

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

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

**Факультет: Компьютерный инжиниринг**

**Кафедра: Компьютерные системы**

**Направление: 5330500 – “Компьютерный инжиниринг (ИТ-Сервис)”**

“Допущена к защите”  
и.о. заведующей кафедрой  
\_\_\_\_\_ Абдуллаева Н.И.  
“ \_\_\_\_\_ ” \_\_\_\_\_ 2018 г.

**ВЫПУСКНАЯ  
КВАЛИФИКАЦИОННАЯ РАБОТА**  
для получения академической степени бакалавра

**на тему: Разработка мобильного приложения по предмету «Основы  
системного моделирования и проектирования»**

Выпускник: \_\_\_\_\_ Баходиров А.

Научный руководитель: \_\_\_\_\_ Гостев Н.В.

Рецензент: \_\_\_\_\_

Консультант по БЖД: \_\_\_\_\_

**САМАРКАНД – 2018**

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>ГЛАВА 1. ИНФОРМАЦИОННЫЕ ХАРАКТЕРИСТИКИ ПРИМЕНЕНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В ОБРАЗОВАНИИ...6</b>	
1.1. Мобильные приложения в современном образовании.....6	
1.2. Зарубежный опыт мобильного обучения в образовательном процессе.....9	
1.3. Анализ средств разработки мобильных приложений.....13	
<b>ГЛАВА 2. ИНФОРМАЦИОННЫЕ ОСОБЕННОСТИ ТЕХНОЛОГИИ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ОС ANDROID.....18</b>	
2.1. Ключевые особенности Android.....18	
2.2. Архитектура операционной системы Android.....20	
2.3. Инструменты Android-разработчика и этапы разработки мобильного приложения .....25	
<b>ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО ПРЕДМЕТУ “ОСНОВЫ СИСТЕМНОГО МОДЕЛИРОВАНИЯ И ПРОЕКТИРОВАНИЯ ”.....33</b>	
3.1. Структура приложения и компоненты среды разработки.....33	
3.2. Описание функциональных возможностей мобильного приложения и инструкция по применению для пользователей.....38	
3.3. Техника безопасности в работе на персональном компьютере.....42	
<b>ЗАКЛЮЧЕНИЕ .....52</b>	
<b>СПИСОК ЛИТЕРАТУРЫ .....53</b>	
<b>ПРИЛОЖЕНИЕ .....55</b>	

## ВВЕДЕНИЕ

**Актуальность.** В современном мире компьютер и другие цифровые устройства являются составной частью жизни каждого человека. В повседневной жизни мы практически не расстаемся со смартфонами, планшетами и другими мобильными устройствами. Для нынешнего поколения обучающихся в образовательных учреждениях высшего образования, родившихся в цифровом обществе и выросших вместе с информационными технологиями, использование таких технологий, как интернет и мобильные устройства является неотъемлемой и совершенно естественной частью их жизни.

Президент Узбекистана Шавкат Мирзиязев своим указом утвердил стратегию действий по приоритетным направлениям развития Узбекистана в 2017–2021 годах. Основными задачами на 2017-2021 годах в Стратегии действий по пяти приоритетным направлениям развития, а также в рамках реализации Государственной Программы «Год диалога с народом», обозначены: создание благоприятной ИКТ-инфраструктуры, в частности, развитие мобильной связи и цифрового телевидения, строительство оптоволоконных линий связи, поддержка и всестороннее содействие в разработке программного обеспечения, совершенствование системы «Электронное правительство» [1].

Однако цифровые мобильные устройства заняли прочное место не только в нашей повседневной жизни, они повсеместно применяются и в процессе обучения. Современные тенденции в преподавании связаны не только с серьезным пересмотром концепций и методов обучения, но и с техническим и технологическим обновлением процесса обучения. Это выражается как в широком применении уже известных всем интерактивных средств обучения (мультимедийных CD-ROMов, программного обеспечения к интерактивной доске, интерактивных и мультимедийных учебников), так и

в постепенном внедрении в процесс обучения приложений для мобильных телефонов на базе различных платформ: Android, iOS и т.д.

Использование планшетных компьютеров, смартфонов, мобильных телефонов, iPad, iPhone и других подобных устройств в образовательных целях привело к формированию в рамках концепции электронного обучения нового направления – мобильного обучения. Многие ученые и педагоги уверены, что будущее обучения с поддержкой информационно-компьютерных технологий связано именно с распространением мобильных средств связи, появлением большого количества учебных приложений и программ, новых технологий, которые расширяют возможности и качество образования.

**Целью** настоящей работы является разработка мобильного приложения по предмету «Основы системного проектирования и моделирования». Для достижения поставленной цели были сформулированы следующие **задачи**:

- Провести обзор и анализ существующих решений.
- Выбрать методы разработки программного обеспечения (ПО) приложения.
- Выбрать инструментальные и программные средства.
- Разработать структуру ПО приложения.
- Разработать ПО приложения.
- Провести тестирование приложения.

**Объектом исследования** являются мобильные приложения, для самостоятельной работы студентов. **Предмет исследования** состоит в анализе структуры объекта исследования, а также методов и средств его реализации.

Предполагается, что результатом данной работы станет мобильное приложение учебного назначения.

**Практическая значимость исследования** состоит в применении разработанного приложения в учебном процессе.

**Структура и объем работы.** Работа состоит из введения, трех глав, заключения и списка литературы.

В первой главе рассматриваются роль мобильных приложений в современном образовании, зарубежный опыт мобильного обучения в образовательном процессе и анализ средств разработки мобильных приложений.

Во второй главе изложены ключевые особенности Android, архитектура операционной системы Android, а также инструменты Android-разработчика и этапы разработки мобильного приложения.

Третья глава содержит подробную информацию об этапах разработки системы, включая описание инструментов разработки приложения, разработку базы данных, реализацию алгоритмов и интерфейса программы.

В заключении приводятся основные результаты работы.

Приложение содержит тестовые данные и интерфейс системы.

# ГЛАВА 1. ИНФОРМАЦИОННЫЕ ХАРАКТЕРИСТИКИ ПРИМЕНЕНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В ОБРАЗОВАНИИ

## 1.1. Мобильные приложения в современном образовании

С каждым днём все больше увеличивается доля мобильного интернета и количество пользователей мобильных устройств. Такие изменения вполне обусловлены – сейчас все стремятся к удобству и мобильности во всех сферах жизни. С появлением коммуникаторов и планшетов, уже нет необходимости брать с собой «громоздкие» ноутбуки и другие устройства в дорогу, в командировки, на встречи и для других целей. Однако мобильные устройства не приносили бы столько пользы без специальных дополнений – мобильных приложений. Значимость мобильных приложений в образовании растёт, а основная причина этого - возможности, которые они дают:

- совместная работа учащихся над заданиями во время урока и во внеурочной деятельности,
- обмен файлами,
- организация дистанционного обучения и взаимодействия с родителями.

Мобильное обучение (M-Learning) – возможность обучения независимо от места и времени, посредством ряда мобильных устройств. Мобильное обучение – электронное обучение с помощью мобильных устройств, не ограниченное местоположением или изменением местоположения учащегося.

Мобильное обучение можно организовать разными способами, это может быть организация мобильного обучения с помощью систем управления мобильным обучением (платформы для разработки мобильных курсов), например: Litmos, LearningCart, Saba, Blackboard Mobile Learn (мобильная версия Moodle). Другим способом применения мобильных устройств для обучения является использование мобильных приложений [16].

**Мобильное приложение** — это компонент, устанавливаемый на мобильное устройство (телефон, коммуникатор, смартфон и т.п.) под конкретную мобильную платформу (Android, iOS, BlackBerry, HP webOS, Symbian OS, Bada от Samsung и Windows Mobile), подключающийся к мобильному серверу и управляющий пользовательским интерфейсом и бизнес-логикой устройства [17].

На самом деле мобильных ОС много, но наиболее популярными среди них являются три: Android, Apple iOS, Windows Phone:

- 1 место - Система **Android** (78%) — операционная система от Google. Она распространяется бесплатно, а требования к техническим параметрам мобильного устройства – минимальны. Эти характеристики сделали данную платформу одной из самых популярных среди крупных производителей смартфонов.

- 2 место – Система **Apple iOS** (15,2%) - эта ОС полностью закрыта, поэтому нужно платить за продукцию. Данная платформа отличается превосходным качеством работы и стабильностью — это ее несомненное преимущество. К тому же, она легка в освоении. Еще один плюс – безопасность. На Apple Store производится тщательная проверка всех приложений. К недостаткам системы можно отнести то, что она есть только на устройствах Apple.

- 3 место – Система **Windows Phone** (2,5%) - эта ОС разработана в Microsoft. Она не настолько популярна, как вышеназванные, но многие пользователи считают, что в перспективе WP сможет соперничать с лидерами. Интерфейс здесь очень удобный и необычный: вместо виджетов – так называемые «живые плитки» (live tiles). Они отображают информацию на экране без открытия самого приложения (например, календарь, погода и пр.). Часто среди плюсов отмечают отзывчивость системы, скорость и плавность работы интерфейса.

На настоящий момент существует огромное многообразие мобильных приложения как образовательного так и необразовательного типов. К

образовательным мобильным приложениям можно отнести разнообразные узкоспециализированные предметные приложения, назначение которых заранее понятно, например:

- **GeoGebra** - это бесплатная динамическая математическая среда, программная оболочка, где есть весь набор необходимых инструментов для полноценного проектирования графиков и различных фигур. Выбор точек, векторов и фигур (в том числе графов и таблиц) есть во вкладке «Инструменты», причем все объекты можно динамически изменять или создавать новые компоненты. Самые последние обновления GeoGebra дают доступ к работе с 3D объектами.

