

**МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**

**САМАРКАНДСКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ИМЕНИ МУХАММАДА АЛЬ-ХОРАЗМИ**

**Факультет: Телекоммуникационные технологии и профессиональное
образование**

Кафедра: Информационно-образовательные технологии

**Направление: 5350400 -“ Профессиональное образование в сфере
информационно-коммуникационных технологий”**

“Допущена к защите”

И.о.заведующий кафедрой

_____ Шодмонов Д.А.

“ _____ ” _____ 2018 г.

ВЫПУСКНАЯ

КВАЛИФИКАЦИОННАЯ РАБОТА

для получения академической степени бакалавра

**на тему: разработка мобильного приложения учебного назначения по
математике**

Выпускник: _____ Боболов Умид

Научный руководитель: _____ Ганиева Н.А.

Рецензент: _____ Кобилов С.С.

Консультант по БЖД: _____ Курбонов Э.

САМАРКАНД – 2018

СОДЕРЖАНИЕ

Введение	1
Глава 1. Теоретические основы систем обучения	5
1.1 Система дистанционного обучения.....	5
1.2 Электронный учебник.....	10
1.3 Обзор функциональных возможностей и типов систем дистанционного обучения.....	12
Глава 2. Средства и среда разработки мобильного приложения для системы дистанционного обучения.....	177
2.1 Среда разработки под ОС Андроид.....	17
2.2 Разработка приложения для платформы Android.....	32
Глава 3. Процесс разработки мобильного приложения	37
3.1 Этап разработки интерфейсной части программы	37
3.2 Этап апробации и тестирования разработанной программы	39
4. Техническая безопасность.....	42
Заключение	48
Литература.....	49
Приложение	51

ВВЕДЕНИЕ

Указом Президента Республики Узбекистан 7 февраля 2017 года была утверждена Стратегия действий по пяти приоритетным направлениям развития Республики Узбекистан в 2017-2021 годах [1].

В наше время электронных технологий, дистанционное обучение приобретает огромную популярность среди учебных заведений. Развитие информационных технологий подстегнуло огромный спрос на дистанционное обучение. Если раньше такое было немислимо—то теперь любой желающий имеет возможность, не выходя из дома получить диплом по любой специальности. Педагогами возлагаются надежды на компьютерные технологии, как на средство способное повысить качество обучения в массовых масштабах. Создание электронных учебников, проведение тестирования и трансляции видео в прямом эфире напрямую из аудитории, все это доступно любому педагогу уже сегодня. В системах дистанционного обучения используются все виды информационных технологий, но преимущественно новые информационные технологии, средствами которых являются компьютеры, мобильные устройства, компьютерные сети, мультимедиа системы и т.д.

В тоже время большую популярность обрели мобильные устройства, которые существенно облегчают жизнь современного человека. Но, к сожалению, данный вид устройств в данный момент не используется как платформа для дистанционного обучения. Из-за огромного количества игр для мобильных устройств, многие ученики используют их для развлечения, что негативно сказывается на успеваемости. Но такую любовь к играм можно обернуть в пользу учебы с помощью использования игровой формы обучения.

Актуальность темы. Огромное распространение мобильных устройств в мире помогает людям во многих сферах в жизни человека. Но, к сожалению, использование этих устройств для обучения пользуется меньшим спросом, хотя имеет большой потенциал. В наше время ученики в

свободное от учебы время в основном используют свои мобильные устройства для изучения, контроля и развлечения, но путем реализации учебного материала в виде игровой формы можно добиться интереса к предмету у учащихся.

Целью работы является создание мобильного приложения для системы дистанционного и самостоятельного обучения с возможностью сохранения содержания курсов школьных предметов на узбекском языке. Тем самым обеспечение необходимым материалом школьников и учителей с узбекским языком обучения.

Постановка задачи. Постановка задачи заключается в разработке приложения для системы обучения. В данной работе рассматривается задача создания среды дистанционного обучения и самостоятельного обучения в мобильном устройстве с возможностью сохранять просмотренные курсы с последующим просмотром. Возможность учащимся проходить тесты, контрольные работы, а так же просматривать курсы в режиме оффлайн.

Практическая и теоретическая ценность. Работа носит практический характер. Результатом работы является программа, бесплатная для использования в некоммерческих целях.

Структура и объем работы. Выпускная квалификационная работа состоит из введения, 3-х глав и заключения, изложенных на страницах машинописного текста. Содержит список использованной литературы из 16 наименований и приложения.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СИСТЕМ ОБУЧЕНИЯ

1.1. Система дистанционного обучения

Дистанционную форму обучения специалисты по стратегическим проблемам образования называют образовательной системой 21 века. Сегодня на нее сделана огромная ставка. Актуальность темы дистанционного обучения заключается в том, что результаты общественного прогресса, ранее сосредоточенные в сфере технологий сегодня концентрируются в информационной сфере. Наступила эра информатики. Этап её развития в настоящий момент можно характеризовать как телекоммуникационный. Эта область общения, информации и знаний. Исходя из того, что профессиональные знания стареют очень быстро, необходимо их постоянное совершенствование. Дистанционную форму обучения дает сегодня возможность создания систем массового непрерывного самообучения, всеобщего обмена информацией, независимо от временных и пространственных поясов. Кроме того, системы дистанционного образования дают равные возможности всем людям независимо от социального положения (школьникам, студентам, гражданским и военным, безработными и т. д.) в любых районах страны и за рубежом реализовать права человека на образование и получение информации. Именно эта система может наиболее адекватно и гибко реагировать на потребности общества и обеспечить реализацию конституционного права на образование каждого гражданина страны. Исходя из вышеуказанных факторов можно заключить, что **дистанционное** обучение войдет в 21 век как самая эффективная система подготовки и непрерывного поддержания высокого квалификационного уровня специалистов [2,3].

В последнее время проблеме дистанционного обучения уделяется большое внимание в педагогической литературе. В характеристике использованных первоисточников отмечается научный подход (употребление терминологии, ее раскрытие, выведение и обоснование основных положений, лаконичность и логичность изложения); однако,

наблюдается некоторое расхождение авторов изданий разных лет по тем или иным вопросам.

Сопоставляя данные статей, можно заключить, что дистанционное обучение — это новая, специфичная форма обучения, несколько отличная от привычных форм очного или заочного обучения. Она предполагает иные средства, методы, организационные формы обучения, иную форму взаимодействия учителя и учащихся, учащихся между собой. Вместе с тем как любая форма обучения, любая система обучения она имеет тот же компонентный состав: цели, обусловленные социальным заказом для всех форм обучения; содержание, также во многом определенное действующими программами для конкретного типа учебного заведения, методы, организационные формы, средства обучения. Последние три компонента. В дистанционной форме обучения обусловлены спецификой используемой технологической основы (например, только компьютерных телекоммуникаций, компьютерных телекоммуникаций в комплексе с печатными средствами, компакт-дисками, так называемой кейс-технологией, пр.).

Не следует смешивать заочное и дистанционное обучения. Их главное отличие в том, что при дистанционном обучении обеспечивается систематическая и эффективная интерактивность. Следует рассматривать дистанционное обучение как новую форму обучения и соответственно дистанционное образование (как результат, так и процесс, систему) как новую форму образования. Хотя оно не может рассматриваться как система совершенно автономная. Дистанционное обучение строится в соответствии с теми же целями и содержанием, что и очное обучение. Но формы подачи материала и формы взаимодействия учителя и учащихся и учащихся между собой различны. Дидактические принципы организации дистанционного обучения (принципы научности, системности и систематичности, активности, принципы развивающего обучения, наглядности, дифференциации и индивидуализации обучения и пр.) те же что и в очном обучении, но отлична

их реализация которая обусловлена спецификой новой формы обучения, возможностями информационной среды Интернет, ее услугами [3].

Таким образом, с одной стороны, дистанционное обучение следует рассматривать в общей системе образования (непрерывно в системе непрерывного образования), предполагая при этом преемственность отдельных ее звеньев. С другой, дистанционное обучение необходимо различать как систему и как процесс. Как и в других формах обучения, дистанционное обучение предполагает теоретическое осмысление этапа педагогического проектирования, ее содержательной и педагогической (в плане педагогических технологий, методов, форм обучения) составляющих.

Следовательно задачами этапа педагогического проектирования являются: создание электронных курсов, электронных учебников, комплексов средств обучения, разработка педагогических технологий организации процесса обучения в сетях [3,4].

Курсы дистанционного обучения предполагают тщательное и детальное планирование деятельности обучаемого, ее организации, четкую постановку задач и целей обучения, доставку необходимых учебных материалов, которые должны обеспечивать интерактивность между обучаемым и преподавателем, обратную связь между обучаемым и учебным материалом, предоставлять возможность группового обучения. Наличие эффективной обратной связи позволяющей ученику получать информацию о правильности своего продвижения по пути от незнания к знанию. Мотивация — также важнейший элемент любого курса дистанционного обучения. Для её повышения, важно применять разнообразные приемы и средства. А так же необходимо предусмотреть инвариантные компоненты при разработке курсов дистанционного обучения.

Эффективность дистанционного обучения напрямую зависит от тех преподавателей, кто ведет работу с учащимися в Интернет. Это должны быть преподаватели с универсальной подготовкой: владеющие современными

педагогическими и информационными технологиями, психологически готовые к работе с учащимися в новой учебно-познавательной сетевой среде.

К сожалению, в нашей стране не ведётся подготовка специалистов подобного рода. Другая проблема - инфраструктура информационного обеспечения студента в сетях. Вопрос о том, какой должна быть структура и композиция учебного материала остаётся открытым. Наряду с этим ставится вопрос об условиях доступа к курсам дистанционного обучения. Не решен так же вопрос организации и проведения оценки знаний "дистанционных" учащихся. Для его решения необходимо создание нормативно-правовой базы оценки знаний учащихся.

Комплекс СДО включает в себя программное обеспечение, реализующее основные компоненты и функции СДО, а также аппаратную часть, состоящую из WEB-сервера и компьютеров пользователей.

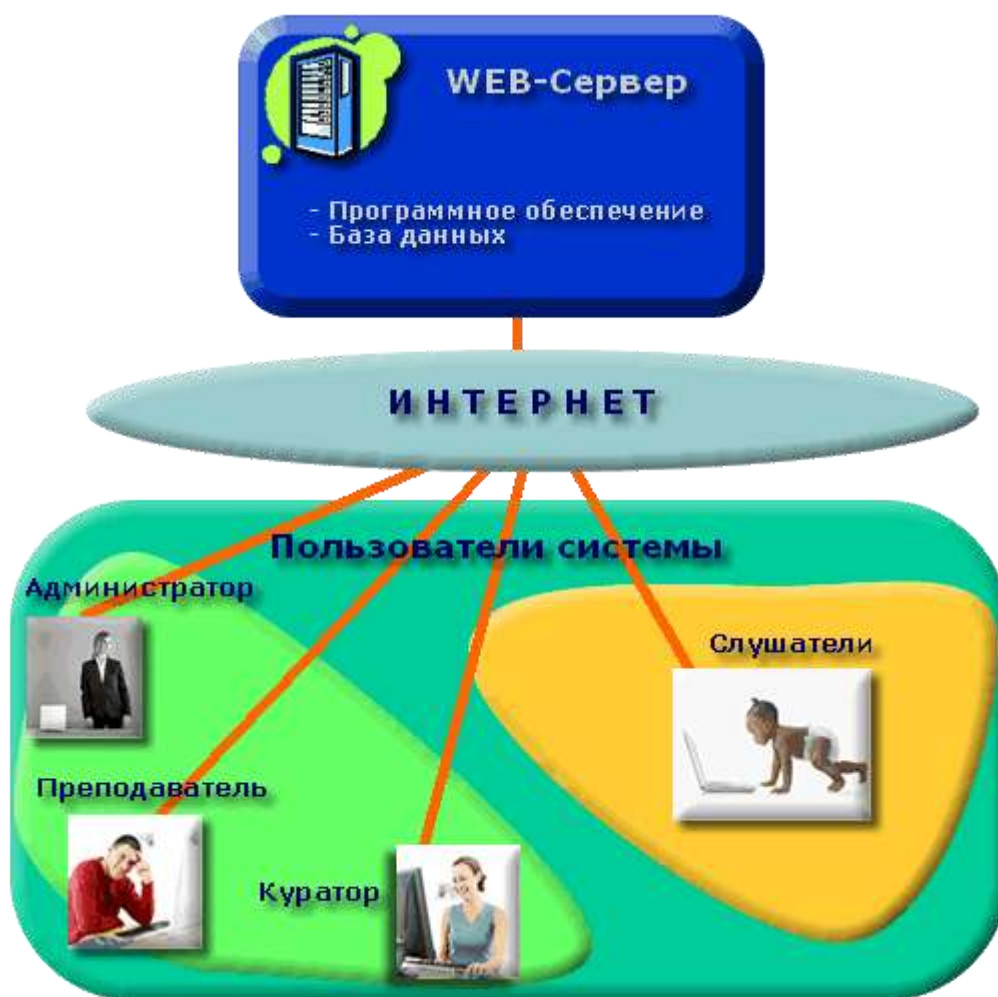


Рис. 1.1 Пользовательские системы дистанционного обучения.

Система реализуется по принципу «Тонкого клиента». Всё программное обеспечение и данные хранятся на WEB-сервере. Пользователям необходимо лишь иметь компьютер, подключённый к интернету и WEB-браузер. Никакого программного обеспечения устанавливать на компьютерах пользователя не надо. Работа с СДО почти ничем не отличается от работы с другими интернет-ресурсами. Все пользователи осуществляют вход в систему со своих компьютеров, используя персональные логины и пароли, и имеют доступ к своим разделам программы.

