_share_holo_light = 2130837516;
public static final int ic_search = 2130837517;
public static final int ic_warning = 2130837518;
public static final int icon = 2130837519;
public static final int ondemand_overlay = 2130837520;
public static final int screen = 2130837521;
public static final int verify_checkmark = 2130837522;
    }

public static final class id {
public static final int ampm = 2131427357;
public static final int arrow_image = 2131427364;
public static final int autofill_keyboard_accessory_item_label = 2131427329;
public static final int autofill_keyboard_accessory_item_sublabel =
2131427330;
public static final int color_button_swatch = 2131427367;
public static final int color_picker_advanced = 2131427335;
public static final int color_picker_simple = 2131427336;
public static final int date_picker = 2131427342;
public static final int date_time_suggestion = 2131427344;
public static final int date_time_suggestion_label = 2131427346;
public static final int date_time_suggestion_value = 2131427345;
public static final int dropdown_icon = 2131427350;
public static final int dropdown_label = 2131427348;
public static final int dropdown_label_wrapper = 2131427347;
public static final int dropdown_popup_window = 2131427328;
public static final int dropdown_sublabel = 2131427349;
public static final int gradient = 2131427333;
public static final int gradient_border = 2131427332;
public static final int hour = 2131427351;
public static final int icon_view = 2131427362;
public static final int main_text = 2131427365;
public static final int milli = 2131427356;
public static final int minute = 2131427352;
public static final int more_colors_button = 2131427338;
public static final int more_colors_button_border = 2131427337;
public static final int pickers = 2131427358;
public static final int position_in_year = 2131427359;
public static final int second = 2131427354;
public static final int second_colon = 2131427353;
public static final int second_dot = 2131427355;
public static final int seek_bar = 2131427334;
public static final int select_action_menu_copy = 2131427370;
public static final int select_action_menu_cut = 2131427369;
public static final int select_action_menu_paste = 2131427371;
public static final int select_action_menu_select_all = 2131427368;
public static final int select_action_menu_share = 2131427372;
public static final int select_action_menu_web_search = 2131427373;
public static final int selected_color_view = 2131427341;
public static final int selected_color_view_border = 2131427340;
public static final int sub_text = 2131427366;
public static final int text = 2131427331;
public static final int text_wrapper = 2131427363;
public static final int time_picker = 2131427343;
public static final int title = 2131427339;
public static final int top_view = 2131427361;
public static final int year = 2131427360;
    }

```

```

public static final class layout {
public static final int autofill_keyboard_accessory_icon = 2130903040;
public static final int autofill_keyboard_accessory_item = 2130903041;
public static final int color_picker_advanced_component = 2130903042;
public static final int color_picker_dialog_content = 2130903043;
public static final int color_picker_dialog_title = 2130903044;
public static final int date_time_picker_dialog = 2130903045;
public static final int date_time_suggestion = 2130903046;
public static final int dropdown_item = 2130903047;
public static final int multi_field_time_picker_dialog = 2130903048;
public static final int two_field_date_picker = 2130903049;
public static final int validation_message_bubble = 2130903050;
}

public static final class menu {
public static final int select_action_menu = 2131492864;
}

public static final class raw {
public static final int contacts_api = 2131034112;
public static final int device_capabilities_api = 2131034113;
public static final int icudtl = 2131034114;
public static final int launch_screen_api = 2131034115;
public static final int messaging_api = 2131034116;
public static final int presentation_api = 2131034117;
public static final int wifidirect_api = 2131034118;
public static final int xwalk = 2131034119;
}

public static final class string {
public static final int accessibility_date_picker_month = 2131099648;
public static final int accessibility_date_picker_week = 2131099649;
public static final int accessibility_date_picker_year = 2131099650;
public static final int accessibility_datetime_picker_date = 2131099651;
public static final int accessibility_datetime_picker_time = 2131099652;
public static final int accessibility_time_picker_ampm = 2131099653;
public static final int accessibility_time_picker_hour = 2131099654;
public static final int accessibility_time_picker_milli = 2131099655;
public static final int accessibility_time_picker_minute = 2131099656;
public static final int accessibility_time_picker_second = 2131099657;
public static final int actionbar_share = 2131099658;
public static final int actionbar_web_search = 2131099659;
public static final int activity_name = 2131099702;
public static final int app_name = 2131099703;
public static final int autofill_popup_content_description = 2131099660;
public static final int color_picker_button_black = 2131099661;
public static final int color_picker_button_blue = 2131099662;
public static final int color_picker_button_cancel = 2131099663;
public static final int color_picker_button_cyan = 2131099664;
public static final int color_picker_button_green = 2131099665;
public static final int color_picker_button_magenta = 2131099666;
public static final int color_picker_button_more = 2131099667;
public static final int color_picker_button_red = 2131099668;
public static final int color_picker_button_set = 2131099669;
public static final int color_picker_button_white = 2131099670;
public static final int color_picker_button_yellow = 2131099671;
public static final int color_picker_dialog_title = 2131099672;
public static final int color_picker_hue = 2131099673;
public static final int color_picker_saturation = 2131099674;
public static final int color_picker_value = 2131099675;
public static final int copy_to_clipboard_failure_message = 2131099676;
public static final int crosswalk_install_title = 2131099704;
public static final int date_picker_dialog_clear = 2131099677;
}

```