- **Арт-музей «Timeline»** - незаменимое приложение для тех, кто изучает историю живописи. 80 наиболее известных художников расположены на шкале времени, так что можно не только читать их биографии и рассматривать картины, но и пронаблюдать, как искусство менялось с годами. О каждой картине можно также узнать, когда она была написана, где хранится, к какому периоду творчества художника относится и так далее.

- **Geography Learning Game** - игра-викторина на знание самого широкого спектра географических объектов: рек, гор, озёр, морей, островов, заливов, каналов, пустынь и многого другого. Помимо физической карты мира узнаётся и политическая — изучаются страны, их флаги и столицы. Приложение оформлено в приятных спокойных тонах, а по принципу начисления баллов оно во многом похоже на GeoGuessr: нужно как можно точнее определить местоположение названного объекта. Одно из главных преимуществ приложения — его доступность на русском языке.

- **Chemik крутой инструмент химии** - ещё один инструмент для изучения химии, который позволяет знакомиться с активностью элементов, изучать окислительно-восстановительные реакции, решать задачи по химии, получать конечные продукты реакции и уравнивать коэффициенты. В приложении есть описание реакций более полутора тысяч химических

соединений. Интерфейс приложения предельно прост, впрочем, как и работа в нём: для реакции достаточно выбрать из таблицы необходимые элементы и соединить их. Приложение можно скачать в Google Play бесплатно [18].

- **айМолекула: Биология ДНК** - приложение открывает удивительный мир клеточной биологии. Вы узнаете о структуре и функциях ДНК и РНК молекул, синтезе белка, а также познакомитесь с такими понятиями биологии как: комплементарность, репликация, мутация и транскрипция.

- **Lingualeo** – приложение для изучения английского языка.

В мире, в котором растёт зависимость от средств связи и доступа к информации, мобильные устройства не будут преходящим явлением. Поскольку мощность и возможности мобильных устройств постоянно растут, они могут шире использоваться в качестве образовательных инструментов и занять центральное место как в официальном, так и в неформальном образовании.

## **1.2. Зарубежный опыт мобильного обучения в образовательном процессе**

Аналитическое агентство Nielsen [20] провело исследование, посвященное выявлению особенностей использования мобильных устройств жителями разных стран (в исследовании участвовали жители 10 стран мира) и составило подробную статистику. Все мобильные телефоны исследователи разделили на три группы: смартфоны («продвинутая» операционная система); телефоны мультимедиа (устройства без современной ОС, но с QWERY-клавиатурой или тачскрином); фичафоны («звонилки»). Результаты исследования показали, что практически во всех странах уровень проникновения мобильных телефонов достаточно высок, в среднем от 81% до 99% населения старше 16 лет имеют мобильные устройства. В большинстве стран пользователи «умных» мобильных телефонов – это молодежь [6].

Внедрение в систему современного отечественного образования такой вид электронного обучение как мобильное обучение ставит перед учеными и практиками ряд вопросов требующих своего решения [2].

Во-первых, следует рассмотреть, что же включает в себя содержание понятия «мобильное обучение»; во-вторых, необходимо исследовать какие изменения следует ввести во все компоненты и уровни образовательной системы (теоретико-методологические основания, цели, содержание, средства, методы, технологи, формы и виды учебной деятельности; в-третьих, выявить дидактические функции, свойства мобильных устройств и условия их включенности в процесс обучения.

В научной литературе в данный момент существует несколько определений и трактовок понятия «мобильное обучение», в основе данных определений лежат либо технологические особенности, либо дидактические возможности данных устройств.

В рамках проекта MoLeNet ([www.molenet.org.uk](http://www.molenet.org.uk)) по мобильным обучением понимают процесс использования портативных, беспроводных, удобных и доступных мобильных устройств в образовании, с целью оптимизации и поддержки обучения, что позволит обучающемуся общаться, создавать или получать информацию, а его деятельность станет более эффективной [21].

Итак, использование мобильных устройств в образовании позволяет облегчить, поддержать, усилить, расширить возможности преподавания и обучения, это означает, что мобильное обучение работает лучше всего, когда является дополнительным к основной форме. Как утверждает Дж. Тракслер (10) и мы с ним полностью согласны, мобильные устройства используемые в процессе обучения значительно изменяют последнее в сторону не только модификации форм обучения и расширяя пространственные возможности организации процесса, но и развития новых форм познания и формирования менталитета.

Интеграция мобильного обучения в современную систему образования делает его своевременным (just-in-time), достаточным (just enough) и индивидуализированным (just-for-me). Именно эти характеристики и отличают мобильное обучение от традиционного электронного (e-learning) и смешенного обучения, которые характеризуются мультимедийностью, интерактивностью, структурированностью.

Мобильное обучение есть новая форма обучения, которая отличается от дистанционного, а технологии мобильного обучения трансформируют процесс обучения и делают реальным постулат о активном участие обучающегося в своем собственном образовании (обучающийся – субъект образовательного процесса). Эту позицию разделяют исследователи проекта Mobil (9) и разделяют смешенное обучение от мобильного, выделяя мобильное обучение как индивидуализированное и неформальное.

Ученые М. Бренсфорд, Дж. Дуглас [18] в своих исследованиях так же разделяют два типа обучения мобильное и электронное (e-learning), отмечая, что мобильное обучение отличается доступностью по времени и в пространстве, обеспечивая постоянный доступ к обучающему материалу, что позволяет сформировать у обучающегося понимание образования как непрерывного процесса сопровождающего его в течение всей жизнедеятельности (life-long education).

В определении термина «мобильное обучение» данное С. Геддесом так же отмечается возможность мобильного обучения влиять на поведение и менталитет обучающегося. Д. Келли отмечает, что мобильное обучение – это не процесс переноса учебных материалов на небольшой экран и применение мобильных устройств, а разработка инновационных образовательных мобильных материалов, которые будут осваиваться через введение инновационных форм обучения: учебный микроблог, новостная лента, курскасты и др. Т. Реккедал, А. Дай ученые, участники многоступенчатого проекта, реализуемого в Институте дистанционного обучения в Норвегии подчеркивают, что они внедряли не только новые мобильные устройства в

образовательный процесс, но и выявляли их эффективное воздействие на него. В рамках эксперимента использование средств мобильной связи проходило в несколько этапов. На первом этапе средства мобильной связи и выхода в Интернет использовались для организации процесса дистанционного образования. Обучающиеся на это этапе получали в мобильном формате инструкции записи на курсы и пути его прохождения, свод требований, предъявляемых к слушателям курсов, календарь-планер событий по курсам.

Таким образом, первоначально мобильные устройства можно активно использовать в организации обучения по вводно-адаптационных курсах/модулях. Особенно эффективны будут мобильные технологии в процессе освоения адаптационного модуля в учебном плане обучения инвалидов и лиц с ограниченными возможностями здоровья [17].

На следующем этапе, средства мобильной связи использовались с целью интерактивной связи субъектов образовательного процесса, а некоторые учебные материалы стали доступны для студентов в мобильной форме (мобильные дидактические тесты, задания и т.д.), поскольку располагались на мобильных браузерах. Этот этап, на наш взгляд позволит постепенно включать в мобильное обучение в процесс изучения большинства учебных дисциплин/модулей. Третий этап позволил обучающимся выполнять учебные задания и презентации в мобильном формате, следовательно, появились целые курсы дисциплин/модули, которые стали доступны для обучающихся в мобильном формате. На всех этапах реализации проекта исследователи [8] изучали мнение обучающихся и их педагогов о внедрении мобильного обучения. Обучающиеся отметили среди существенных преимуществ данного вида обучения доступность учебных материалов, облегченность коммуникаций и консалтинга, наличие высококачественного мобильного образовательного контента. Преподаватели же отметили, что данный формат обучения изменяет отношение студентов к собственному образованию, позволяет им развивать аналитическое

мышление, навыки самоорганизации, саморазвития, профессионального становления, умения критического анализа полученных знаний и эффективного применения их на практике.

Таким образом, мобильное обучение обладает высоким дидактическим потенциалом, а его технологии интегрируясь в образование создают новую модель профессиональной подготовки. Реализация на практике данной модели возможно при эффективном использовании интерактивных, инновационных методов обучения, методов, основанных на автономии обучающегося. Следует отметить, что наиболее перспективный путь внедрение мобильных устройств в обучение заключается в грамотном сочетании новых форм обучения (интерактивные лекции, вебинары, симуляции, тренинги, дискуссии), новых видов учебных заданий (слайд-презентации, веб-проекты, учебные подкасты) и традиционных. Таким образом мобильное обучение в системе профессионального образования должно строиться на принципе интерактивного управляемого самообучения, что позволит снизить деструктивное влияние информационно-коммуникационных технологий на социальную и когнитивную деятельность обучающегося [14].

### **1.3. Анализ средств разработки мобильных приложений**

В данный момент для создания мобильных приложений существует два вида средств разработки: средства разработки для создания нативных мобильных приложений и средства создания web-приложений адаптированных под мобильные приложения. Для анализа нами были выбраны средства разработки нативных приложений так как в данной среде они более популярны и востребованы. Мы рассмотрели три наиболее популярных средства разработки: Android studio, Eclipse, NetBeans IDE. Для анализа данных программных продуктов были выбраны следующие критерии:

- функциональность;
- удобство интерфейса;
- возможность подключения дополнительных модулей;
- требовательность к системе;
- возможность подключения к системам контроля версий;
- наличие встроенных компонентов тестирования приложения [2].