1.2 Электронный учебник

Электронный учебник - это методический комплекс, предназначенный для изучения курса материала. Он является интегрированным средством, содержащим теорию, практику, задачи и другие компоненты.

При создании электронного учебника авторам необходимо учитывать, что он должен содержать минимум текста, потому что длительное чтение текста с экрана компьютера утомительно и снижает восприятие новой информации. Очень важно правильно подобрать шрифты текста, заранее ознакомившись с теорией шрифтов. Электронный учебник должен содержать как можно больше графических изображений, т.к. усвоение и восприятие информации при чтении иллюстративных учебников намного выше. Важно корректно подобрать цветовую гамму, чтобы при изучении материала цвета не напрягали ученика, а наоборот, успокаивали [4,5].

Также рекомендуется использовать в электронном учебнике видео и аудио фрагменты, гиперссылки на элементы учебника и на другую информацию в интернете. Использование видео фрагментов позволяет передать информацию в динамике и увеличить заинтересованность учащихся, особенно школьников, и этим повысить усвояемость. Использование аудио фрагментов помогает ЭУ приблизиться к привычным методам обучения и также активизировать слуховые центры головного мозга.

Создание электронного учебника является трудоемким процессом. Обычно его созданием занимается группа специалистов. Я рекомендую создание электронного учебника разбить на следующие этапы:

- Оценка аудитории, для которой разрабатывается данный учебник.
- Сбор и подготовка текстовой и графической информации
- Подготовка содержания
- Непосредственное создание электронного учебника.
- Процесс создания электронного учебника очень отличается от процесса создания традиционного учебника.

Электронный учебник можно назвать самоучителем, т.к. он рассчитан на самостоятельное обучение. Иллюстративный электронный учебник, состоящий из текста, графической, аудио и видеоинформации, позволяет проводить индивидуализированное обучение. В отличие от традиционного учебника он приближает обучение к обучению с преподавателем. Как было отмечено выше, интерактивный электронный самоучитель, содержащий графические фрагменты, способен привлекать школьников и молодежь намного успешней, чем традиционный. Электронный учебник является компактным, он может хранить большое количество информации. Следует отметить, что на одном переносном носителе, таком как внешний винчестер, флешка и даже CD/DVD диск, можно хранить целую библиотеку традиционных учебников. Каждый преподаватель хорошего уровня может легко настроить электронный учебник на свою программу обучения, легко редактировать какие-то его части или комбинировать несколько электронных учебников в один. Преподаватель в любое время суток может передать учебник ученикам, разослав его по электронной почте или по скайпу. Также сделать его общедоступным для своих учеников, загрузив его в интернет. ЭУ не портится со временем в отличие от традиционного учебника. Создание и публикация электронного средства обучения не связаны с большими затратами. Электронные учебники могут иметь встроенную систему тестирования, что дает возможность учащемуся проверить как он усвоил

пройденный материал. Учащийся и учитель могут мгновенно найти нужную им часть информации благодаря поисковой системе. При обучении можно легко улучшить читабельность ЭУ, увеличив размер шрифта или изменив его цвет [6,7].

В современных условиях наблюдаются некоторые проблемы, связанные с образованием:

Большинство учебных заведений с высоким уровнем обучения находятся в центральных городах, поэтому для людей, живущих в регионах, является проблемой повышение уровня образования.

Специалисты имеют довольно тяжелый график работы, поэтому у них не остается времени на повышение своей квалификации.

В группе учащихся часто наблюдается разный уровень усвояемости, и многие учащиеся желают иметь собственный темп обучения.

У большинства молодежи не хватает финансов на учебу, на улучшение уровня своих знаний.

Дистанционное образование является решением вышеперечисленных проблем. При дистанционном обучении большую часть времени обучаемый занимается самостоятельно, поэтому возникает потребность в использовании электронных учебников. Как мы видим, электронный учебник является основным учебным средством дистанционного образования, основным средством для повышения квалификации и образовательного уровня, получения второго образования, возможности получения диплома престижного университета, колледжа и т.п.

На сегодняшний день дистанционное образование становится все более и более популярным во всем мире. Открываются сети институтов и школ, использующих систему дистанционного образования. Это требует создания большого количества электронных образовательных средств, в частности, электронных учебников [7].

Наряду с большим количеством преимуществ следует отметить и недостатки электронных средств:

Необходимость дополнительного оборудования, в основном, компьютера и колонок.

Утомляемость, вызванная чтением с экрана, связанная с непривычностью и новизной.

Дети, использующие только электронные учебники, теряют возможность общения со своими сверстниками, поэтому ЭУ должны использоваться для детей только как дополнительный метод обучения. Время нахождения за компьютером должно быть строго лимитировано и чередоваться с подвижными играми, желательно на воздухе.

На сегодняшний день цены на электронные средства все еще значительно превышают цены на традиционные учебники.

Все вышесказанное позволяет утверждать, что несмотря на определенные недостатки электронных учебников, они являются мощным фактором повышения образовательного уровня всех слоев общества, что необходимо в сегодняшних условиях.

1.3 Обзор функциональных возможностей и типов систем дистанционного обучения

Возможности системы дистанционного обучения:

- управление электронным, очным и смешанным обучением;
- тестирование знаний, отработка умений и навыков;
- оценка и анализ результатов обучения;
- подготовка учебных курсов и программ обучения;
- управление библиотекой учебных материалов.

В каждой организации, безусловно, есть свои специфика и принципы решения задач ДО. На этапе выбора СДО необходимо оценить существующие на рынке системы и платформы в виде их применения в вашей организации.

Основные критерии выбора системы дистанционного обучения

Функциональность. Наличие (отсутствие) необходимых функций для решения поставленных задач.

Масштабируемость. Возможность расширения круга пользователей, добавление программ обучения.

Модульность. В современных СДО курс может представлять собой набор модулей или блоков учебного материала, которые могут быть использованы в других курсах.

Кросс-платформенность. СДО не должна быть привязана к какой-либо операционной системе или среде. Пользователи должны использовать стандартные средства без загрузки дополнительных модулей.

Мультимедийность. Возможность использования в качестве контента не только текстовых, гипертекстовых и графических файлов, но и звуковых, видеофайлов, анимации и 3D-графики.

Система проверки знаний. Система позволяет в режиме реального времени оценить знания слушателей. Обычно такая система включает в себя тесты, задания и контроль активности пользователей [7,14].

Качество технической поддержки. Возможность поддержки работоспособности системы, устранение ошибок с привлечением разработчика и специалистами собственной службы поддержки.

Перспективы развития. СДО должна быть развивающейся средой. Постоянно должны выходить новые, улучшенные версии системы с поддержкой новых технологий и стандартов.

Стоимость. Бесплатная или платная СДО? Стоимость коммерческой системы складывается из стоимости самой системы, затрат на ее внедрение и сопровождение, наличие (отсутствие) ограничений по количеству лицензий на слушателей, разработку курсов.

Стоит отметить, что СДО – это не только программная платформа и оборудование. СДО – это люди и бизнес-процессы.

Основными пользователями системы дистанционного обучения являются: администратор, куратор, преподаватель и обучаемые (слушатели).

Администратор организует работу учебного центра.

Куратор организует учебный процесс в группе. Он формирует учебные группы на основе заявок на обучение, полученных по Интернет, по корпоративным каналам связи или любым иным способом.

Преподаватель готовит учебно-методические материалы по учебным курсам, проводит групповые и индивидуальные консультации со слушателями. Совместно с куратором группы принимает решение об аттестации слушателей на основании результатов очного или заочного, автоматизированного или автоматического контроля и тестирования.

Слушатели объединяются в учебные группы. Все слушатели учебной группы изучают одни и те же учебные курсы. Количество слушателей в учебной группе не ограничивается. К группе прикрепляется куратор, отвечающий за организацию учебного процесса, и преподаватели, ведущие учебные курсы. Сложный курс в группе могут вести несколько преподавателей.

Таким образом, система управления контентом обучения реализует следующие основные функции:

- 1.Разработка учебного контента;
- 2.Сохранение учебного контента в репозитории;
- 3.Регистрация слушателей;
- 4.Выявление потребностей в обучении;
- 5.Формирование групп слушателей;
- 6.Формирование учебных курсов для групп слушателей в соответствии с потребностью в обучении;
- 7.Доставка учебного контента слушателям;
- 8.Проверка знаний.

На сегодняшний день существует большое количество СДО. У каждой из них есть свои преимущества и недостатки, но в общем целом к каждой из них применяются следующие основные требования:

- **Функциональность.** Наличие различных методов преподнесения учебного материала, реализованные в СДО для удовлетворения любой методики обучения.
- **Стабильность.** Параметр, который отображает степень отказоустойчивость системы при любой степени активности пользователей.
- **Прозрачность.** Данный параметр отображает удобство использования системы, как для пользователя, так и для администратора.
- **Модульность.** В современных системах учебный курс может представлять из себя набор модулей и блоков учебного материала, которые могут быть использованы в других курсах.
- **Мультимедийность.** Система должна поддерживать любой вид мультимедиа контента.
- **Кроссплатформенность.** Система должна быть не зависима от конкретной операционной системы, как на стороне сервера, так и на стороне клиента.
- **Стоимость.** Стоимость складывается из стоимости самой системы, затрат на ее внедрение[8,15].

На рынке систем дистанционного обучения все продукты делятся на две группы:

- В случае необходимости внесения, каких либо изменений в систему производитель также может взимать дополнительную сумму.
- Проекты с открытым исходным кодом, бесплатные системы которые дорабатываются всеми участниками проекта. В таких системах любой желающий может вносить изменения в системе и делиться с ними со всеми участниками проекта, что положительно сказывается на развитии всей системы в целом.

ГЛАВА 2. СРЕДСТВА И СРЕДА РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

2.1 Среда разработки под ОС Android

Android - бесплатная операционная система, основанная на Linux с интерфейсом программирования Java.

Разработка приложений ведется на Java, нам нужно скачать и установить соответствующее SDK, называемое еще JDK (если, конечно, оно уже не установлено) [7].

Набор программ и классов JDK содержит:

- компилятор `javac` из исходного текста в байт-коды; интерпретатор `java`, содержащий реализацию JVM;
- облегченный интерпретатор `jre` (в последних версиях отсутствует);
- программу просмотра апплетов `appletviewer`, заменяющую браузер;
- отладчик `jdt`;
- дизассемблер `javap`;
- программу архивации и сжатия `jar`;
- программу сбора документации `javadoc`;
- программу `javah` генерации заголовочных файлов языка C;
- программу `javakey` добавления электронной подписи;
- программу `native2ascii`, преобразующую бинарные файлы в текстовые;
- программы `rmic` и `rmiregistry` для работы с удаленными объектами;
- программу `serialver`, определяющую номер версии класса;
- программу `native2ascii`, преобразующую бинарные файлы в текстовые;
- программы `rmic` и `rmiregistry` для работы с удаленными объектами;
- программу `serialver`, определяющую номер версии класса
- библиотеки и заголовочные файлы "родных" методов;
- библиотеку классов Java API (Application Programming Interface).

В прежние версии JDK включались и отладочные варианты исполнимых программ: `javac_g`, `java_g` и т. д.

Android - бесплатная операционная система, основанная на Linux с интерфейсом программирования Java[9].

Android поддерживает 2D и 3D-графику, используя библиотеки OpenGL, а также хранение данных в базе данных SQLite [7].

Каждое Android-приложение запускается в своем собственном процессе. Поэтому приложение изолировано от других запущенных приложений, и неправильно работающее приложение не может беспрепятственно навредить другим приложениям.

Android SDK - Android SDK включает в себя инструменты, необходимые для разработки Android-приложений. Комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

Eclipse - свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается EclipseFoundation. На данный момент является наиболее удобным средством разработки для Android.

ADT плагин - по умолчанию Eclipse не особо годится для разработки Android-приложений. ADT - плагин, который настраивает среду разработки для использования Android SDK и добавляет возможность удобной разработки. Плагин довольно легко устанавливает в среду Eclipse.

SDKmanager (выбор платформы) – здесь мы выбираем необходимые нам компоненты для разработки приложений, а именно: платформа (API), утилиты (оптимизаторы для определенных устройств, набор готовых примеров и т.д.), тулсы (прямая отладка на ANDROID-устройстве, интеграция дополнительных возможностей и т.д.).

AVD (android virtual device)—это эмулятор Android-смартфона, на который Eclipse сможет устанавливать, созданные нами приложения, и запускать их там [10].

Таким образом, в совокупности, мы получаем среду разработки с необходимым набором плагинов и виртуальной машины для отладки и проверки скомпилированных приложений. Далее можно приступить непосредственно к разработке приложений [10,16].

Ключевые особенности Android

Android основан на Linux. Между приложением и ядром лежит слой API и слой библиотек на нативном коде. Приложение выполняется на виртуальной машине Java (DalvikVirtualMachine).

В Android можно запускать много приложений. Но одно из них есть главным и занимает экран.

Каждый экран пользовательского интерфейса представлен классом Activity в коде. Различные Activity содержатся в процессах. Activity может даже жить дольше процесса. Activity может быть приостановлена и запущена вновь с сохранением всей нужной информации.

В Android можно запускать много приложений. Но одно из них есть главным и занимает экран.

Android использует специальный механизм описания действий основанный на Intent. Когда нужно выполнить действие (сделать звонок, послать письмо, показать окно), вызывается Intent.

Также Android содержит сервисы подобные демонам в Linux для выполнения нужных действий в фоновом режиме (например, проигрывание музыки).

Для обмена данными между приложениями используются Content providers (провайдеры содержимого).