```
public static final int date_picker_dialog_other_button_label = 2131099678;
public static final int date_picker_dialog_set = 2131099679;
public static final int date_picker_dialog_title = 2131099680;
public static final int date_time_picker_dialog_title = 2131099681;
public static final int decompression_progress_message = 2131099705;
public static final int download_already_exists_toast = 2131099706;
public static final int download_failed_device_not_found = 2131099707;
public static final int download_failed_insufficient_space = 2131099708;
public static final int download_failed_message = 2131099709;
public static final int download_failed_time_out = 2131099710;
public static final int download_failed_toast = 2131099711;
public static final int download_finished_toast = 2131099712;
public static final int download_no_permission_toast = 2131099713;
public static final int download_progress_message = 2131099714;
public static final int download_start_toast = 2131099715;
public static final int http_auth_log_in = 2131099716;
public static final int http_auth_password = 2131099717;
public static final int http_auth_title = 2131099718;
public static final int http_auth_user_name = 2131099719;
public static final int js_alert_title = 2131099720;
public static final int js_confirm_title = 2131099721;
public static final int js_prompt_title = 2131099722;
public static final int launcher_name = 2131099723;
public static final int low_memory_error = 2131099682;
public static final int market_open_failed_message = 2131099724;
public static final int media_player_error_button = 2131099683;
public static final int media_player_error_text_invalid_progressive_playback
= 2131099684;
public static final int media_player_error_text_unknown = 2131099685;
public static final int media_player_error_title = 2131099686;
public static final int media_player_loading_video = 2131099687;
public static final int month_picker_dialog_title = 2131099688;
public static final int opening_file_error = 2131099689;
public static final int password_generation_popup_content_description =
2131099690;
public static final int profiler_error_toast = 2131099691;
public static final int profiler_no_storage_toast = 2131099692;
public static final int profiler_started_toast = 2131099693;
public static final int profiler_stopped_toast = 2131099694;
public static final int ssl_alert_title = 2131099725;
public static final int startup_architecture_mismatch_message = 2131099726;
public static final int startup_architecture_mismatch_title = 2131099727;
public static final int startup_incomplete_library_message = 2131099728;
public static final int startup_incomplete_library_title = 2131099729;
public static final int startup_newer_version_message = 2131099730;
public static final int startup_newer_version_title = 2131099731;
public static final int startup_not_found_message = 2131099732;
public static final int startup_not_found_title = 2131099733;
public static final int startup_older_version_message = 2131099734;
public static final int startup_older_version_title = 2131099735;
public static final int startup_signature_check_error_message = 2131099736;
public static final int startup_signature_check_error_title = 2131099737;
public static final int time_picker_dialog_am = 2131099695;
public static final int time_picker_dialog_hour_minute_separator =
2131099696;
public static final int time_picker_dialog_minute_second_separator =
2131099697;
public static final int time_picker_dialog_pm = 2131099698;
public static final int time_picker_dialog_second_subsecond_separator =
2131099699;
public static final int time_picker_dialog_title = 2131099700;
public static final int week_picker_dialog_title = 2131099701;
public static final int xwalk_cancel = 2131099738;
public static final int xwalk_close = 2131099739;
```

```
public static final int xwalk_continue = 2131099740;
public static final int xwalk_get_crosswalk = 2131099741;
public static final int xwalk_retry = 2131099742;
}

public static final class style {
public static final int ButtonCompat = 2131165184;
public static final int ButtonCompatBase = 2131165185;
public static final int ButtonCompatBorderless = 2131165186;
public static final int ButtonCompatBorderlessOverlay = 2131165187;
public static final int ButtonCompatOverlay = 2131165188;
public static final int DropdownPopupWindow = 2131165189;
public static final int SelectActionMenuShare = 2131165190;
public static final int SelectActionMenuWebSearch = 2131165191;
public static final int SelectPopupDialog = 2131165192;
}

public static final class styleable {
public static final int[] ButtonCompat = new int[]{2130771970};
public static final int ButtonCompat_buttonColor;
}

public static final class xml {
public static final int config = 2130968576;
}
}
```