Android Studio – продукт компании Google. Основана на программном обеспечении IntelliJ IDEA от компании JetBrains, официальное средство разработки Android приложений. Актуальная на данный момент версия приложения 2.2. Данная среда разработки доступна для Windows, OS X и Linux. Функционал данного приложения использует язык Java для написания программного кода. Разработка интерфейса производится drag–n–drop методом, но так же имеется возможность использовать XML. Для удобства разработки интерфейса так же существуют шаблоны, ориентируемые на задачу, которую должно выполнять приложение. Интерфейс данного ПО перегружен. Интерфейс библиотек приложения имеет вид выпадающего древа и под него приходится отводить очень много места в общем интерфейсе, в противном случае, информация становится нечитаемой. Та же самая ситуация и с окном отладки. В функционале Android Studio возможность подключения дополнительных плагинов отсутствует. Данное средство разработки очень требовательно к технической составляющей ЭВМ, по сравнению с другими средствами разработки. Минимальное количество ОЗУ требуемое для данного продукта 2 гигабайта [2].

Eclipse – среда разработки созданная компанией Eclipsefoundation. Актуальная на данный момент версия Eclipse 4.6(Neon). Используемый язык для написания мобильных приложений – Java. Включает в свой функционал не только средства для разработки мобильных приложений, но и web – приложений, поддерживает языки C++и PHP. Для разработки интерфейса нет шаблонов, или готовых объектов. Имеется возможность синхронизации

разных ПК для разработки одного проекта посредством облачных сервисов. Простой и удобный интерфейс. Панель библиотек имеет древовидную структуру, но проблема предыдущего продукта здесь решена просто – существует возможность полностью сворачивать неиспользуемые в данный момент окна. Существует встроенный помощник для написания простого Hello world приложения. Есть возможность подключения дополнительных плагинов для расширения функционала. Для этого в продукте предусмотрен целый модуль Eclipse Marketplace, предоставляющий на выбор сразу три «Рынка» плагинов: Eclipse Marketplace, Obeo Marketplace и RedHat. Так же существует возможность написания собственных плагинов и их использования без получения лицензии или обязательного предоставления разработки на рынке. Системные требования для данного ПО не описаны разработчиком, но при использовании на ПК средней производительности 2012 года затруднений замечено не было. Ввиду наличия синхронизации с облаком в системе полностью отсутствуют какие-либо интегрированные средства для подключения к системам контроля версий. Эмуляция устройства отсутствует полностью [7].

NetBeans IDE– продукт компании NetBeans Community. Актуальная на данный момент версия приложения 8.2. В данном программном продукте функционал полностью реализуется посредством плагинов. Вследствие этого, ПО поддерживает большое количество используемых языков. Для разработки нативных мобильных приложений используется Java, но существует возможность создания web–приложения написанного на HTML5 или JS+PHP. Встроенный отладчик полностью отсутствует, но имеется возможность подключения удаленного отладчика через сеть «Интернет». Интерфейс очень похож на интерфейс Eclipse, за исключением отсутствия окна отладки, и наличия разметки номеров строк. Требования к ЭВМ либеральные. Для минимальной работы продукта требуется 512 мегабайт ОЗУ, для более комфортной работы рекомендуется использовать компьютер с 2 гигабайтами. Имеет встроенную возможность подключения к системам

контроля версий. Взаимодействует с GitHub, Mercurial и Subversion. Встроенные компоненты для тестирования приложения полностью отсутствуют. Таким образом, анализ средств разработки мобильных приложений можно свести в таблицу, оценивая рассмотренные критерии по пятибалльной шкале (табл. 1.).

Таблица 1. Сравнительный анализ средств разработки мобильных приложений по 5- балльной шкале

Критерии оценки	Средства разработки мобильных приложений		
	AndroidStudio	Eclipse	NetBeansIDE
Функциональность	5	4	3
Удобство интерфейса	2	5	5
Возможность подключения дополнительных модулей	1	5	4
Требовательность к системе	1	5	4
Возможность подключения к системам контроля версий	1	1	5
Наличие встроенных компонентов тестирования приложения	3	1	1

Следуя заключениям, описанным выше, каждый из данных программных продуктов подходит для разных направленностей и предпочтений в разработке мобильных приложений. Если необходим мощный инструмент, который не требует временных затрат для поиска или

разработки дополнительных модулей, то стоит выбрать Android studio. Если необходима более гибкая и мобильная разработка, то рекомендуется выбрать NetBeans IDE. Если разработчика интересует нечто среднее между двумя первыми, отличным выбором будет разработка в Eclipse [10].

## **ГЛАВА 2. ИНФОРМАЦИОННЫЕ ОСОБЕННОСТИ ТЕХНОЛОГИИ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ОС ANDROID**

### **2.1. Ключевые особенности Android**

Android является распространенной операционной системой (ОС) для мобильных устройств - телефонов и планшетов. Данная система имеет множество отличительных черт, которые делают ее узнаваемой и привлекательной для большого количества пользователей по всему миру.

Операционная система Android является нетребовательной и способна работать на разных конфигурациях. Именно поэтому большинство мировых производителей оснащают свои устройства данной ОС, поскольку другие программные продукты предназначены для отдельных аппаратов, соответствующих определенной спецификации. Такая гибкость Android связана с тем, что система построена на ядре Linux, имеющей открытый программный код, что дает неограниченные возможности разработчикам. Android может быть запущен на устройствах, имеющих объем оперативной памяти менее 256 Мб. Наиболее новые версии системы требуют 512 Мб оперативки, что также является небольшим значением для современных аппаратов [7].

Система не требует наличия высокопроизводительного процессора и может работать на устройствах, оснащенных ядром с частотой 600 МГц.

Операционная система дает возможность установки приложений с официального репозитория Google, который предоставляет самую большую в мире базу программ. Это связано с тем, что каждый разработчик может самостоятельно написать любую программу для аппарата и разместить ее в магазине. Возможность также реализована благодаря открытости операционной системы. Стоит отметить, что приложения на устройства под управлением Android могут быть установлены как непосредственно с телефона или планшета, так и через компьютер путем загрузки файла .apk и его последующей установки на аппарате.

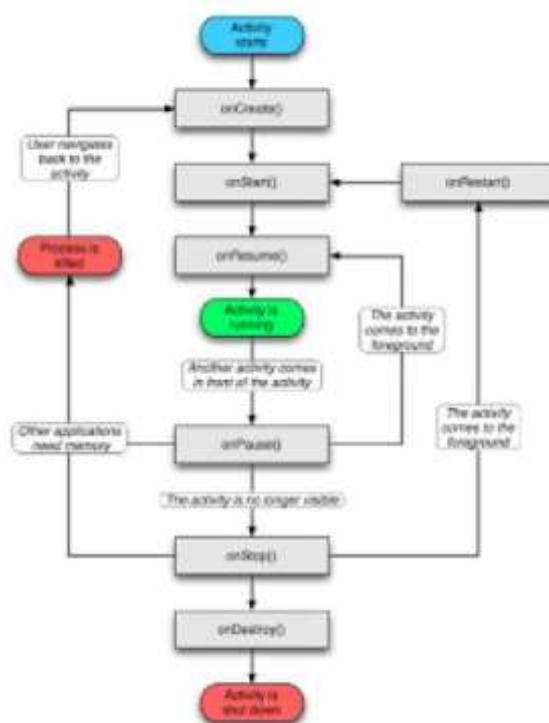
Отличительной особенностью Android является его интегрированность с сервисами Google - Gmail, Hangouts, Voice Search и т.п. На Android официально реализована поддержка Chrome, что позволяет синхронизировать открываемые в браузере вкладки на смартфоне с компьютерным браузером.

Например, вы можете начать просмотр страниц с вашего телефона и при желании продолжить изучать информацию, открыв эту же вкладку на компьютере, не прибегая к помощи повторного поиска.

«Андроид» имеет достаточно простой и интуитивно понятный интерфейс. Все нужные приложения размещаются одновременно на главном экране и в меню аппарата, которое вызывается нажатием на центральную сенсорную клавишу или соответствующую кнопку на экране. Все настройки располагаются в секции «Настройки», а каждое действие пользователя поясняется комментариями и подсказками при первом запуске аппарата. Операционная система быстро реагирует на нажатия пользователя и производит установку и скачивание нужных программ и файлов со скоростью, которая не проигрывает другим современным мобильным ОС. [8]

Приложения для Android в своей работе используют окна(аналогично Windows), однако в данной системе вышеуказанные окна носят иное название - Activity. Как и в Windows, каждое окно имеет свой жизненный цикл и свои особенности. При создании нового окна вызывается метод onCreate(), при разработке данный метод переопределяется и в нем происходит инициализация приложения и его компонентов. Далее вызываются методы onStart() и onResume(). Оба метода вызываются перед отображением окна при его создании, либо восстановлении(при переключении из другого приложения, при разворачивании свернутого приложения и тп). При сворачивании вызываются методы onPause() и onStop(). При закрытии приложения и окна вызывается onDestroy(), в данном методе можно сохранить пользовательские данные и параметры. Полное описание и последовательность вызова методов можно найти на

официальном сайте. Общая схема жизненного цикла приложения для Android представлена на рисунке 2.

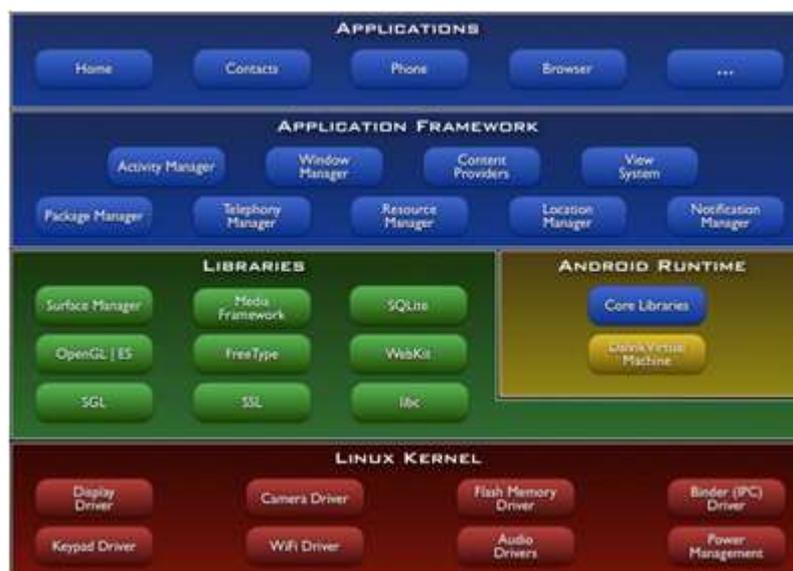


*Рисунок 2.1. - Жизненный цикл приложения для системы под управлением Android*

## **2.2. Архитектура операционной системы Android**

Если представить компонентную модель Android в виде некоторой иерархии, то в самом низу, как самая фундаментальная и базовая составляющая, будет располагаться ядро операционной системы.