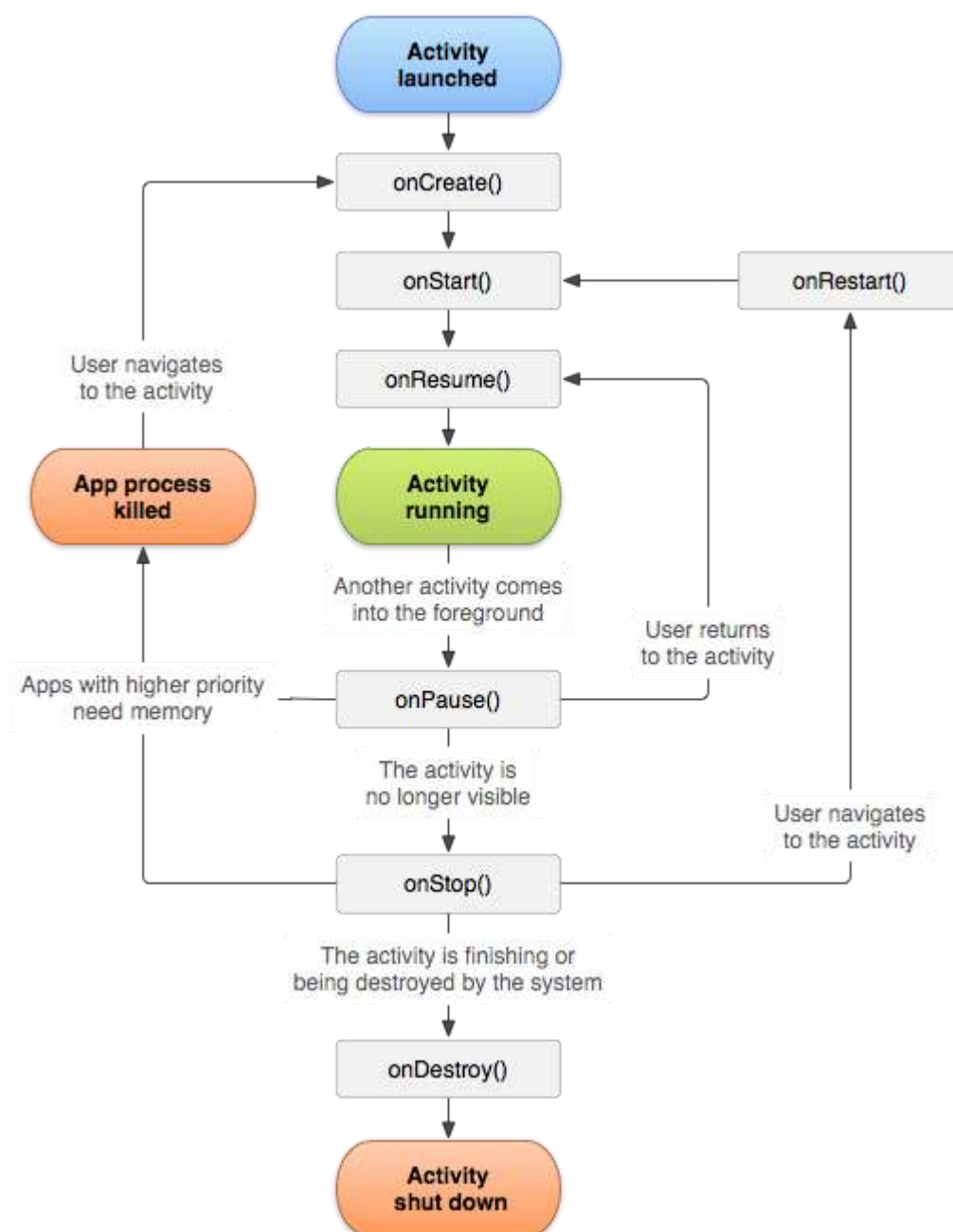


Рис.2.1. Жизненный цикл ПО в Android

Приложения для Android пишутся на языке программирования Java. Инструменты Android SDK (Software Development Kit – комплект разработки программного обеспечения) компилируют написанный вами код — и все требуемые файлы данных и ресурсов — в файл APK – программный пакет Android, который представляет собой файл архива с расширением .apk. В файле APK находится все, что требуется для работы Android-приложения, и

он позволяет установить приложение на любом устройстве под управлением системы Android.

Каждое приложение Android, установленное на устройстве, работает в собственной "песочнице" (изолированной программной среде):

- операционная система Android представляет собой многопользовательскую систему Linux, в которой каждое приложение является отдельным пользователем;
- по умолчанию система назначает каждому приложению уникальный идентификатор пользователя Linux (этот идентификатор используется только системой и неизвестен приложению); система устанавливает полномочия для всех файлов в приложении, с тем чтобы доступ к ним был разрешен только пользователю с идентификатором, назначенным этому приложению;
- у каждого процесса имеется собственная виртуальная машина (VM), так что код приложения выполняется изолированно от других приложений;
- по умолчанию каждое приложение выполняется в собственном процессе Linux. Android запускает процесс, когда требуется выполнить какой-либо компонент приложения, а затем завершает процесс, когда он больше не нужен либо когда системе требуется освободить память для других приложений.

Таким образом система Android реализует принцип предоставления минимальных прав. То есть каждое приложение по умолчанию имеет доступ только к тем компонентам, которые ему необходимы для работы, и ни к каким другим. Благодаря этому формируется исключительно безопасная среда, в которой приложение не имеет доступа к недозванным областям системы.

Однако у приложения есть варианты предоставления своих данных другим приложениям и доступа к системным службам:

- двум приложениям можно назначить один идентификатор пользователя Linux. В этом случае каждый из них сможет обращаться к файлам другого приложения. Для экономии ресурсов системы также можно сделать так,

чтобы приложения с одинаковым идентификатором пользователя выполнялись в одном процессе Linux и использовали одну ВМ (приложения также должны быть подписаны одним сертификатом);

- приложение может запросить разрешение на доступ к данным устройства, например к контактам пользователя, SMS-сообщениям, подключаемой карте памяти (SD-карте), камере, Bluetooth и др. Все разрешения должны предоставляться приложению при его установке[9].

Основные компоненты Android

Activity (Активность, Деятельность) — представляет собой экран пользовательского интерфейса, который представлен классом Activity и разметкой в виде XML-файла (разметку можно создать и программно, но теперь так не делают). Android-приложение способно состоять из нескольких форм (Activity) и может переключаться между ними во время выполнения приложения

Intents (Намерения) — асинхронные сообщения, которые позволяют приложению запросить функции из других служб или действий. Приложение может делать прямые запросы службе или действию (явное намерение) или запросить у Android зарегистрированные службы и приложения (неявное намерение). Для примера, приложение может запросить через Intent приложение Контакты для получения данных или запустить браузер при нажатии ссылки в тексте вашего приложения[11].

Views (Представление, Вид, Элемент управления, Компонент) — пользовательский интерфейс, создаваемый виджетами классов. Это знакомые нам элементы управления: кнопки, текстовые поля, флажки, переключатели и т.п.

Services (Службы) — выполняют фоновые задачи без предоставления пользовательского интерфейса (например, проигрывание музыки). Они могут уведомлять пользователя через систему уведомлений Android.

Content Provider (Контент-провайдер) — предоставляет данные приложениям. С помощью контент-провайдера ваше приложение может обмениваться данными с другими приложениями.

Broadcast Receiver (Приемник широковещательных сообщений/запросов) — принимает системные сообщения и неявные намерения, может использоваться для реагирования на изменение состояния системы. Приложение может регистрироваться как приемник определенных событий и может быть запущено, если такое событие произойдет.

Активация компонентов

Компоненты трех из четырех возможных типов — операции, службы и приемники широковещательных сообщений — активируются асинхронным сообщением, которое называется Intent (намерение). Объекты Intent связывают друг с другом отдельные компоненты во время выполнения, будь то это компоненты вашего или стороннего приложения (эти объекты Intent можно представить себе в виде мессенджеров, которые посылают другим компонентам запрос на выполнение действий).

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

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

дать пользователю возможность выбрать контакт и вернуть его вам — в ответном сообщении Intent будет содержаться URI, указывающий на выбранный контакт).

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

Компоненты четвертого типа – поставщики контента – сообщениями Intent не активируются. Они активируются по запросу от ContentResolver. Процедура определения контента (content resolver) обрабатывает все прямые транзакции с поставщиком контента, с тем чтобы этого не пришлось делать компоненту, который выполняет транзакции с поставщиком. Вместо этого он вызывает методы для объекта ContentResolver. Это формирует слой, абстрагирующий (в целях безопасности) поставщика контента от компонента, запрашивающего информацию[10,11].

Для активации компонентов каждого типа имеются отдельные методы:

- Можно запустить операцию (или определить для нее какое-то новое действие), передав объект Intent методу startActivity() или startActivityForResult() (если требуется, чтобы операция вернула результат).
- Можно запустить службу (либо выдать работающей службе новые инструкции), передав объект Intent методу startService(). Либо можно установить привязку к службе, передав объект Intent методу bindService().
- Можно инициировать рассылку сообщений, передав объект Intent таким методам, как sendBroadcast(), sendOrderedBroadcast() и sendStickyBroadcast().
- Можно выполнить запрос к поставщику контента, вызвав метод query() для объекта ContentResolver.

Файл манифеста

Для запуска компонента приложения системе Android необходимо знать, что компонент существует. Для этого она читает файл AndroidManifest.xml приложения (файл манифеста). В этом файле, который

должен находиться в корневой папке приложения, должны быть объявлены все компоненты приложения.

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

- указание всех полномочий пользователя, которые требуются приложению, на пример разрешения на доступ в Интернет или на чтение контактов пользователя;
- объявление минимального уровня API, требуемого приложению, с учетом того, какие API-интерфейсы оно использует;
- объявление аппаратных и программных функций, которые нужны приложению или используются им, например камеры, службы Bluetooth или сенсорного экрана;
- указание библиотек API, с которыми необходимо связать приложение (отличные от API-интерфейсов платформы Android), например библиотеки Google Maps ;
- и многое другое.

Ресурсы приложения

Приложение Android состоит не только из кода — ему необходимы такие существующие отдельно от исходного кода ресурсы, как изображения, аудиофайлы и все, что связано с визуальным представлением приложения. Например, необходимо определять анимацию, меню, стили, цвета и макет пользовательских интерфейсов операций в файлах XML. Используя ресурсы приложения, можно без труда изменять его различные характеристики, не меняя код, а, кроме того, — путем предоставления наборов альтернативных ресурсов — можно оптимизировать свое приложение для работы с различными конфигурациями устройств (например, для различных языков или размеров экрана).

Для каждого ресурса, включаемого в проект Android, инструменты SDK задают уникальный целочисленный идентификатор, который может использоваться, чтобы сослаться на ресурс из кода приложения или из

других ресурсов, определенных в XML. Например, если в вашем приложении имеется файл изображения с именем `logo.png` (сохраненный в папке `res/drawable/`), инструменты SDK сформируют идентификатор ресурса под именем `R.drawable.logo`, с помощью которого на изображение можно будет ссылаться и вставлять его в пользовательский интерфейс.

Один из наиболее важных аспектов предоставления ресурсов отдельно от исходного кода заключается в возможности использовать альтернативные ресурсы для различных конфигураций устройств. Например, определив строки пользовательского интерфейса в XML, вы сможете перевести их на другие языки и сохранить эти переводы в отдельных файлах. Затем по квалификатору языка, добавленному к имени каталога ресурса (скажем `res/values-fr/` для строк на французском языке), и выбранному пользователем языку система Android применит к вашему пользовательскому интерфейсу строки на соответствующем языке[11,12].

Android поддерживает разные квалификаторы для соответствующих ресурсов. Квалификатор представляет собой короткую строку, которая включается в имена каталогов ресурсов с целью определения конфигурации устройства, для которой эти ресурсы следует использовать. В качестве другого примера можно сказать, что для своих операций следует создавать разные макеты, которые будут соответствовать размеру и ориентации экрана устройства. Например, когда экран устройства имеет книжную ориентацию (расположен вертикально), кнопки в макете можно также размещать по вертикали, а когда экран развернут горизонтально (альбомная ориентация), кнопки следует размещать по горизонтали. Чтобы при изменении ориентации экрана изменялся макет, можно определить два разных макета и применить соответствующий квалификатор к имени каталога каждого макета. После этого система будет автоматически применять соответствующий макет в зависимости от ориентации устройства.

Обзор пользовательского интерфейса

Все элементы интерфейса пользователя в приложении Android создаются с помощью объектов View и ViewGroup. Объект View формирует на экране элемент, с которым пользователь может взаимодействовать. Объект ViewGroup содержит другие объекты View (и ViewGroup) для определения макета интерфейса.

Android предоставляет коллекцию подклассов View и ViewGroup, которая включает в себя обычные элементы ввода (такие как кнопки и текстовые поля) и различные модели макет (такие как линейный или относительный макет)[11].

Макеты пользовательского интерфейса

Пользовательский интерфейс для каждого компонента вашего приложения определяется с помощью иерархии объектов View и ViewGroup, как показано на рисунке 3. Каждая группа просмотра представляет собой невидимый контейнер, в котором объединены дочерние виды, причем дочерние виды могут представлять собой элементы ввода или другие виджеты, которые составляют часть пользовательского интерфейса. Эта древовидная иерархия может быть настолько простой или сложной, насколько требуется (чем проще, тем лучше для производительности).

Чтобы объявить свой макет, можно создать экземпляры объектов View в коде и запустить построение дерева, но самый простой и наиболее эффективный способ — определение макета с помощью файла XML. XML позволяет создавать удобочитаемую структуру макета, подобно HTML.

Имя элемента XML для вида соответствует классу Android, к которому он относится. Так, элемент `<TextView>` создает виджет `TextView` в пользовательском интерфейсе, а элемент `<LinearLayout>` создает группу просмотра `LinearLayout`.

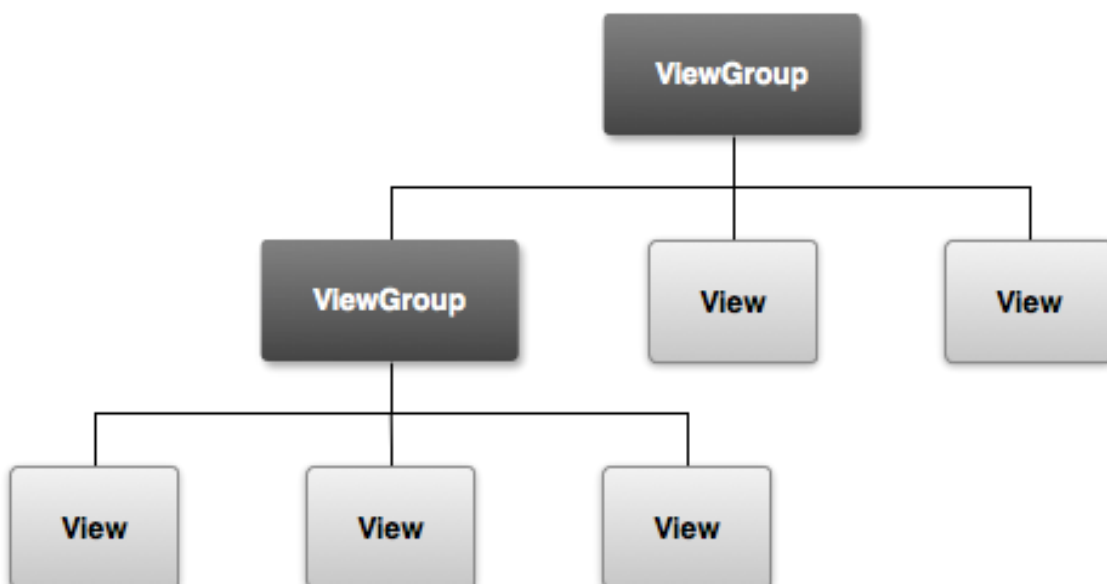


Рис. 2.2. Иллюстрация иерархии, которая определяет макет интерфейса.