○ Часто компонентную модель ещё называют программным стеком. Действительно, это определение тут уместно, потому что речь идет о наборе программных продуктов, которые работают вместе для получения итогового результата. Действия в этой модели выполняются последовательно, и уровни иерархии также последовательно взаимодействуют между собой.



Как известно, Андроид основан на несколько урезанном ядре ОС Linux и поэтому на этом уровне мы можем видеть именно его (версии 2.6.x). Оно обеспечивает функционирование системы и отвечает за безопасность, управление памятью, энергосистемой и процессами, а также предоставляет сетевой стек и модель драйверов. Ядро также действует как уровень абстракции между аппаратным обеспечением и программным стеком.

«Выше» ядра, как программное обеспечение промежуточного слоя, лежит набор библиотек (Libraries), предназначенный для обеспечения важнейшего базового функционала для приложений. То есть именно этот уровень отвечает за предоставление реализованных алгоритмов для вышележащих уровней, поддержку файловых форматов, осуществление кодирования и декодирования информации (в пример можно привести мультимедийные кодеки), отрисовку графики и многое другое. Библиотеки реализованы на C/C++ и скомпилированы под конкретное аппаратное обеспечение устройства, вместе с которым они и поставляются производителем в предустановленном виде.

Перечислим некоторые из них:

Surface Manager – в ОС Android используется композитный менеджер окон, наподобие Compriz (Linux), но более упрощенный. Вместо того чтобы производить отрисовку графики напрямую в буфер дисплея, система

посылает поступающие команды отрисовки в закадровый буфер, где они накапливаются вместе с другими, составляя некую композицию, а потом выводятся пользователю на экран. Это позволяет системе создавать интересные бесшовные эффекты, прозрачность окон и плавные переходы.

Media Framework – библиотеки, реализованные на базе PacketVideo OpenCORE. С их помощью система может осуществлять запись и воспроизведение аудио и видео контента, а также вывод статических изображений. Поддерживаются многие популярные форматы, включая MPEG4, H.264, MP3, AAC, AMR, JPG и PNG.

SQLite – легковесная и производительная реляционная СУБД, используемая в Android в качестве основного движка для работы с базами данных, используемыми приложениями для хранения информации.

3D библиотеки — используются для высокооптимизированной отрисовки 3D-графики, при возможности используют аппаратное ускорение. Их реализации строятся на основе API OpenGL ES 1.0.

- OpenGL ES (OpenGL for Embedded Systems) – подмножество графического программного интерфейса OpenGL, адаптированное для работы на встраиваемых системах.

FreeType – библиотека для работы с битовыми картами, а также для растеризации шрифтов и осуществления операций над ними. Это высококачественный движок для шрифтов и отображения текста.

LibWebCore – библиотеки известного шустрого браузерного движка WebKit, используемого также в десктопных браузерах Google Chrome и Apple Safari [9].

SGL (Skia Graphics Engine) – открытый движок для работы с 2D-графикой. Графическая библиотека является продуктом Google и часто используется в других их программах. SSL - библиотеки для поддержки одноименного криптографического протокола.

Libc – стандартная библиотека языка C, а именно её BSD реализация, настроенная для работы на устройствах на базе Linux. Носит название Bionic.

На этом же уровне располагается Android Runtime – среда выполнения. Ключевыми её составляющими являются набор библиотек ядра и виртуальная машина Dalvik. Библиотеки обеспечивают большую часть низкоуровневой функциональности, доступной библиотекам ядра языка Java.

Каждое приложение в ОС Android запускается в собственном экземпляре виртуальной машины Dalvik. Таким образом, все работающие процессы изолированы от операционной системы и друг от друга. И вообще, архитектура Android Runtime такова, что работа программ осуществляется строго в рамках окружения виртуальной машины. Благодаря этому осуществляется защита ядра операционной системы от возможного вреда со стороны других её составляющих. Поэтому код с ошибками или вредоносное ПО не смогут испортить Android и устройство на его базе, когда сработают. Такая защитная функция, наряду с выполнением программного кода, является одной из ключевых для надстройки Android Runtime. Уровнем выше располагается Application Framework, иногда называемый уровнем каркаса приложений. Именно через каркасы приложений разработчики получают доступ к API, предоставляемым компонентами системы, лежащими ниже уровнем. Кроме того, благодаря архитектуре фреймворка, любому приложению предоставляются уже реализованные возможности других приложений, к которым разрешено получать доступ [11].

В базовый набор сервисов и систем, лежащих в основе каждого приложения и являющихся частями фреймворка, входят:

Богатый и расширяемый набор представлений (Views), который может быть использован для создания визуальных компонентов приложений, например, списков, текстовых полей, таблиц, кнопок или даже встроенного web-браузера.

Контент-провайдеры (Content Providers), управляющие данными, которые одни приложения открывают для других, чтобы те могли их использовать для своей работы.

Менеджер ресурсов (Resource Manager), обеспечивающий доступ к ресурсам без функциональности (не несущими кода), например, к строковым данным, графике, файлам и другим.

Менеджер оповещений (Notification Manager), благодаря которому все приложения могут отображать собственные уведомления для пользователя в строке состояния.

Менеджер действий (Activity Manager), который управляет жизненными циклами приложений, сохраняет данные об истории работы с действиями, а также предоставляет систему навигации по ним.

Менеджер местоположения (Location Manager), позволяющие приложениям периодически получать обновленные данные о текущем географическом положении устройства.

Таким образом, благодаря Application Framework, приложения в ОС Android могут получать в своё распоряжение вспомогательный функционал, благодаря чему реализуется принцип многократного использования компонентов приложений и операционной системы. Естественно, в рамках политики безопасности [12].

- Стоит отметить, просто на понятийном уровне, что фреймворк лишь выполняет код, написанный для него, в отличие от библиотек, которые исполняются сами. Ещё одно отличие заключается в том, что фреймворк содержит в себе большое количество библиотек с разной функциональностью и назначением, в то время как библиотеки объединяют в себе наборы функций, близких по логике.

На вершине программного стека Android лежит уровень приложений (Applications). Сюда относится набор базовых приложений, который предустановлен на ОС Android. Например, в него входят браузер, почтовый клиент, программа для отправки SMS, карты, календарь, менеджер контактов и многие другие. Список интегрированных приложений может меняться в зависимости от модели устройства и версии Android. И помимо этого базового набора к уровню приложений относятся в принципе все приложения

под платформу Android, в том числе и установленные пользователем. Считается, что приложения под Android пишутся на языке Java, но нужно отметить, что существует возможность разрабатывать программы и на C/C++ (с помощью Native Development Kit), и на Basic (с помощью Simple) и с использованием других языков. Также можно создавать собственные программы с помощью конструкторов приложений, таких как App Inventor. Словом, возможностей тут много. Ну и ещё раз про архитектуру ОС Android можно узнать из первых рук, посмотрев это видео, в котором разработчики представляют презентацию, посвященную разбору системы.

### **2.3. Инструменты Android-разработчика и этапы разработки мобильного приложения**

Непосредственно переход к практике программирования под любую платформу начинается с рассмотрения имеющегося арсенала инструментов разработки. Уже после того, как инструменты выбраны (руководствуясь какими-то соображениями и предпочтениями), настроены для работы и опробованы в написании каких-нибудь простых инструкций, начинается «большое» программирование и совершенствование навыков.

Прежде чем рассматривать инструменты для Android, необходимо ввести некоторые общие понятия, с которыми предстоит столкнуться на данном этапе. Первое базовое понятие – это SDK или Software Development Kit («devkit»). Под ним понимают набор средств разработки, позволяющий программистам создавать приложения под некоторую архитектуру — для определенного пакета программ, аппаратной или программной платформы, операционной или компьютерной систем. SDK, как правило, предоставляет разработчику широкий набор инструментов, который может включать в себя средства отладки и справочные материалы (примеры кода, заметки и т.д.). SDK часто предоставляется непосредственно разработчиком целевой платформы и в нем учитываются её особенности. Распространяется (как и в случае с Android) такой инструментарий может бесплатно. Это помогает

платформе обзаводиться разнообразными приложениями за счет сторонних разработчиков.

Ещё одним понятием, которое часто можно встретить, когда речь идет об инструментах разработки, является IDE (Integrated development environment) – интегрированная среда разработки. Она представляет собой набор программных средств, призванный максимизировать производительность программиста за счет построения для него удобной и дружелюбной среды для процесса программирования. Этот сложный программный комплекс может включать в себя текстовый редактор (чаще всего с подсветкой синтаксиса и подсказками), компилятор или интерпретатор (иногда поддерживаются несколько языков), интегрированный отладчик и средства для автоматизации сборки. Часто поддерживаются возможности интеграции со сторонним программным обеспечением, например, средствами проектирования или контроля версий. Также чаще всего в IDE есть инструменты для быстрого построения и визуального редактирования графического интерфейса будущей программы, а также инструменты вроде диаграмм иерархии классов, браузера классов, инспектора объектов или менеджера ресурсов. Таким образом, интегрированная среда разработки призвана объединить различные инструменты в один программный комплекс для обеспечения удобства и повышения скорости разработки ПО. Хотя на деле можно и не использовать для программирования какую-либо IDE, предпочтя несвязные между собой инструменты [13].