Например, простой вертикальный макет с текстом и кнопкой выглядит следующим образом:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextViewandroid:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a TextView"/>
    <Buttonandroid:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a Button"/>
</LinearLayout>
  
```

При загрузке ресурсов макетов в приложение Android инициализирует каждый узел макета в объект режима выполнения, который можно использовать для определения дополнительного поведения, запроса состояния объекта или изменения макета.

Компоненты пользовательского интерфейса

Не обязательно создавать все элементы пользовательского интерфейса с помощью объектов View и ViewGroup. Android предоставляет несколько компонентов приложений, которые содержат стандартный макет пользовательского интерфейса, где остается лишь определить содержимое. Каждый из этих компонентов пользовательского интерфейса содержит уникальный набор API, который описан в соответствующих документах, таких как Строка действий, Диалоги и Уведомления о состоянии[10,12].

Операции

Activity — это компонент приложения, который выдает экран, и с которым пользователи могут взаимодействовать для выполнения каких-либо действий, например набрать номер телефона, сделать фото, отправить письмо или просмотреть карту. Каждой операции присваивается окно для прорисовки соответствующего пользовательского интерфейса. Обычно окно отображается во весь экран, однако его размер может быть меньше, и оно может размещаться поверх других окон.

Как правило, приложение состоит из нескольких операций, которые слабо связаны друг с другом. Обычно одна из операций в приложении обозначается как «основная», предлагаемая пользователю при первом запуске приложения. В свою очередь, каждая операция может запустить другую операцию для выполнения различных действий. Каждый раз, когда запускается новая операция, предыдущая операция останавливается, однако система сохраняет ее в стеке («стек переходов назад»). При запуске новой операции она помещается в стек переходов назад и отображается для пользователя. Стек переходов назад работает по принципу «последним вошёл — первым вышел», поэтому после того как пользователь завершил

текущую операцию и нажал кнопку Назад, текущая операция удаляется из стека (и уничтожается), и возобновляется предыдущая операция. (Подробные сведения о стеке переходов назад представлены в статье Задачи и стек переходов назад.)

Когда операция останавливается по причине запуска новой операции, для уведомления об изменении ее состояния используются методы обратного вызова жизненного цикла операции. Существует несколько таких методов, которые может принимать операция вследствие изменения своего состояния — создание операции, ее остановка, возобновление или уничтожение системой; также каждый обратный вызов представляет возможность выполнить определенное действие, подходящее для соответствующего изменения состояния. Например, в случае остановки операция должна освободить любые крупные объекты, например, подключение к сети или базе данных. При возобновлении операции вы можете повторно получить необходимые ресурсы и возобновить выполнение прерванных действий. Такие изменения состояния являются частью жизненного цикла операции.

Создание операции

Чтобы создать операцию, сначала необходимо создать подкласс класса Activity (или воспользоваться существующим его подклассом). В таком подклассе необходимо реализовать методы обратного вызова, которые вызывает система при переходе операции из одного состояния своего жизненного цикла в другое, например при создании, остановке, возобновлении или уничтожении операции. Вот два наиболее важных метода обратного вызова:

onCreate()

Этот метод необходимо обязательно реализовать, поскольку система вызывает его при создании вашей операции. В своей реализации вам необходимо инициализировать ключевые компоненты операции. Наиболее важно именно здесь вызвать setContentView() для определения макета пользовательского интерфейса операции.

onPause()

Система вызывает этот метод в качестве первого признака выхода пользователя из операции (однако это не всегда означает, что операция будет уничтожена). Обычно именно здесь необходимо применять любые изменения, которые должны быть сохранены помимо текущего сеанса работы пользователя (поскольку пользователь может не вернуться назад).

Существуют также и некоторые другие методы обратного вызова жизненного цикла, которые необходимо использовать для того, чтобы обеспечить плавный переход между операциями, а также для обработки непредвиденных нарушений в работе операции, в результате которых она может быть остановлена или даже уничтожена. Более подробное описание всех методов обратного вызова жизненного цикла представлены в разделе, посвященном управлению жизненным циклом операций[12,15].

Реализация пользовательского интерфейса

Для реализации пользовательского интерфейса операции используется иерархия представлений — объектов, полученных из класса View. Каждое представление отвечает за определенную прямоугольную область окна операции и может реагировать на действия пользователей. Например, представлением может быть кнопка, нажатие на которую приводит к выполнению определенного действия[15].

В Android предусмотрен набор уже готовых представлений, которые можно использовать для создания дизайна макета и его организации. Виджеты — это представления с визуальными (и интерактивными) элементами, например, кнопками, текстовыми полями, чекбоксами или просто изображениями. Макеты — это представления, полученные из класса ViewGroup, обеспечивающие уникальную модель компоновки для своих дочерних представлений, таких как линейный макет, сетка или относительный макет. Также можно создать подкласс для классов View и ViewGroup (или воспользоваться существующими подклассами), чтобы

создать собственные виджеты и макеты, и затем применить их к макету своей операции.

Чаще всего для задания макета с помощью представлений используется XML-файл макета, сохраненный в ресурсах приложения. Таким образом вы можете хранить дизайн пользовательского интерфейса отдельно от исходного кода, который служит для задания поведения операции. Чтобы задать макет в качестве пользовательского интерфейса операции, можно использовать метод `setContentView()`, передав в него идентификатор ресурса для макета. Однако вы также можете создать новые `View` в коде вашей операции и создать иерархию представлений.

2.2 Разработка приложений для платформы Android

В современном мире существует огромное разнообразие мобильных устройств. Мобильный — значит переносимый, т.е. сюда мы относим не только сотовые телефоны, но и планшеты, плееры и прочие устройства.

Как и у любого компьютерного устройства, у мобильных есть операционные системы. Мобильные ОС достаточно распространены давайте рассмотрим каждую из наиболее известных в порядке убывания их популярности.



Рис. 2.3. Мобильные ОС

Android

Это операционная система для смартфонов, планшетов, приставок, нетбуков и других устройств. Является собственностью компании Google и признана самой популярной в мире. 22 октября 2008 года Google объявила об открытии онлайн-магазина приложений для ОС Android — Android Market. Количество приложений для Андроид в данном магазине превышает 1,43 млн. Первая версия ОС вышла в свет в 2008 году на смартфоне HTC и с тех пор неустанно обновляется[9,14].

Мобильные ОС — достоинства Android:

- быстрая интеграция с сервисами google
- наличие файловой системы
- абсолютная независимость от аппаратной начинки мобильного устройства
- Android является системой с открытым кодом
- многозадачность
- легкость установки приложений с разных ресурсов
- широкие возможности индивидуализации
- отсутствие ограничений при выборе мобильного оператора
- поддержка Flash-проигрывателя
- обновление через Интернет
- возможность замены/удаления дефолтных приложений

iOS

iOS — операционная система для смартфонов, планшетов и др. устройств, разработанная компанией Apple исключительно для своих устройств.

Пользовательский интерфейс iOS основан на концепции прямого взаимодействия с использованием жестов «мультикас». Элементы управления интерфейсом состоят из ползунков, переключателей и кнопок. Впервые операционная система была представлена в 2007 году.

Мобильные ОС — достоинства iOS:

- качество поставляемых приложений

- доступность обновлений сразу после выпуска новой версии ОС для всех устройств одновременно
- продолжительная поддержка старых устройств
- разработчики в первую очередь анонсируют свои приложения для iOS
- удобство использования, удобство интерфейса
- акцент на надежность и качество ОС
- семейный доступ для покупок в Appstore
- возможность объединить работу обновленных до последней ОС мобильных устройств

Windows Phone

Windows Phone OS — мобильная операционная система, выпущенная в 2010 году корпорацией Майкрософт. Данную операционку можно легко узнать по своеобразным “живым” плиткам на начальном экране. Не так популярна как 2 предыдущие, но тем не менее имеет ряд своих преимуществ: Windows — самая распространенная операционная система в мире, и устройства, работающие на ней, легко синхронизируются с компьютером, ноутбуком, планшетом, работающим на Windows нет проблем с оперативной памятью разнообразие приложений по умолчанию ***Blackberry***.

Blackberry OS — операционная система для устройств Blackberry на основе QNX, впервые выпущенная в 2009 году. По своей сути ориентирована на бизнес-сегмент, корпоративных пользователей.

Мобильные ОС — достоинства Blackberry OS:

- технология, которая даёт возможность более эффективно вести дела, оперативно управлять собственным бизнесом и рационально использовать человеческие ресурсы;
- возможность работать удалённо, вне офиса, возможность вовремя получить договор, презентацию, отчёт, передать важные документы в работу, не промедлив ни минуты;
- постоянная синхронизация с рабочим местом;

– высокая безопасность корпоративных коммуникаций. Даже если аппарат потерялся или разбился, вы не потеряете своей переписки, своих контактов, переписку и другую информацию с утерянного аппарата можно дистанционно удалить.

– позволяет быстро просматривать на аппарате вложения больших размеров практически любых форматов

Существует также ряд других операционных систем, например Symbian OS, BADA (Samsung) и другие. Но они не настолько популярны как вышеперечисленные[10].

Мобильные ОС их популярность

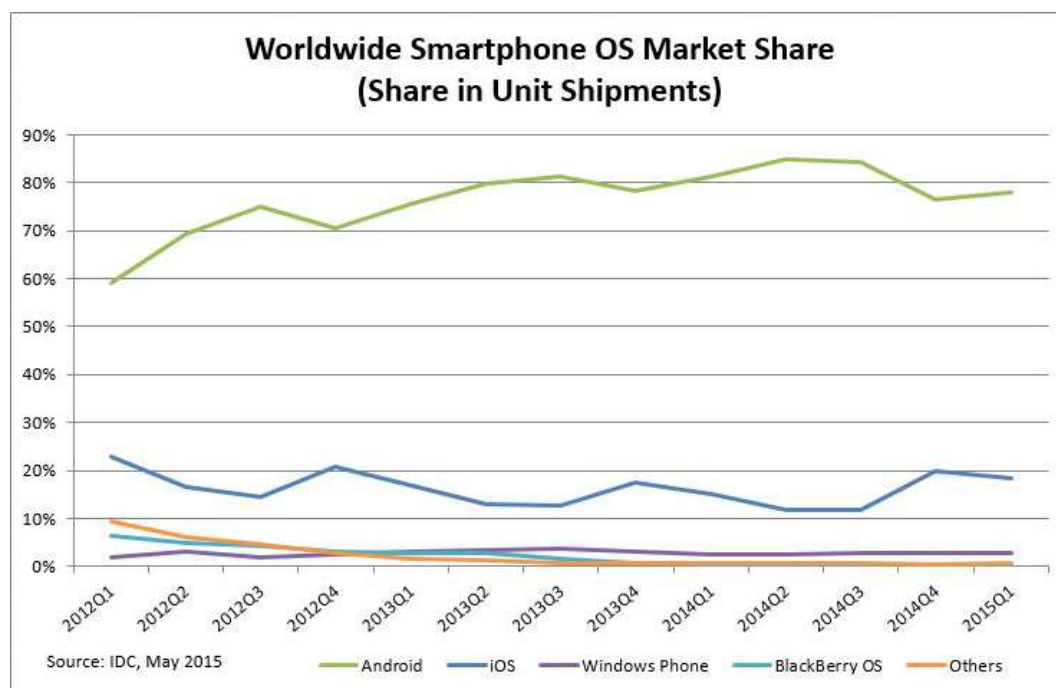


Рис 2.4. Мобильные ОС популярность

Рассмотрим мобильные устройства на платформах Android, iOS, Windows OS и BlackBerry, охарактеризуем их популярность в разных регионах.

В настоящее время мобильных устройств существует огромное множество. Мобильные телефоны, планшеты, электронные книги, нетбуки, умные часы и так далее. Давайте разберем на примерах.

На платформе Android работает ну очень много самых разных устройств по всему миру. Это в большинстве своем планшеты и смартфоны.

Штампуют их в невероятных количествах. Из наиболее известных — это планшеты и смартфоны Samsung, HTC, Huawei и другие. Особенность популярности Android в том, что его можно устанавливать на самые разнообразные устройства. Наиболее распространены такие устройства на территории Азии, Европы.

iOS может быть установлена только в устройствах компании Apple, что значительно уменьшает их количество по всему миру (как видно из графика). I pads, I phones, I pods, Macbooks, I Macs и I watch наиболее распространены на территории США и Европы.

Windows OS установлены практически на всех мобильных устройствах Nokia нового поколения, успешно пришел на замену Symbian. Чаще всего такие устройства покупают в США, Бразилии и Индии и Европе.

Устройства на BlackBerry OS применимы исключительно к устройствам данной компании, которые разрабатывают в Канаде и соответственно больше всего их именно там. Также его популярность можно отметить в США, Европе.

Надо понимать, что деление по регионам очень условное, поскольку динамика постоянно меняется с учетом развития прогресса в IT в настоящее время.

ГЛАВА 3. ПРОЦЕСС РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

3.1 Этап разработки интерфейсной части программы

В ходе изучения поставленной задачи выяснилось, что, разработке интерфейса стоит уделить немалое внимание. Плохо продуманный интерфейс может привести к нежеланию работать с программой независимо от примененных аппаратно–программных средств для ее реализации.

Интерфейс обеспечивает связь между пользователем и компьютером – он позволяет достигать поставленных целей, успешно находить решение поставленной задачи.

Интерфейс приложения был разработан в соответствии с принципами разработки интерфейсов. Также было проведено не большое исследование с целью выявить на рынке ЭУ примеры хорошего интерфейса. Была выведена закономерность, что приложения, интерфейс которых дружелюбен к пользователю, пользуются наибольшим спросом, чем те приложения, интерфейс которых был очень посредственным и не соответствовал основным принципам разработки интерфейса.

Для создания пользовательского интерфейса необходимо пользоваться следующими свойствами интерфейса для создания положительного восприятия у пользователей:

- Естественный интерфейс;
- Согласованность интерфейса;
- Дружественность интерфейса;
- Принцип «обратной связи»;
- Простота интерфейса;
- Гибкость интерфейса;
- Эстетическая привлекательность.

Также существуют правила, которые всегда должны придерживаться при создании мобильных приложений[9,15]:

- **Целеориентированность.** Вы создаете дизайн для конкретного пользователя. Сейчас в Сети большое количество данных о различных категориях пользователей, причем многие материалы — исследования, обзоры — доступны бесплатно. Изучение этой информации поможет вам создать приложение, которое полностью соответствует потребностям целевой аудитории.
- **Юзабилити.** Ваше приложение должно быть удобным и интуитивно понятным. К примеру, если вы хотите указать ссылку для перехода на сторонний ресурс, то оформляйте ее привычным образом — с помощью подчеркнутого голубого текста. Удобство и практичность — это первый шаг на пути к тому, чтобы ваша программа стала желанной для пользователя.
- Возможность ([аффóрданс](#)) и символичность. Аффóрданс — это функция. Для простоты снова воспользуемся приемом со ссылкой. Так, голубой подчеркнутый текст указывает на то, что клик по нему переведет пользователя по какому-то адресу. Подобные символы нужно использовать таким образом, чтобы пользователь не размышлял о том, что может означать тот или иной элемент интерфейса. Практичность и рациональность — наше все.
- **Обучаемость.** В идеале пользователь должен без труда догадываться, как работать с программой. Здесь приходят на помощь знакомые и привычные схемы оформления приложения. Они должны помочь человеку без проблем привыкнуть к программе.
- **Фидбек и время ответа.** Отклик приложения должен давать пользователю представление о том, выполнена задача или нет. Это может быть обычный звуковой сигнал или нечто более сложное — например, модальное окно.
- После завершения разработки и внедрения обеспечивается техническая и информационная поддержка, включая разработку обновлений, доработки и модернизацию продукта.

3.2 Этап апробации и тестирования разработанной программы

Устанавливаем приложение на наше мобильное устройство. Запускаем программу «Учебники». После запуска открывается главное окно нашей программы.



Рис.3.1. Главное окно мобильного приложения «Учебники»



Рис.3.2. Каталог книг мобильного приложения



Рис.3.3. Дополнительные компоненты мобильного приложения

При нажатии компоненты «Тесты» можно увидеть базу тестов по предметам. Выбрав по каталогу определенный предмет можно приступить к вопросам.

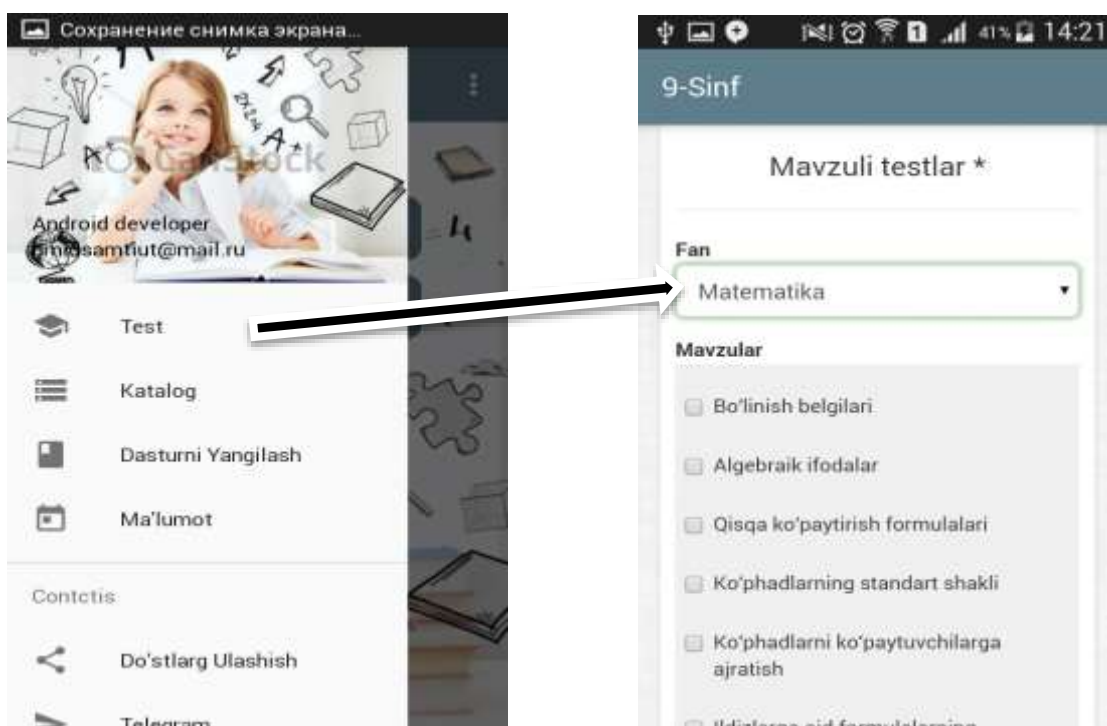


Рис.3.4. Окно каталога предметов и тем по предметам



Рис.3.5. Окно тестов по математике



Рис.3.6. Электронный учебник 9 класса «Алгебра» в мобильном приложении

4. Техническая безопасность

Анализ опасных и вредных факторов, возникающих при работе на компьютер. Исследовательская работа в рамках данного проекта заключается в выполнении многих этапов, практически все из которых проходят в тесном контакте с ЭВМ. Длительная работа инженера-программиста с компьютером сопряжена с целым рядом вредных и опасных факторов. Рассмотрим некоторые из них[13].

Постоянное напряжение глаз. Работа с компьютером характеризуется высокой напряженностью зрительной работы. В выполняемом исследовании значительный объем информации на разных стадиях обработки представлен в графической форме с большим количеством мелких деталей, что дает серьезную нагрузку на зрение. Постоянное напряжение глаз может привести к снижению остроты зрения. Экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии 600...700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов. Также для снижения утомляемости рекомендуется делать 15-минутные перерывы в работе за компьютером в течение каждого часа.

Влияние электростатических и электромагнитных полей. Большинство ученых считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Однако исчерпывающих данных относительно опасности воздействия излучения от мониторов на работающих с компьютерами не существует и исследования в этом направлении продолжаются.

Допустимые значения параметров неионизирующих электромагнитных излучений от монитора компьютера представлены в табл.1.

Максимальный уровень рентгеновского излучения на рабочем месте оператора обычно не превышает 10 мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10...100 мВт/м².

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

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

Для снижения воздействия этих видов излучения рекомендуется применить мониторы с пониженным уровнем излучения (MPR-II, TCO-92, TCO-99), устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

Длительное неизменное положение тела. Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей пользы человека-оператора.

Шум ухудшает условия труда, оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение

памяти, повышенную утомляемость, понижение аппетита, боли в ушах и так далее. В табл. 4.2 указаны предельные уровни звука в зависимости от категории тяжести и напряженности труда, являющиеся безопасными в отношении сохранения здоровья и работоспособности.

Техника безопасности при работе на ПК. Техника безопасности на ПК тесно связана с электробезопасностью. Большинство компьютеров имеют трехштырьковый разъем кабеля питания (в отличие от двухштырьковой вилки большинства бытовых приборов). Третий разъем - это так называемая «земля». В грамотно оборудованных компьютеризированных помещениях существует настоящая земля — заземляющий контур, который через металлическую ленту выводится на заземляющий штырь. Все это довольно громоздкое и дорогостоящее оборудование и часто разъем заземления не используется или используется «земляная» фаза обычной электросети. Результатом неправильного заземления могут быть удары статического напряжения от корпуса ПК. Частенько из-за этого сгорают сетевое оборудование компьютеров. Следует соблюдать не только правила электробезопасности, но и следить за состоянием здоровья обслуживающего персонала и пользователей ПК, защищая их от вредного воздействия электромагнитных излучений, заботясь о зрении, так как именно оно испытывает основные нагрузки при работе с вычислительной техникой.

Важно также следить за состоянием техники и соблюдать основные правила обращения с ней.

Системный блок следует включать как можно реже (обычно включается в начале рабочего дня и выключается, выключается работа на нем — в конце дня). Для того, чтобы не выгорал экран и не расходовалась лишняя энергия, в компьютере предусмотрен специальный режим гашения экрана — через определенное время, если никто не работает на нем, т.е. нет обращения к клавиатуре или мыши, он выключается. Если монитор получает питание от системного блока, включая системный блок, включаем и монитор. Если соединение монитора и системного блока параллельно, то

сначала необходимо включить монитор, потом системный блок. Выключать в обратной последовательности.

Экран монитора стеклянный, а потому и хрупкий, и поэтому надо обращаться с ним осторожно. Недопустимо попадание жидкости за заднюю часть экрана может замкнуть проводка, что выведет его из строя и может привести к возникновению пожара. В случае попадания жидкости следует отключить электропитание.

Защита от излучения расположена только на экране, поэтому, находясь прямо перед экраном, пользователь наиболее защищен от вредного воздействия излучения. На заднюю и боковые части монитора в целях экономии защиту не устанавливают. Следовательно, находясь сбоку или сзади монитора, можно получить максимально вредное воздействие.

При работе с клавиатурой стоит придерживаться следующих правил:

- 1) сильно не ударять по клавишам, это приводит к быстрой изнашиваемости прибора.
- 2) не распивать напитки над клавиатурой, так как попадание жидкости в нее приводит к короткому замыканию и выводит из строя клавиатуру, в случае попадания необходимо обесточить компьютер.
- 3) не кушать над клавиатурой бутерброды, семечки, так как крошки, попадающие в клавиатуру, нарушают ее работу.
- 4) при наличии защитной панели следует закрывать клавиатуру, тем самым, защищая ее от пыли.

Требования к параметрам микроклимата помещения. Под метеорологическими условиями (ГОСТ 12.1.005-88) понимают сочетание температуры, относительной влажности, скорости движения и запыленности воздуха. Перечисленные параметры оказывают огромное влияние на функциональную деятельность человека, его самочувствие и здоровье и на надежность средств вычислительной техники. Эти микроклиматические параметры влияют как каждый в отдельности, так и в различных сочетаниях[13].

Температура воздуха является одним из основных параметров, характеризующих тепловое состояние микроклимата. Суммарное тепловыделение в помещении поступает от:

- ЭВМ;
- вспомогательного оборудования;
- приборов освещения;
- людей;
- внешних источников.

Наибольшее количество теплоты выделяют ЭВМ и вспомогательное оборудование. Средняя величина тепловыделения от компьютеров колеблется до 100 Вт/м . Тепловыделения от приборов освещения также велики. Удельная величина их составляет 35 Вт/м . При этом, чем больше уровень освещенности, тем выше удельные величины тепловыделений.

Количество теплоты от обслуживающего персонала незначительно. Оно зависит от числа работающих в помещении, интенсивности работы, выполняемой человеком.

К внешним источникам поступления теплоты относят теплоту, поступающую через окна от солнечной радиации, приток теплоты через непрозрачные ограждения конструкций. Интенсивность этих источников зависит от расположения здания, ориентации по частям света, цветовой гаммы и прочее.

Для обеспечения нормальных условий труда необходимо придерживаться вышеуказанных данных. В целях поддержания температуры и влажности воздуха в помещении можно использовать системы отопления, вентиляции и кондиционирования воздуха.

Параметры микроклимата производственных помещений

Параметры	Значения параметров	
	Оптимальные	Допустимые

Температура	20-22 °С	17-22 °С
Относительная влажность	40-60 %	до 75%
Скорость движения воздуха	ОД м/с	не более 0,3 м/с

На исследуемом предприятии температура воздуха, влажность и скорость движения воздуха держится в рамках оптимальных параметров[13].

ЗАКЛЮЧЕНИЕ

Говоря о дистанционной форме образования, следует говорить о создании единого информационно-образовательного пространства, куда следует включить всевозможные электронные источники информации (включая сетевые): виртуальные библиотеки, базы данных, консультационные службы, электронные учебные пособия и пр.

В ходе создания мобильного приложения учебного назначения были решены такие задачи как:

- удобство и простота интерфейса программы;
- ввод электронных источников информации;
- создание базы школьных учебников (для узбекского языка обучения);
- создание системы контроля знаний;
- возможность работы в оффлайн режиме.

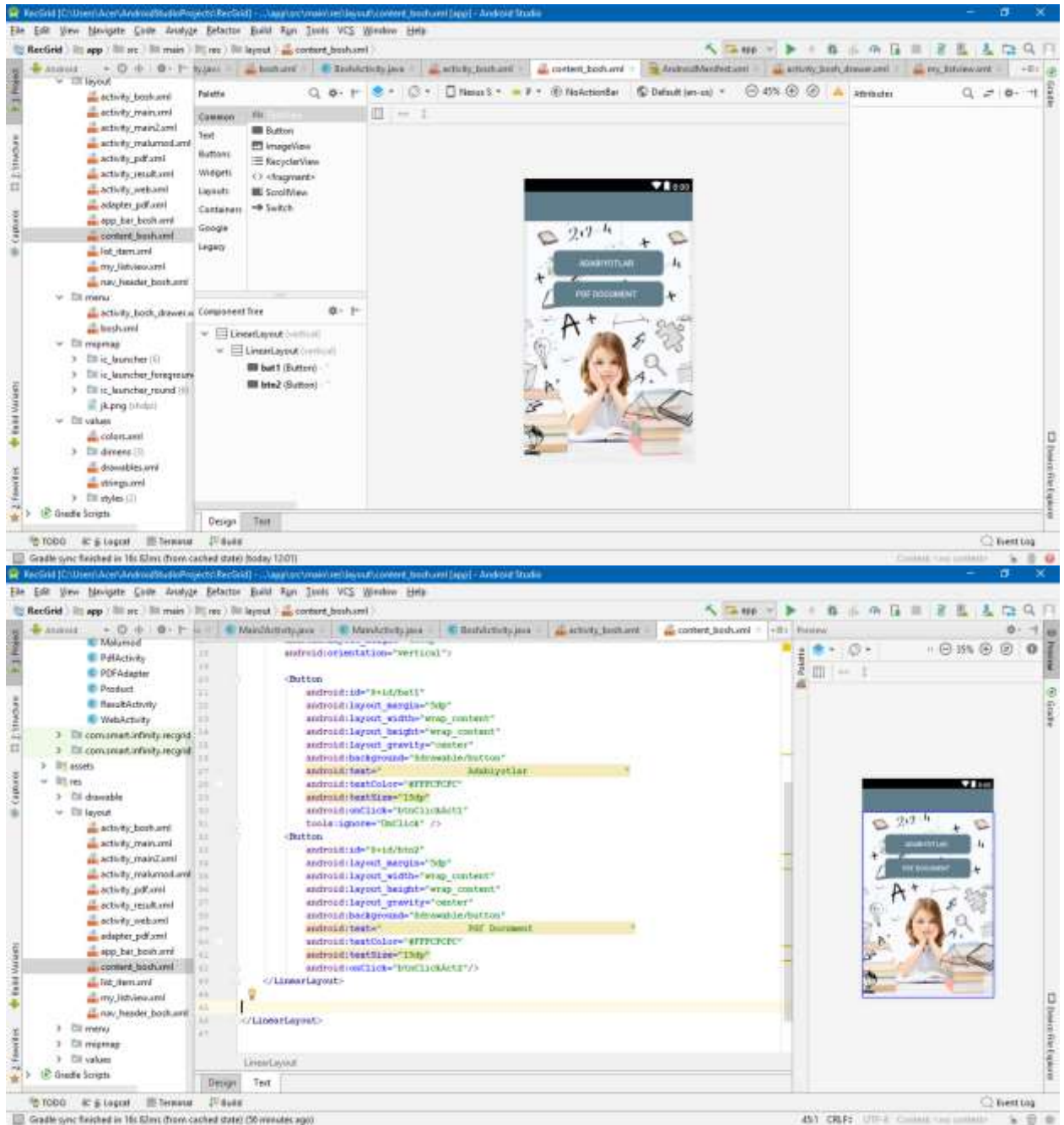
Именно поэтому создание электронных учебников для мобильных устройств является перспективным направлением, которое обеспечивает доступность информации, и тем самым может повысить качество образования.

ЛИТЕРАТУРА

1. Указ президента Республики Узбекистан «О стратегии действий по дальнейшему развитию Республики Узбекистан». Собрание законодательства Республики Узбекистан, 2017 г. №6, ст.70, №20, ст.354.
2. Беспалько В.П. Педагогика и прогрессивные технологии обучения. –М., 1995.
3. Захарова И.Г. Информационные технологии в образовании. Учебное пособие. –М.: Издательский центр «Академия», 2007.
4. Ларнер И.Я. Внимание о технологии обучения. – М.: «Педагогика», 1990.
5. Левитас Г.Г. Компьютер на уроках математики. Информатика и образование. –2002.
6. Андреев А. В., Андреева С. В, Доценко И.Б. Практика электронного обучения с использованием Moodle [Электронный ресурс]. – Таганрог: ТТИ ЮФУ, 2013 г.
7. Полат Е.С. Новые педагогические и информационные технологии в системе образования. // М.: Издательский центр "Академия". - 2003.
8. Гильмутдинов А.Х., Ибрагимов Р.А., Цивильский И.В. Электронное образование. 2012 г.
9. Голощапов А. Л. Google Android: системные компоненты и сетевые коммуникации, БХВ-Петербург 2012 г.
10. Билл Ф., Брайан Х., Программирование под Android. Для профессионалов, Питер 2013 г.
11. Хабибуллин И. Ш., Java, БХВ-Петербург 2012 г.
12. Герберт Шилдт, Java. Полное руководство, Вильямс 2012 г.
13. Кудратов А., Ганиев Т.и др. «Безопасность жизнедеятельности», Ташкент, Алокачи, 2005
14. <http://developer.android.com/>
15. <http://startandroid.ru/ru/>
16. <http://scienceproblems.ru/images/PDF/2016/1/VNO-1-13.pdf>

ПРИЛОЖЕНИЕ

Приложение



<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout_width="match_parent"

android:layout_height="match_parent"

app:layout_behavior="@string/appbar_scrolling_view_behavior"

tools:context="com.smart.infinity.recgrid.BoshActivity"

tools:showIn="@layout/app_bar_bosh"

android:background="@drawable/bosh_sahifa"

android:orientation="vertical">

<LinearLayout