Разработка приложений для платформы Android связана с группой инструментов, которые предоставляются набором Android SDK. Также понадобятся инструментарий для разработки приложений в Java SE (JDK) и интегрированная среда разработки. В качестве последней принято использовать Eclipse IDE, что будем делать и мы, хотя стоит отметить, что разработку приложений можно вести и с помощью простейшего текстового редактора или других IDE, а также вызывать инструменты с помощью

скриптов или использования командной строки. Однако разработка в Eclipse является предпочтительным методом, потому что, во-первых, эта среда может напрямую обращаться к необходимым инструментам, а во-вторых, для неё существует специальный плагин, Android Development Toolkit (ADT), который обеспечивает переход к созданию приложений под Android с помощью Eclipse IDE ещё более удобным образом.

В заключении отметим, что Android SDK требует JDK версии 5 или выше, а также Eclipse версии 3.3 или выше. На сайтах разработчиков доступны версии SDK, Java и Eclipse для операционных систем Windows, Linux и Mac OS, а в SDK входит эмулятор для каждой из них. Вообще говоря, приложения под Android выполняются в виртуальной машине, и выбор любой из существующих операционных систем не дает никаких преимуществ перед другими, так что можно работать с той, которая более удобна. Наши дальнейшие примеры будут рассмотрены для ОС Windows, но, если будет необходимость, мы рассмотрим и работу с Linux [4].

Рассмотрим ключевые этапы создания мобильного приложения.

Первый этап создания мобильного приложения - это составление технического задания. Техническое задание (также — техзадание, ТЗ) — технический документ (спецификация), оговаривающий набор требований к системе и утверждённый как заказчиком/пользователем, так и исполнителем/производителем системы. Такая спецификация может содержать также системные требования и требования к тестированию.

Необходимо так же составить план задач. Он должен быть подробно описан, чтобы не переделывать его вновь. Ведь упустив даже незначительную деталь, может порушиться все строение мобильного приложения. Функционал приложения – основа всей работы.

Второй этап разработки мобильного приложения - это проектирование UI/UX. User Experience Design в переводе означает «опыт взаимодействия» и включает в себя различные UX-компоненты: информационную архитектуру, проектирование взаимодействия, графический дизайн и контент. На данном

этапе реализуются все работы, которые были описаны в техническом задании. Создается графическая карта взаимодействия между экранами. Также на этом этапе работы надо определить, каким образом будет работать приложение, и как будет проходить работа пользователя с ним.

Третий этап: работа с дизайном. Дизайн создается на основе пожеланий клиента и нацелен на целевую аудиторию. Изначально прорабатывается дизайн 1-3 страниц, который закладывает всю основу приложения. По пожеланию клиента может быть создано несколько стилей дизайна, оценив которые, заказчик определит лучший вариант.

Четвертый этап: разработка. Специалисты из статичного состояния переводят приложение в интерактивную модель. Этого можно добиться с помощью верстки. Законченный вариант работы отправляется клиенту, чтобы он оценил итог работы и высказал свое мнение [13].

Несколько дней мобильное приложение активно тестируется. Если выявляются какие-либо ошибки, то приложение дорабатывают.

Доработка занимает примерно половину всего времени, которое было потрачено ранее. Следует сказать, что редактирование приложения неизбежно. В процессе его создания нельзя предсказать все возможные недочеты, которые могут возникать в момент использования приложения.

Тестирование. На различных этапах разработки приложения обязательным является внутреннее тестирование приложения как на симуляторах, так и на реальных устройствах. Цель тестирования — убедиться, что взаимодействие приложения с аппаратной и программной платформой смартфонов и планшетов будет именно таким, как предполагалось на этапе прототипирования.

Создание предрелизной версии. В результате серии тестов и доработок приложения должна быть получена рабочая версия приложения. Именно эту версию и предстоит добавить в магазин приложений: Apple App Store, Google Play, магазин приложений Windows Phone (в зависимости от того, для какой платформы ведется разработка) или любой аналогичный сервис для

дистрибуции приложений.

Добавление приложения в магазин. Финальный этап работы студии — добавление приложения на ревью в один из указанных выше магазинов приложений (в случае Componentix речь идет об App Store или о Google Play).

Требования к системе. GooglePlay(предыдущее название—Android Market)—магазин приложений от компании Google, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения.

Пользовательский интерфейс является связующим звеном между пользователем и компьютером. Он обеспечивает взаимодействие человека и машины, обмен действиями и реакцией на них. Юзабилити — степень удобства пользование программой, является одним из важнейших этапов разработки программы. Хороший пользовательский интерфейс учитывает особенности людей, минимизирует ошибки и раздражение пользователя. Обычный пользователь, впервые, открывая приложение видит именно его интерфейс, и от того, насколько этот интерфейс одновременно прост и интуитивно понятен зависит то, будет ли приложение пользоваться популярностью. Критерии к пользовательскому интерфейсу:

1. Интерфейс должен быть понятен и легок к освоению наибольшему кругу лиц (т.е. иметь наименьший порог вхождения);
2. Предоставлять пользователю возможность ввода информации, не показывая процесс вычислений;
3. Отвечать всем потребностям пользователя, не заостряя его внимания на процессе обработки данных [18].

Следуя данным критериям, делается вывод что интерфейс должен быть простым и понятным с первого взгляда. У пользователя не должно возникать никаких вопросов по взаимодействию с приложением и навигации между экранами для достижения поставленных задач. Также следует минимизировать число шагов, за которые достигается решение данных задач.

Пользовательский интерфейс должен удовлетворять следующим требованиям:

1. Цветовая гамма приложения не должна быть яркой и содержать «кислотные цвета», но при этом сочетание цветов должно быть контрастным;

2. Дизайн должен быть простым и понятным, необходимо избегать использование непонятных иконок и пространственных описаний;

3. Навигация между окнами также должна быть предельно простой и понятной. У пользователя нет никакого желания блуждать в «лабиринте» из разных окон;

4. Хорошим тоном также является наличие обратной связи с пользователем. Приложение должно уведомлять пользователя о том, что оно откликнулось на какой-либо запрос.

Также следует учитывать то, что приложение будет использоваться на разных экранах с разным разрешением, пропорциями соотношения сторон дисплея. Благодаря данным особенностям, на этапе проектирования необходимо позаботиться о том, чтобы интерфейс легко масштабировался и подстраивался под разные экраны.

Разработчики мобильных операционных систем создают свои рекомендации к созданию пользовательского интерфейса. Такие рекомендации создают свод правил, которым должны следовать сторонние разработчики. Данные правила именуется Guidelines. Гайдлайны необходимы для того, чтобы все приложения, разрабатываемые для операционной системы, строились по одному принципу, а пользователю, для достижения необходимых целей, необходимо следовать одним и тем же алгоритмам. Но это не означает, что разработчики и проектировщики дизайнов имеют жесткие рамки, скорее наоборот – гайдлайны облегчают разработку интерфейса и приложения становятся более понятны простому пользователю [18].

Процесс разработки мобильных приложений делится на несколько этапов:

1. Разработка технического задания – процесс подробного анализа и составления функционала приложения, выбора средства и среды разработки, определения временных рамок и возможных финансовых затрат.

2. Проектирование – процесс составления схемы экранов и их функционала, а также составление схемы взаимодействия между пользователем и приложением.

3. Создание концепции дизайна – процесс разработки максимально эффективного в плане использования дизайна приложения, на примере нескольких экранов. Будущий дизайн зависит от функционала и целей приложения, а также ее аудитории.

4. Разработка – процесс реализации проекта и создания рабочего прототипа приложения.

5. Тестирование – процесс проверки функционала приложения с целью выявления ошибок и недочетов, и дальнейшего анализа методов их устранения.

6. Отладка – процесс исправления выявленных ошибок и недочетов, добавления и улучшения имеющегося функционала.

7. Релиз – выпуск полноценной рабочей версии приложения в свободное использование. Возможны процессы дальнейшего поддержания работоспособности и улучшение функционала [16].

При выполнении выпускной работы была поставлена следующая цель: спроектировать и реализовать мобильное приложение по предмету “Основы системного моделирования и проектирования” под операционную систему Android.

Для достижения данной цели были поставлены следующие задачи:

- Выбор оптимального языка программирования
- Знакомство с операционной системой Android
- Знакомство с современной разработкой с учетом новейших особенностей платформы Android.
- Изучение технологий, используемых при разработке приложений

под Android платформу(Android SDK, SQLite, Google Service);

- Знакомство с необходимыми в процессе разработки инструментами (Android Studio, AVD Manager);

- Изучение общих подходов разработки программного обеспечения (ООП, клиент-серверная архитектура);

- Изучение общих подходов к тестированию кода и приложений в целом;

- Изучение основных паттернов проектирования для построения основной архитектуры приложения(MVC, MVVC и другие).

- Изучение жизненного цикла приложений

- Изучение методов хранения информации в данной операционной системе

- Изучение технологии ORM для упрощения работы с БД

- Изучение VK SDK и VK API

- Используя полученные навыки создание мобильного приложения под операционную систему Android «Расписание для вуза».

Этапы выполнения работы:

- Разработка структуры приложения;

- Проектирование дизайна приложения;

- Тестирование дизайна;

- Адаптация дизайна под разные размеры экранов;

- Выбор базы данных;

- Разработка структуры базы данных;

- Разработка базы данных;

- Тестирование базы данных;

- Адаптация работы дизайна приложения и базы данных.