```
    android:layout_margin="50dp"  
    android:layout_width="match_parent"  
    android:layout_height="400dp"  
    android:orientation="vertical">
```

<Button

```
    android:id="@+id/bat1"  
    android:layout_margin="5dp"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:background="@drawable/button"  
    android:text="Adabiyotlar"  
    android:textColor="#FFFCFCFC"  
    android:textSize="15dp"  
    android:onClick="btnClickAct1"  
    tools:ignore="OnClick" />
```

<Button

```
    android:id="@+id/btn2"  
    android:layout_margin="5dp"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:background="@drawable/button"  
    android:text="Pdf Document"  
    android:textColor="#FFFCFCFC"  
    android:textSize="15dp"  
    android:onClick="btnClickAct2"/>
```

</LinearLayout>

</LinearLayout>

JAVA

```
package com.smart.infinity.recgrid;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.support.design.widget.FloatingActionButton;  
import android.support.design.widget.Snackbar;  
import android.view.View;  
import android.support.design.widget.NavigationView;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
import android.support.v7.app.ActionBarDrawerToggle;  
import android.support.v7.app.AppCompatActivity;
```

```

import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.Toast;

public class BoshActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {

    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bosh);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            this, drawer, toolbar, R.string.navigation_drawer_open,
            R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView)
            findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);

        button = (Button) findViewById(R.id.bat1);
    }

    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
        } else {
            super.onBackPressed();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.

```

```

    getMenuInflater().inflate(R.menu.bosh, menu);
    return true;
}

```

@Override

```

public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

@SuppressWarnings("StatementWithEmptyBody")

@Override

```

public boolean onOptionsItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.nav_camera) {
        Intent i = new Intent(this,WebActivity.class);
        startActivity(i);
        // Handle the camera action
    } else if (id == R.id.nav_gallery) {
        Intent i = new Intent(this,MainActivity.class);
        startActivity(i);
    } else if (id == R.id.nav_slideshow) {
        Toast.makeText(this, "Bazzada ma'lumod mavjud mas
",Toast.LENGTH_SHORT).show();
    } else if (id == R.id.nav_manage) {
        Intent i = new Intent(this,Malumod.class);
        startActivity(i);
    } else if (id == R.id.nav_share) {
        Intent share = new Intent( Intent.ACTION_SEND);
        share.setType("text/plain");
        share.putExtra(Intent.EXTRA_SUBJECT, "My new app");
        String appLink = "https://telegram.me/maktab1991";
        share.putExtra(Intent.EXTRA_TEXT, "Try my app: " + appLink);
    }
}

```

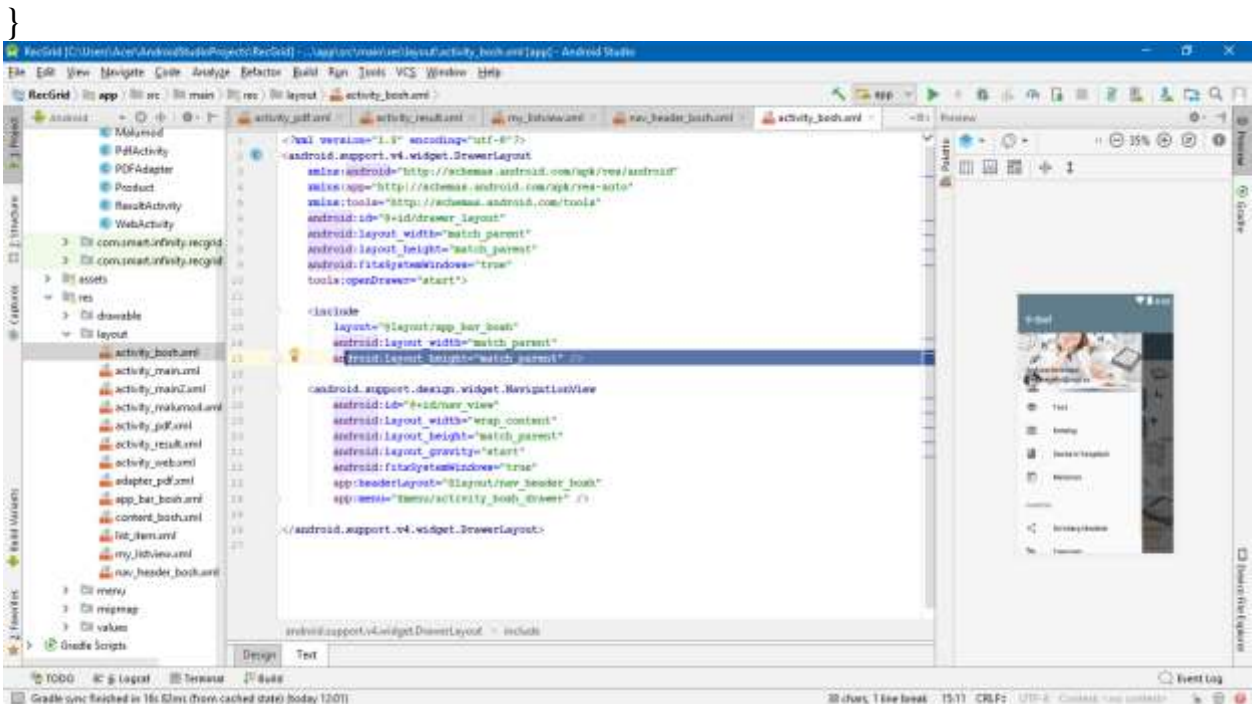
```
startActivity(Intent.createChooser(share, "Share Via"));
```

```
    } else if (id == R.id.nav_send) {
        Intent telegram = new Intent(Intent.ACTION_VIEW ,
Uri.parse("https://telegram.me/maktab1991"));
        startActivity(telegram);
    }
}
```

```
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
return true;
}
```

```
public void btnClickAct1(View v){
    Intent i = new Intent(this,MainActivity.class);
    startActivity(i);
}
```

```
public void btnClickAct2(View view) {
    Intent i = new Intent(this,Main2Activity.class);
    startActivity(i);
}
```



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
```

```
android:id="@+id/drawer_layout"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:fitsSystemWindows="true"  
tools:openDrawer="start">
```

```
<include  
    layout="@layout/app_bar_bosh"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

```
<android.support.design.widget.NavigationView  
    android:id="@+id/nav_view"  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:layout_gravity="start"  
    android:fitsSystemWindows="true"  
    app:headerLayout="@layout/nav_header_bosh"  
    app:menu="@menu/activity_bosh_drawer" />
```

```
</android.support.v4.widget.DrawerLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:app="http://schemas.android.com/apk/res-auto"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:orientation="vertical"  
tools:context="com.smart.infinity.recgrid.MainActivity"  
android:background="@drawable/ss">
```

```
<ViewStub
```

```
    android:id="@+id/stub_list"  
    android:layout_marginTop="10dp"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:inflatedId="@+id/showlayout"  
    android:layout="@layout/my_listview"/>
```

```
</LinearLayout>
```

Java

```

package com.smart.infinity.recgrid;

import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.ViewStub;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.Toast;

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

public class MainActivity extends AppCompatActivity {
    private ViewStub stublist;
    private List<Product> productList;
    private ListView listView;
    private ListViewAdapter listViewAdapter;

    private int CursentViewMode = 0;

    static final int VIEW_MODE_LISTVIEW = 0;

    private List<Product> productlist;
    Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //button = (Button) findViewById(R.id.button);

        stublist = (ViewStub) findViewById(R.id.stub_list);

        stublist.inflate();

        listView = (ListView) findViewById(R.id.mylistview);
        listView.setOnItemClickListener(onItemClickListener);
        getProductList();
        switchView();
    }
}

```

```

    SharedPreferences sharedPreferences =
getSharedPreferences("ViewMode",MODE_PRIVATE);
    CursentViewMode =
sharedPreferences.getInt("CursentViewMode",VIEW_MODE_LISTVIEW);
    }
private void switchView() {

    if(VIEW_MODE_LISTVIEW == CursentViewMode){
        stublist.setVisibility(View.VISIBLE);
    }

    setAdapter();
}
private void setAdapter() {
    if(VIEW_MODE_LISTVIEW == CursentViewMode){
        listViewAdapter = new
ListViewAdapter(this,R.layout.list_item,productsList);
        listView.setAdapter(listViewAdapter);
    }
}
public List<Product> getProductList() {
    productsList = new ArrayList<>();
    productsList.add(new Product(R.drawable.adab , "Adabiyot", "hello world"));
    productsList.add(new Product(R.drawable.alg , "Algebra", "hello world"));
    productsList.add(new Product(R.drawable.uz , "O‘zbekiston tarixi", "hello world"));
    productsList.add(new Product(R.drawable.tili , "Ona tili", "hello world"));
    productsList.add(new Product(R.drawable.jaxon , "Jahon tarixi", "hello world"));
    productsList.add(new Product(R.drawable.biy , "Biologiya", "hello world"));
    productsList.add(new Product(R.drawable.fizka , "Fizika", "hello world"));
    productsList.add(new Product(R.drawable.informat , "Informatika", "hello world"));
    productsList.add(new Product(R.drawable.englis , "Ingliz Tili", "hello world"));
    productsList.add(new Product(R.drawable.kimyo , "Kimyo", "hello world"));
    productsList.add(new Product(R.drawable.iqtisodiy , "Iqtisodiy", "hello world"));
    productsList.add(new Product(R.drawable.geyografiya , "Jahon Iqtisodiy Geografiya", "hello world"));
}

```

```

        productsList.add(new Product(R.drawable.r , "Rus Tili", "hello world"));
        productsList.add(new Product(R.drawable.konus , "Konstitutsiyaviya",
"hello world"));
        return productList;
    }

```

```

    AdapterView.OnItemClickListener onItemClickListener = new
    AdapterView.OnItemClickListener() {

```

@Override

```

        public void onItemClick(AdapterView<?> adapterView, View view, int i,
long l) {

```

```

            Intent intent = new Intent(MainActivity.this,ResultActivity.class);
            intent.putExtra("key",i);
            startActivity(intent);
        }
    }

```

```

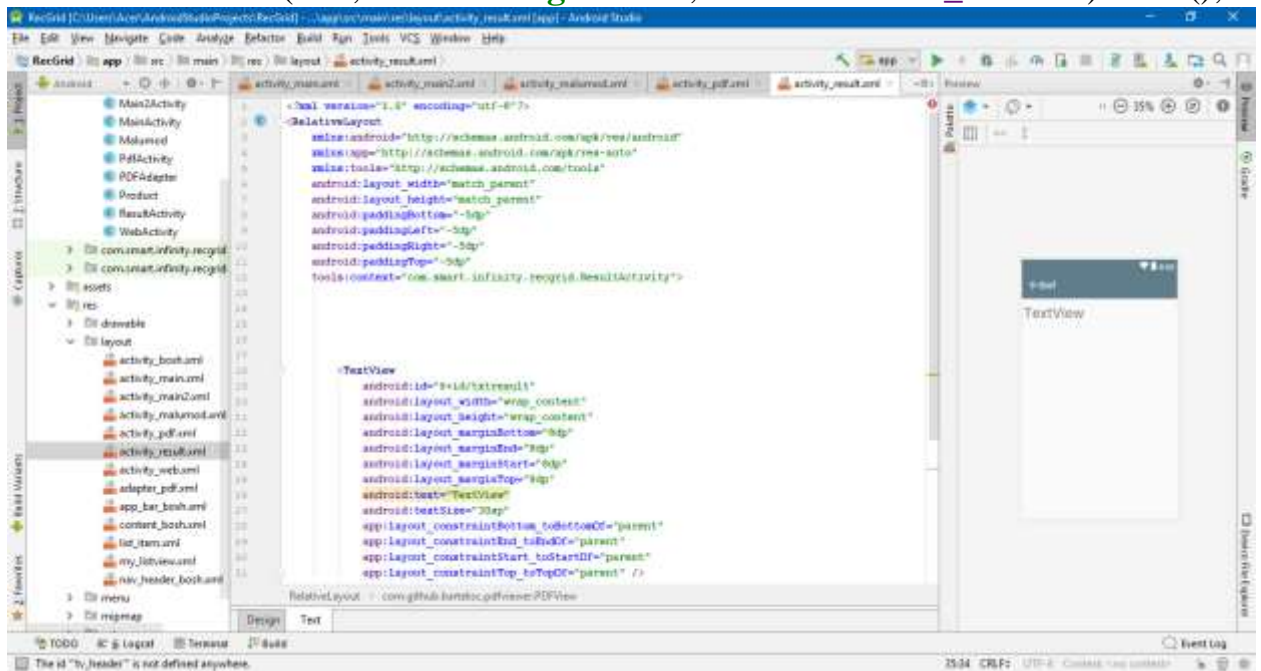
    public void btnClickAct6(View view){

```

```

        Toast.makeText(this, "Sezda Sungi kitob", Toast.LENGTH_SHORT).show();
    }

```



```

<?xml version="1.0" encoding="utf-8" ?>

```

```

<RelativeLayout

```

```

    xmlns:android="http://schemas.android.com/apk/res/android"

```

```

    xmlns:app="http://schemas.android.com/apk/res-auto"

```

```

    xmlns:tools="http://schemas.android.com/tools"

```

```

    android:layout_width="match_parent"

```

```

    android:layout_height="match_parent"

```

```

    android:paddingBottom="-5dp"

```

```

    android:paddingLeft="-5dp"

```

```

    android:paddingRight="-5dp"

```

```
android:paddingTop="-5dp"  
tools:context="com.smart.infinity.recgrid.ResultActivity">
```

```
<TextView  
    android:id="@+id/txtresult"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="8dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="8dp"  
    android:text="TextView"  
    android:textSize="30sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<com.github.barteksc.pdfviewer.PDFView  
    android:id="@+id/pdfView"  
    android:layout_below="@+id/tv_header"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

```
</RelativeLayout>
```

Java

```
package com.smart.infinity.recgrid;
```

```
import android.content.Intent;
```

```
import android.content.res.AssetManager;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.widget.Scroller;
```

```
import android.widget.TextView;
```

```
import com.github.barteksc.pdfviewer.PDFView;
```

```
import com.github.barteksc.pdfviewer.scroll.DefaultScrollHandle;
```

```
public class ResultActivity extends AppCompatActivity {
```

```
    PDFView pdfView;
```

```
    TextView txtresult;
```

```
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_result);

    txtresult = findViewById(R.id.txtresult);

    pdfView = (PDFView) findViewById(R.id.pdfView);

    if (getSupportActionBar() != null)
        getSupportActionBar().hide();

    //txtresult = (TextView) findViewById(R.id.txtresult);
    Intent getIntent = getIntent();
    int result = getIntent.getIntExtra("key", 0);
    String myres;
    myres = String.valueOf(result);
    //    txtresult.setText(result);
    kitob(myres);
}

private void kitob(String i) {
    switch (i) {
        case "0":
            pdfView.fromAsset("Adab.pdf").defaultPage(1)
                .swipeHorizontal(false)
                .enableAnnotationRendering(true)
                .scrollHandle(new DefaultScrollHandle(this))
                .enableSwipe(true)
                .load();
            txtresult.setText("");
            break;
        case "1":
            pdfView.fromAsset("Algebr.pdf")
                .swipeHorizontal(false)
                .enableAnnotationRendering(true)
                .scrollHandle(new DefaultScrollHandle(this))
                .enableSwipe(true)
                .load();
            txtresult.setText("");
            break;
        case "2":
            pdfView.fromAsset("O'zbeT.pdf")
                .swipeHorizontal(false)
                .enableAnnotationRendering(true)

```

```

        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "3":
    pdfView.fromAsset("Ona.pdf")
        .swipeHorizontal(false)
        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "4":
    pdfView.fromAsset("JahonT.pdf")
        .swipeHorizontal(false)
        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "5":
    pdfView.fromAsset("Biyalogiya.pdf")
        .swipeHorizontal(false)
        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "6":
    pdfView.fromAsset("fizika.pdf")
        .swipeHorizontal(false)
        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "7":
    pdfView.fromAsset("Informatika.pdf")
        .swipeHorizontal(false)

```

```

        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "8":
    pdfView.fromAsset("Ingliz.pdf")
        .swipeHorizontal(false)
        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "9":
    pdfView.fromAsset("kimyo.pdf")
        .swipeHorizontal(false)
        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "10":
    pdfView.fromAsset("iqtsod.pdf")
        .swipeHorizontal(false)
        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "11":
    pdfView.fromAsset("geyo.pdf")
        .swipeHorizontal(false)
        .enableAnnotationRendering(true)
        .scrollHandle(new DefaultScrollHandle(this))
        .enableSwipe(true)
        .load();
    txtresult.setText("");
    break;
case "12":
    pdfView.fromAsset("rus.pdf")

```

```

.swipeHorizontal(false)
.enableAnnotationRendering(true)
.scrollHandle(new DefaultScrollHandle(this))
.enableSwipe(true)
.load();

```

```

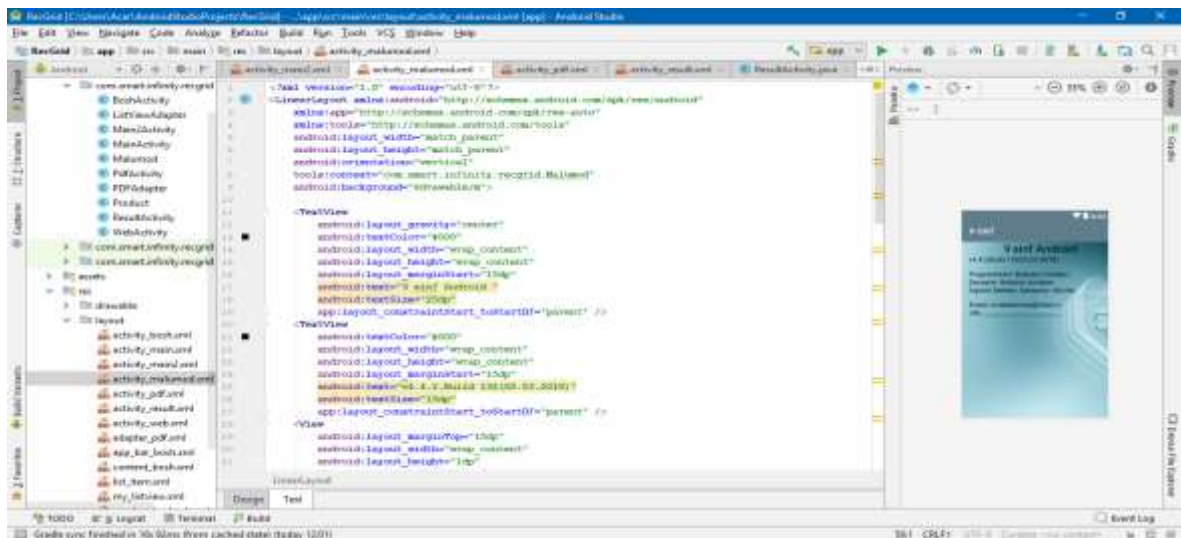
txtresult.setText("");
break;
case "13":
txtresult.setText("basic developer");
break;
default:
txtresult.setText("Bazzada kitob mavjud emas");
break;

```

```

};
}
}

```



```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<LinearLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
android:orientation="vertical"
```

```
tools:context="com.smart.infinity.recgrid.Malumod"
```

```
android:background="@drawable/m">
```

```
<TextView
```

```
android:layout_gravity="center"
```

```
android:textColor="#000"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```

    android:layout_marginStart="15dp"
    android:text="9 sinf Android "
    android:textSize="25dp"
    app:layout_constraintStart_toStartOf="parent" />
<TextView
    android:textColor="#000"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:text="v4.4.2, Build 132(05.03.2018)"
    android:textSize="15dp"
    app:layout_constraintStart_toStartOf="parent" />
<View
    android:layout_marginTop="15dp"
    android:layout_width="wrap_content"
    android:layout_height="1dp"
    />
<TextView
    android:textColor="#000"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:text="Programmist: Bobolov Umidjon"
    android:textSize="15dp"
    app:layout_constraintStart_toStartOf="parent" />

<TextView
    android:textColor="#000"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:text="Dezaynir: Bobolov Umidjon"
    android:textSize="15dp"
    app:layout_constraintStart_toStartOf="parent" />
<TextView
    android:textColor="#000"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:text="Diplom Rahbar: Ganiyeva. Nilufar"
    android:textSize="15dp"
    app:layout_constraintStart_toStartOf="parent" />