### Глава 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО ПРЕДМЕТУ “ОСНОВЫ СИСТЕМНОГО МОДЕЛИРОВАНИЯ И ПРОЕКТИРОВАНИЯ ”

#### 3.2. Структура приложения и компоненты среды разработки

За основу структуры приложения был выбран паттерн MVC (Model View Controller). Схема представлена на рисунке 3.1.1.

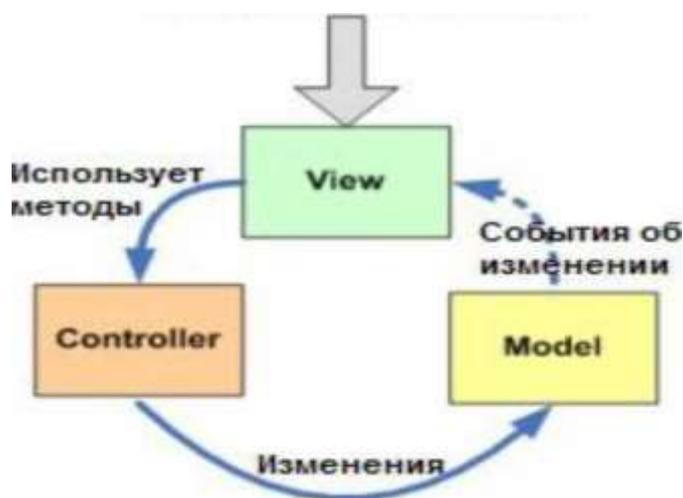


Рисунок 3.1. Схема паттерна Model View Controller

Модель это функциональная бизнес-логика. Признаки модели:

- бизнес-логика приложения заключена в модели;
- модель знает только себя и ничего о других элементах (контроллер и представление);
- модель это слой данных такая как база данных или XML-файл, но в некоторых случаях это менеджер базы данных, логика приложения или просто набор объектов.

Представление отображает данные полученные от модели. Представление не может и должно влиять на модель на прямую.

Признаки представления:

- Отображение данных полученные от модели;

Контроллер реагирует на события (действие пользователя), и сообщает, какие данные модели нужно дать на отображение представлению.

Признаки контроллера:

- Контроллер определяет, что должно быть отображено на представление;
- Представление может влиять на контроллер;
- Контроллер может влиять на модель и задать представление;

*Идея:*

Модель не зависит ни от контроллера, ни от представления, а представление и контроллер наоборот зависят от модели.

*Реализация:*

Контроллер перехватывает событие извне и реагирует на это событие, определенным образом, запрашивая данные у модели. После модель отдает запрошенные у неё данные напрямую представлению, и представление отображает эти самые данные.

Для дизайна Android приложений я использовал принципы Material Design результаты в приложении А.

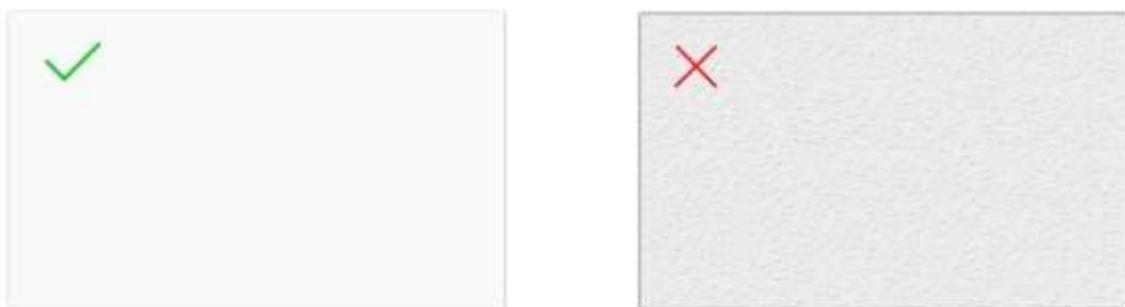
Основные принципы Material Design:

1. **Тактильные поверхности.** В Material слои располагаются на разной высоте и отбрасывают тени друг на друга, чтобы обеспечить пользователям лучшее понимание интерфейса и принципа взаимодействия с ним рисунок 3.2.
2. **Полиграфический дизайн.** В данном случае используется подход из традиционного графического дизайна: например, журнального и плакатного.
3. **Осмысленная анимация.** В реальном мире предметы не исчезают и не появляются из ниоткуда. Material Design даёт пользователям подсказки с помощью анимации.
4. **Адаптивный дизайн.** Как правильно применять перечисленные принципы на различных устройствах с разными размерами экранов [7].



*Рисунок 3.2. Тактильные поверхности*

Поверхность это некий контейнер с тенью, чтобы отличить один объект от другого и показать, как они расположены друг относительно друга. В Material Design необходимо стремиться к простоте и чистому дизайну.

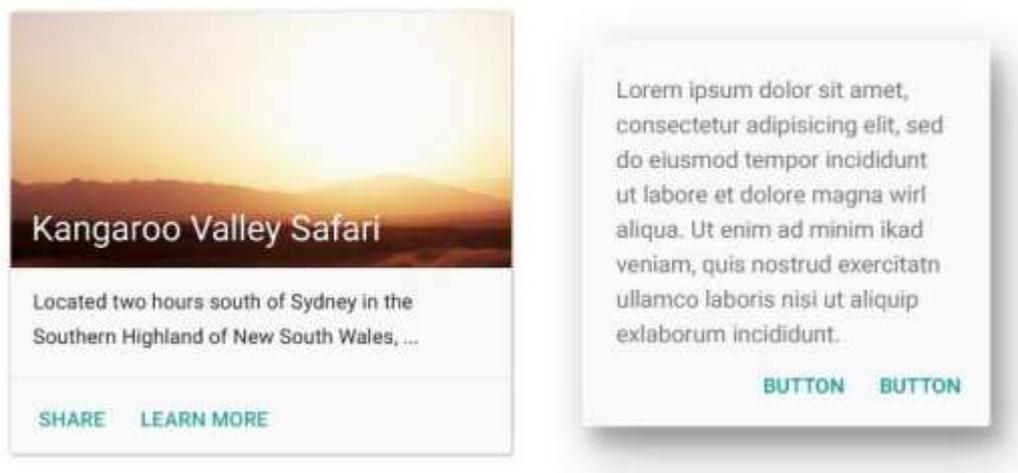


*Рисунок 3.3. Поверхности*

Не нужно использовать нестандартные текстуры и градиенты, чтобы показать светотень. Не рекомендуется давать визуальные свойства простая и в тоже время аккуратная тень способна выразить очень многое. Но не нужно забывать, что у каждой поверхности должна быть своя высота. Каждая поверхностей отбрасывает тень на нижнюю, как и в реальном мире (рисунок 3.4).

В традиционном дизайне следует избегать таких теней, как всяческих проявлений объема. Но нужно помнить, что они исполняют важную функцию обозначения иерархии и структуры элементов на экране. Например,

если подъем элемента больше, то и тень у него больше. Такая глубина поможет сфокусировать внимание пользователей на каких-то важных вещах и сделает это довольно изящно.



*Рисунок 3.4. Глубина*

Методы хранения информации в Android:

- Shared Preferences;
- Internal Storage;
- External Storage; • SQLite Databases;

Shared Preferences Shared Preferences — постоянное хранилище на платформе Android, обычно используется для хранения настроек приложения. Это хранилище является относительно постоянным, пользователь может зайти в настройки приложения и очистить данные приложения, тем самым очистив все данные в хранилище.

Для работы с данными постоянного хранилища нам понадобится экземпляр класса Shared Preferences, который можно получить у любого объекта, унаследованного от класса android.content.Context (например, Activity или Service). У объектов этих классов (унаследованных от Context) есть метод getSharedPreferences, который принимает 2 параметра:

name— выбранный файл настроек. Если файл настроек с таким именем не существует, он будет создан при вызове метода edit() и фиксации изменений с помощью метода commit().

mode— режим работы. Возможные значения:

MODE\_PRIVATE — используется в большинстве случаев для приватного доступа данным приложением-владельцем

MODE\_WORLD\_READABLE — только для чтения

MODE\_WORLD\_WRITEABLE — только записи

MODE\_MULTI\_PROCESS — несколько процессов совместно используют один файлSharedPreferences.

Internal Storage Internal phone storage (“Внутренняя память телефона”) – это второй объём памяти для пользовательских данных, включая загруженные приложения и их сохраненные данные (из RAM). В Android 2.2, она монтируется в /mnt/asec, в котором – арк-файлы загруженных приложений. Поэтому этот вид памяти называют Internal phone storage (Хранилище внутренней памяти телефона). Свободного пространства в Internal phone storage (“Внутренняя память телефона”) становится меньше, когда вы устанавливаете больше приложений. Может наступить момент, когда уже нельзя установить приложения – когда на Internal phone storage слишком мало свободного пространства.

Принципы:

- Оно всегда доступно;
- Файлы, которые сохранены здесь, по умолчанию доступны только вашему приложению;
- Не требуется запрашивать разрешения на чтение или запись;
- Когда пользователь деинсталлирует приложение, то система также удалит все файлы приложения.

External Storage

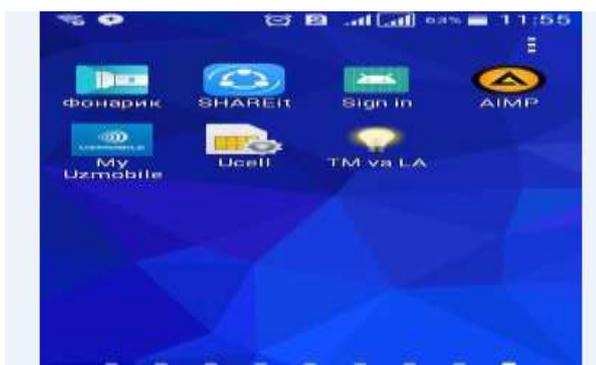
Это внешняя память, обычно в качестве неё используется micro sd карта. Принципы:

- Оно не всегда доступно, потому что пользователь может просто извлечь карту из устройства;

- Доступно на чтение для всех, так что файлы, сохраненные там, могут быть прочитаны без вашего ведома;
- Приложению требуется получить разрешение на чтение и запись в файле манифеста.
- Когда пользователь деинсталлирует приложение, система удалит файлы приложения из external storage, только если они сохранились в директории из getExternalFilesDir().