<View

```

```

    android:layout_marginTop="15dp"
    android:layout_width="wrap_content"
    android:layout_height="1dp"

```

```

/>

```

```

<TextView

```

```

    android:textColor="#000"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:text="Email: umidsamtiut@mail.ru"
    android:textSize="15dp"
    app:layout_constraintStart_toStartOf="parent" />

```

```

<TextView

```

```

    android:textColor="#000"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:text="URL: _____"
    android:textSize="15dp"
    app:layout_constraintStart_toStartOf="parent" />

```

```

</LinearLayout>

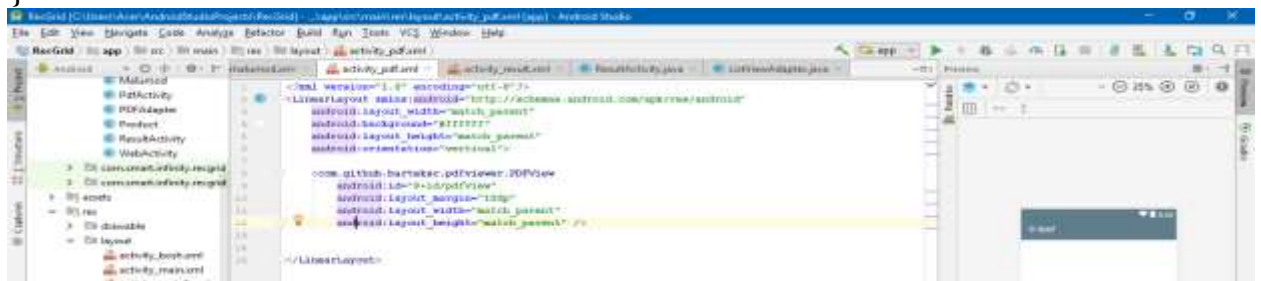
```

Java

```

package com.smart.infinity.recgrid;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class Malumod extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_malumod);
    }
}

```



```

<?xml version="1.0" encoding="utf-8"?>

```

```

<LinearLayout

```

```

    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:background="#ffffff"

```

```

    android:layout_height="match_parent"
    android:orientation="vertical">
    <com.github.barteksc.pdfviewer.PDFView
        android:id="@+id/pdfView"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
package com.smart.infinity.recgrid;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import com.github.barteksc.pdfviewer.PDFView;
import com.github.barteksc.pdfviewer.listener.OnLoadCompleteListener;
import com.github.barteksc.pdfviewer.listener.OnPageChangeListener;
import com.github.barteksc.pdfviewer.scroll.DefaultScrollHandle;
import com.shockwave.pdfium.PdfDocument;
import java.util.List;
public class PdfActivity extends AppCompatActivity implements
OnPageChangeListener,OnLoadCompleteListener {
    public static final String SAMPLE_FILE = "android_tutorial.pdf";
    PDFView pdfView;
    Integer pageNumber = 0;
    String pdfFileName;
    String TAG="PdfActivity";
    int position=-1;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pdf);
        init();
    }
    private void init(){
        pdfView= (PDFView)findViewById(R.id.pdfView);
        position = getIntent().getIntExtra("position",-1);
        displayFromSdcard();
    }

    private void displayFromSdcard() {
        pdfFileName = Main2Activity.fileList.get(position).getName();

        pdfView.fromFile(Main2Activity.fileList.get(position))
            .defaultPage(pageNumber)

```

```

        .enableSwipe(true)

        .swipeHorizontal(false)
        .onPageChange(this)
        .enableAnnotationRendering(true)
        .onLoad(this)
        .scrollHandle(new DefaultScrollHandle(this))
        .load();
    }
    @Override
    public void onPageChanged(int page, int pageCount) {
        pageNumber = page;
        setTitle(String.format("%s %s / %s", pdfFileName, page + 1, pageCount));
    }
    @Override
    public void loadComplete(int nbPages) {
        PdfDocument.Meta meta = pdfView.getDocumentMeta();
        printBookmarksTree(pdfView.getTableOfContents(), "-");
    }

    public void printBookmarksTree(List<PdfDocument.Bookmark> tree, String
sep) {
        for (PdfDocument.Bookmark b : tree) {

            Log.e(TAG, String.format("%s %s, p %d", sep, b.getTitle(),
b.getPageIdx()));

            if (b.hasChildren()) {
                printBookmarksTree(b.getChildren(), sep + "-");
            }
        }
    }
}
package com.smart.infinity.recgrid;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.TextView;
import java.io.File;
import java.util.ArrayList;
public class PDFAdapter extends ArrayAdapter<File> {

```

```

Context context;
ViewHolder viewHolder;
ArrayList<File> al_pdf;

public PDFAdapter(Context context, ArrayList<File> al_pdf) {
    super(context, R.layout.adapter_pdf, al_pdf);
    this.context = context;
    this.al_pdf = al_pdf;
}
@Override
public int getItemViewType(int position) {
    return position;
}

@Override
public int getViewTypeCount() {
    if (al_pdf.size() > 0) {
        return al_pdf.size();
    } else {
        return 1;
    }
}

@Override
public View getView(final int position, View view, ViewGroup parent) {
    if (view == null) {
        view = LayoutInflater.from(getContext()).inflate(R.layout.adapter_pdf,
parent, false);
        viewHolder = new ViewHolder();
        viewHolder.tv_filename = (TextView)view.findViewById(R.id.tv_name);
        view.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) view.getTag();
    }
    viewHolder.tv_filename.setText(al_pdf.get(position).getName());
    return view;
}
public class ViewHolder {
    TextView tv_filename;
}

package com.smart.infinity.recgrid;

/**

```

** Created by Jasur on 07.01.2018.*

**/*

```
public class Product {  
  
    private int imageId;  
    private String title;  
    private String description;  
    public Product(int imageId, String title, String description) {  
        this.imageId = imageId;  
        this.title = title;  
        this.description = description;  
    }  
  
    public int getImageId() {  
        return imageId;  
    }  
  
    public void setImageId(int imageId) {  
        this.imageId = imageId;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
  
    public String getDescription() {  
        return description;  
    }  
  
    public void setDescription(String description) {  
        this.description = description;  
    }  
}  
package com.smart.infinity.recgrid;  
  
import android.content.Context;  
import android.support.annotation.NonNull;  
import android.support.annotation.Nullable;  
import android.view.LayoutInflater;
```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ImageView;
import android.widget.TextView;
import java.util.List;

/**
 * Created by Jasur on 07.01.2018.
 */
public class ListViewAdapter extends ArrayAdapter<Product> {
    public ListViewAdapter(@NonNull Context context, int resource, @NonNull
List<Product> objects) {
        super(context, resource, objects);
    }
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
        View v = convertView;
        if(v == null) {
            LayoutInflater inflater =
(LayoutInflater)getContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
            v = inflater.inflate(R.layout.list_item,null);

            Product product = getItem(position);

            ImageView img = (ImageView) v.findViewById(R.id.imageView);
            TextView txtTitle = (TextView) v.findViewById(R.id.txtTitle);
            //TextView txtDescription = (TextView) v.findViewById(R.id.txtDescription);

            img.setImageResource(product.getImageId());
            txtTitle.setText(product.getTitle());
            // txtDescription.setText(product.getDescription());

            return v;
        }
    }
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".WebActivity">

```

```

    <WebView
        android:id="@+id/webView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

```

```

</RelativeLayout>

```

```

package com.smart.infinity.recgrid;

```

```

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

```

```

public class WebActivity extends AppCompatActivity {

```

```

    private WebView webView;

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_web);

```

```

        webView = (WebView) findViewById(R.id.webView);
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl("https://www.abt.uz/training/start");

```

```

        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);

```

```

        //@uzabt

```

```

        @Override

```

```

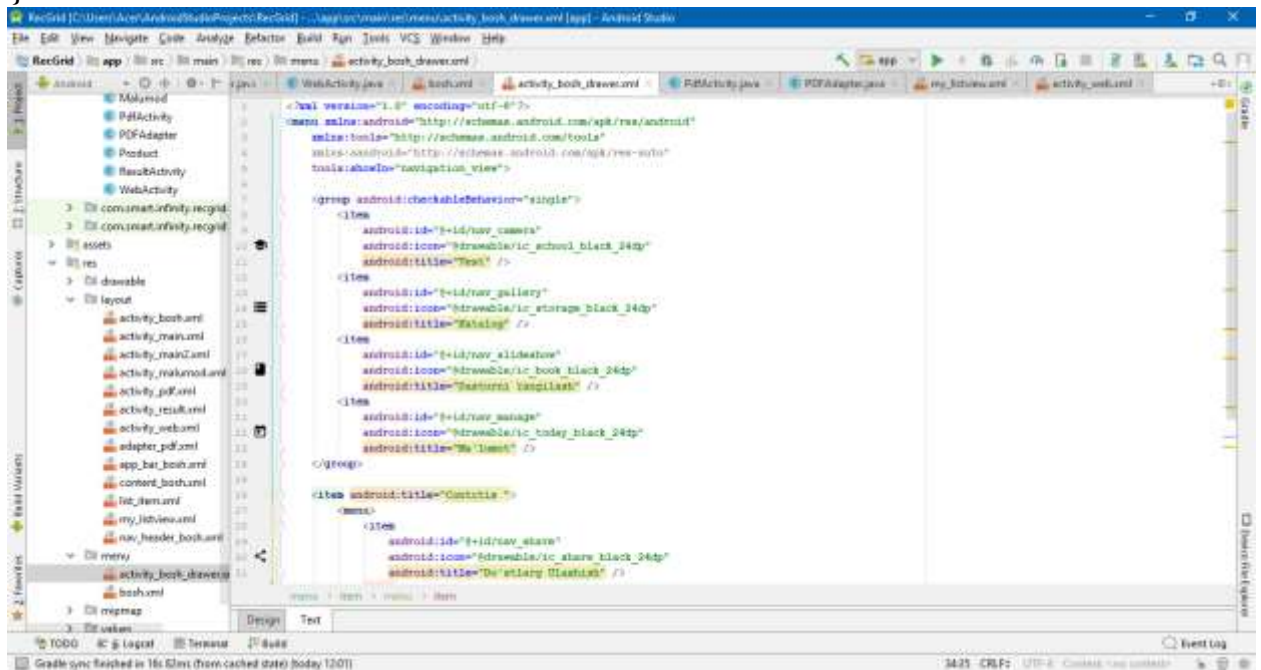
    public void onBackPressed() {
        if (webView.canGoBack()){
            webView.goBack();
        }else {
            super.onBackPressed();
        }
    }

```

`super.onBackPressed();`

}

}



`<?xml version="1.0" encoding="utf-8"?>`

`<menu xmlns:android="http://schemas.android.com/apk/res/android"`
`xmlns:tools="http://schemas.android.com/tools"`
`xmlns:android="http://schemas.android.com/apk/res-auto"`
`tools:showIn="navigation_view">`

`<group android:checkableBehavior="single">`

`<item`

`android:id="@+id/nav_camera"`

`android:icon="@drawable/ic_school_black_24dp"`

`android:title="Test" />`

`<item`

`android:id="@+id/nav_gallery"`

`android:icon="@drawable/ic_storage_black_24dp"`

`android:title="Katalog" />`

`<item`

`android:id="@+id/nav_slideshow"`

`android:icon="@drawable/ic_book_black_24dp"`

`android:title="Dasturni Yangilash" />`

`<item`

`android:id="@+id/nav_manage"`

`android:icon="@drawable/ic_today_black_24dp"`

`android:title="Ma'lumot" />`

`</group>`

`<item android:title="Contctis">`

```

<menu>
  <item
    android:id="@+id/nav_share"
    android:icon="@drawable/ic_share_black_24dp"
    android:title="Do'stlarg Ulashish" />
  <item
    android:id="@+id/nav_send"
    android:icon="@drawable/ic_send_telig_24dp"
    android:title="Telegram" />

</menu >

```

```

</item >

```

```

</menu>

```

AndroidManifest

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.smart.infinity.recgrid">

  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE" />
  <uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".ResultActivity" />
    <activity
      android:name=".MainActivity"
      android:label="@string/app_name"
      android:screenOrientation="portrait" />
    <activity
      android:name=".Malumod"
      android:label="@string/app_name"
      android:screenOrientation="portrait" />
    <activity

```

```

    android:name=".BoshActivity"
    android:label="@string/title_activity_bosh"
    android:screenOrientation="portrait"
    android:theme="@style/AppTheme.NoActionBar">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".Main2Activity"></activity>
<activity android:name=".PdfActivity" />
<activity android:name=".WebActivity"></activity>
</application>

</manifest>

```

Gradle Scripts

apply plugin: **'com.android.application'**

```

android {
    compileSdkVersion 26
    buildToolsVersion "27.0.3"
    defaultConfig {
        applicationId "com.smart.infinity.recgrid"
        minSdkVersion 19
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
}

```

```
implementation 'com.android.support.constraint:constraint-layout:1.0.2'  
testImplementation 'junit:junit:4.12'  
androidTestImplementation 'com.android.support.test:runner:1.0.1'  
androidTestImplementation 'com.android.support.test.espresso:espresso-  
core:3.0.1'  
compile 'com.android.support:design:26.1.0'  
compile 'com.android.support:cardview-v7:26.1.0'  
compile 'com.github.barteksc:android-pdf-viewer:1.4.0'  
implementation 'com.android.support:design:26.1.0'  
androidTestImplementation 'com.android.support.test:runner:0.5'  
androidTestImplementation 'com.android.support.test.espresso:espresso-  
core:2.2.2'  
compile 'com.github.barteksc:android-pdf-viewer:2.0.3'  
}
```