### 3.2. Описание функциональных возможностей мобильного приложения и инструкция по применению для пользователей

Для того чтобы, запустить работу мобильного приложения из списка установленных приложений на мобильном устройстве выбираем мобильное приложение “TM va LA”.



*Рисунок 3.5. Вид экрана мобильного устройства с установленным мобильным приложением*

На экране мобильного устройства открывается главное меню разработанного мобильного приложения.



*Рисунок 3.6. Запуск работы мобильного приложения.*

Меню мобильного приложения состоит из следующих разделов:

1. Ma`ruza matnlari (текст лекций)
2. Amaliy dars topshiriqlari (задания для практических занятий)
3. Test (тесты)
4. Nazorat savollari (контрольные вопросы)
5. Videorolik (видеоролики)
6. Mustaqil ish mavzulari (Темы самостоятельных работ)
7. Glossariy (Глоссарий)
8. Yordam (Помощь)
9. Muallif (Авторы)

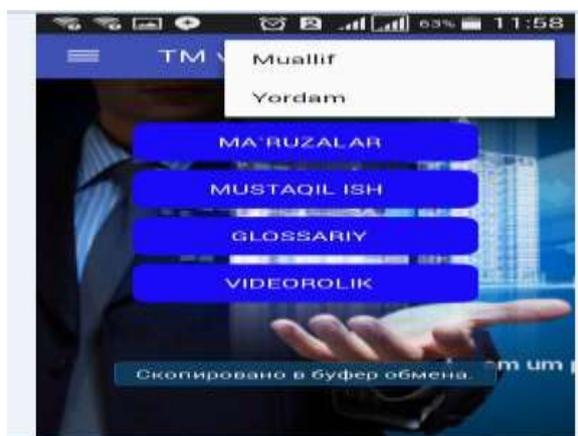


Рисунок 3.7. Основное окно мобильного приложения

1. В этом окне имеется 18 кнопок для выбора нужных лекций. При обращении к одному из них появляется текстовая информация. Вы можете вернуться, нажав клавишу возврата.



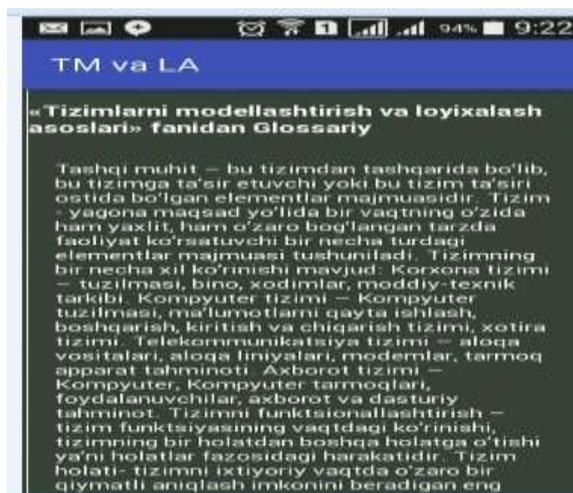
3.8–рисунок. Часть мобильного приложения по лекциям

2. Если выбрана кнопка «Amaliy mashg'ulotlar», на дисплее появится следующий модуль. Этот модуль содержит 9 практических занятий. Используя практические занятия по основам системного моделирования и проектирования можно найти материалы и задания по практическим занятиям. В также по каждой теме практических занятий для студентов предоставляются варианты.



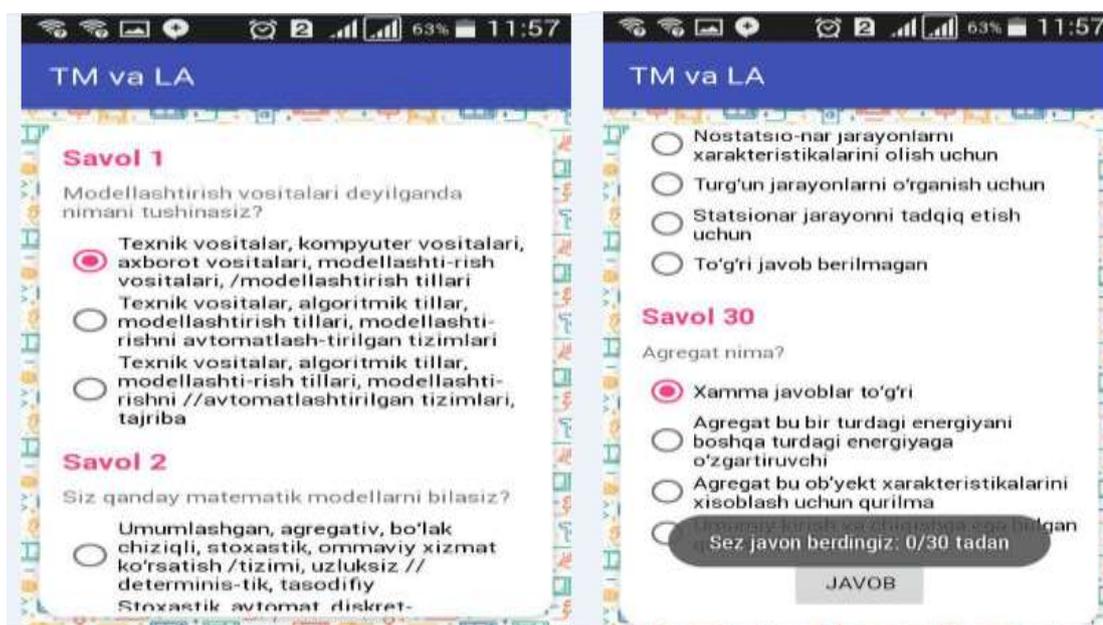
3.9. –рисунок. Модуль практических занятий и изложение выбранной практической работы

3. При нажатии кнопок “Mustaqil ish mavzulari” и “glossariy” на экране открываются следующие окна:



3.9 –рисунок. Окно программы “glossariy” и “mustaqil ish mavzulari”

4. При выборе меню “Test topshiriqlari” происходит следующий процесс выполнения тестовых заданий:



5. При выборе меню “Video” появляется список можно посмотреть видеоуроков по темам. Пользователь может выбрать нужный видеоурок.



3.10 – рисунок. Меню “video” мобильного приложения

6. Следующий модуль отображает эту справку, то есть отображается кнопка справки, схема обслуживания программного обеспечения. В этом

модуле пользователь сможет полностью ознакомиться рекомендациями по использованию мобильного приложения.

7. Последний модуль содержит информацию об авторе. У Пользователя есть возможность представить автору предложения о недостатках и по улучшению мобильного приложения .

### **3.3. Техника безопасности в работе на персональном компьютере**

Вероятность негативных последствий от использования персонального компьютера такая же, как и при эксплуатации другой бытовой техники. Ведь общеизвестно, что важно соблюдать меры безопасности с микроволновой печью, утюгом или электрочайником.

Пренебрежение элементарными рекомендациями имеет серьезные последствия для их владельцев. Компьютер – это такой же потенциальный источник угроз для здоровья, имущества и даже жизни пользователя.

Непрямой вред, который незаметен сразу, это ущерб здоровью:

- уже ни у кого не вызывает сомнений в существовании огромной негативной нагрузки на зрение, которое становится причиной его необратимого ухудшения, покраснений и синдрома «сухого глаза»;

- неправильная поза при работе за компьютером вызывает многочисленные заболевания суставов, грудной клетки и регулярные боли различного характера;

- бомба замедленного действия – чрезмерная нагрузка на психику пользователя. Необходимость постоянно концентрировать свое внимание на многочисленных деталях и смене картинок на мониторе становятся причиной переутомления.



## ОБЩИЕ ПРАВИЛА БЕЗОПАСНОСТИ

Техника безопасности при работе с компьютером на предприятии предусматривает наличие общедоступной инструкции, в которой указаны обязательные требования к обустройству рабочего места и процессу использования техники. Эти правила едины для всех организаций, их выполнение контролируется руководящими органами.

Основные правила организации пространства вокруг рабочего места:

- при длительном и интенсивном использовании, на поверхности модулей ПК (системный блок, монитор, мышка и т.д.) возникают небольшие разряды тока. Эти частицы активизируются во время прикосновений к ним и приводят к выходу техники из строя. Нужно регулярно использовать нейтрализаторы, увлажнители воздуха, антистатиками;

- вокруг стола не должно быть свисающих проводов, пользователь не должен контактировать с ними; важна целостность корпуса розетки и штепсельной вилки;

- отсутствие заземления предэкранного фильтра проверяется с помощью измерительных приборов;

- желательно во время строительных работ в офисе использовать минимальное количество легко воспламеняемых материалов (дерева,

пенопласта), а также горючего пластика в изоляции. Рекомендуется отдавать предпочтение кирпичу, стеклу, металлу и т.д.;

- помещение должно хорошо вентилироваться и охлаждаться в жаркую пору года. Важен своевременный отвод избыточного тепла от техники.

### ТРЕБОВАНИЯ БЕЗОПАСНОСТИ

Для работников офиса должен быть проведен устный базовый инструктаж, в дальнейшем его печатный текст должен предоставляться для подробного изучения. Организация в обязательном порядке размещает информационный лист на видном месте. Инструктаж охватывает полный цикл контакта человека с компьютером. Он начинается с установки оборудования сервисной службой и заканчивается утилизацией непригодного устройства.

### ПЕРЕД НАЧАЛОМ РАБОТЫ

Даже если речь идет о рабочем месте, которое используется каждый день и регулярно проверяется специалистами (как, например, в офисе или учебном заведении), нельзя терять бдительность.

Перед тем, как включить компьютер, необходимо уделить пару минут следующим действиям:

- нужно убедиться в том, что в зоне досягаемости отсутствуют оголенные провода и различные шнуры. Они не только мешают работе, но и несут потенциальную опасность в случае короткого замыкания;



- нельзя начинать работу на технике с видимым повреждением. В случае обнаружения трещины на корпусе или повреждений другого рода, нужно обратиться за помощью в сервисный центр. Это же относится к ПК с неисправным индикатором включения/выключения.

- предметы на столе не должны мешать обзору, пользованию мышкой и клавиатурой. Поверхность экрана должна быть абсолютно чистой;

- на системном блоке не должно находиться никаких предметов, так как в результате вибраций может нарушиться работа устройства. Нужно убедиться в том, что никакие посторонние предметы не мешают работе системе охлаждения.

- недопустимо включать персональный компьютер в удлинители и розетки, в которых отсутствует заземляющая шина.



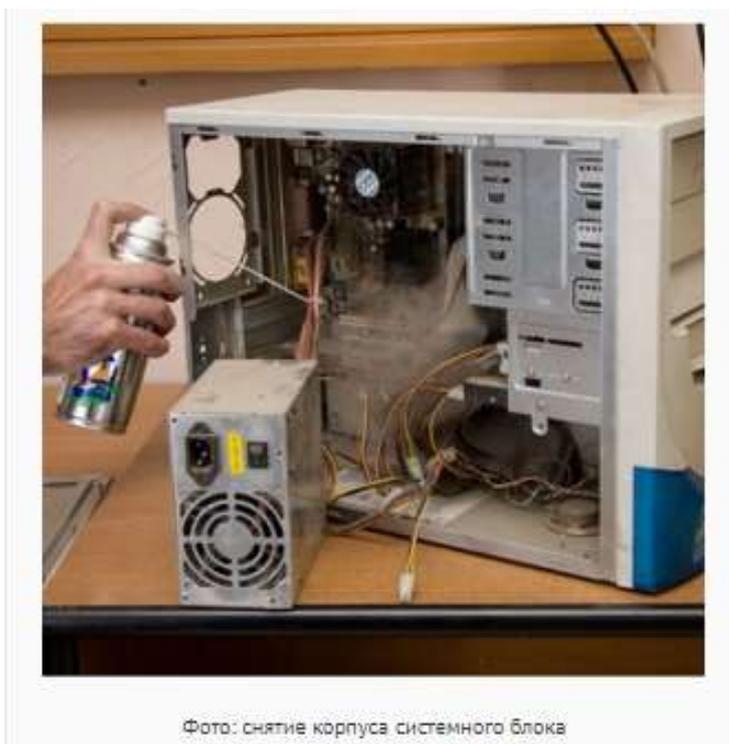
- запрещается начинать работу в помещениях с повышенной влажностью, а также в случае, если рядом присутствуют открытые источники влаги (лужи, мокрый пол). Включить технику можно лишь после полного высыхания окружающих предметов.

- недопустимо часто включать и выключать компьютер в течение рабочего дня без особой нужды. Система просто не справляется с необходимостью быстро сворачивать все процессы.

## ПРИ ВЫПОЛНЕНИИ РАБОТЫ

Поскольку персональный компьютер обладает всеми свойствами электрического прибора, то на него распространяются основные правила безопасности при взаимодействии с проводниками тока:

- нельзя размещать какие-либо вещи на проводах, а также самостоятельно менять их расположение без особой нужды;
- рекомендуется избегать расположения жидкостей рядом с модулями компьютера. Поэтому кулер с водой или кофейный автомат необходимо размещать в стороне от рабочих мест в офисе. Пользователи должны осознавать опасность потенциального замыкания в случае пролития воды на клавиатуру или системный блок. Нельзя работать на ПК с мокрыми руками;
- нельзя очищать поверхность компьютера от загрязнений, когда он находится во включенном состоянии;
- недопустимо снимать корпус любой из составных частей ПК во время его работы.

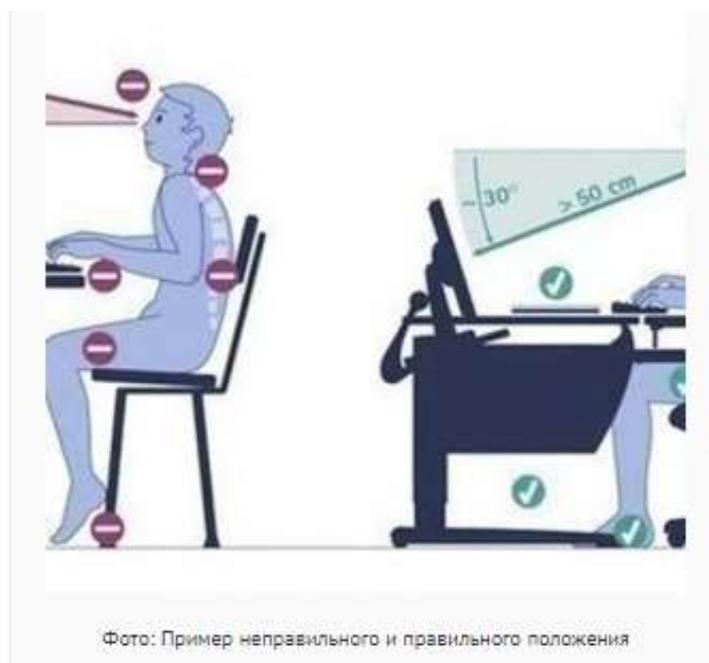


Кроме того, разбор и ремонт техники имеют совершают только специализированные работники:

- во время работы на компьютере нельзя одновременно прикасаться к другим металлическим конструкциям, которые стоят на той же поверхности. Это касается отопительных батарей или трубопроводов;
- в помещении с компьютерами непозволительно курить или употреблять пищу непосредственно на рабочем месте;
- при ощущении даже незначительного запаха гари, нужно как можно быстрее выключить ПК из сети и обратиться к ответственному за обслуживание компьютерной техники.

Как было сказано выше, неправильная работа с персональным компьютером таит в себе множество угроз для здоровья человека. Что бы минимизировать это влияние даже при длительном нахождении за монитором, стоит навсегда запомнить следующие постулаты:

- расстояние между глазами пользователя и экраном составляет не менее полуметра. Но пользователь должен быть в состоянии дотянуться кончиками пальцев до верхнего края монитора;



- клавиатура размещается за 20-30 сантиметров от края стола;

- стул стоит таким образом, что бы спина лишь немного упиралась в его спинку. Высота сидения позволяет держать ровную осанку;
- локти согнуты под прямым углом, а в кистях рук, лежащих на столе, не чувствуется напряжения;
- локти не висят в воздухе, а комфортно располагаются на подлокотниках кресла или столешнице. Их позиция существенно не меняется при передвижении мышки;
- ноги упираются в твердую поверхность, распрямлены вперед, а не подогнуты под себя;
- если пользователь носит очки, то нужно убедиться в том, что он может свободно регулировать угол наклона экрана. чрезвычайно важна периодическая зарядка. Каждый час нужно вставать с кресла, разминать мышцы и суставы. Ведь, несмотря на неподвижность, они испытывают огромную нагрузку, пребывая в неестественном положении. Обязательно нужно делать разминку для глаз: круговые и линейные движения открытыми глазами, моргание и расфокусирование.

## В АВАРИЙНЫХ СИТУАЦИЯХ

Своевременная бдительность поможет избежать опасных ситуаций для жизни и сохранить целостность техники.

Действия в аварийных ситуациях:

- при неполадках любого рода в электроснабжении устройства необходимо сразу отключить компьютер от сети;
- если обнаружен оголенный провод, то необходимо оперативно оповестить всех работников офиса, не допуская чьего-либо контакта с ним;
- в каждом учреждении должны находиться огнетушители ОУБ-3 или ОУ-2, а также ведра и полотна в необходимом количестве. Персонал обязан знать о том, где находятся средства для гашения пламени и куда нужно звонить в случае пожара;

- при поражении человека электрическим током, прежде всего, оказывается первая помощь: искусственное дыхание и внешний интенсивный массаж сердца. В первые же мгновения после удара током, вызывается скорая помощь.

## ПО ОКОНЧАНИИ РАБОТЫ

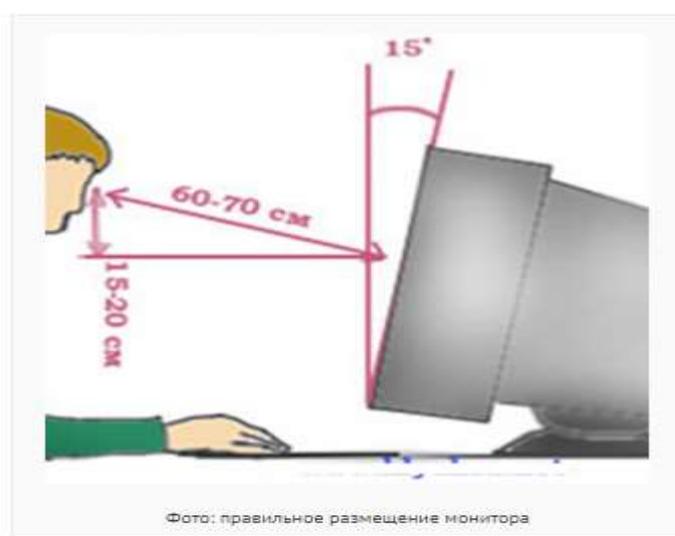
Перед завершением нужно правильно закрыть все программы и окна. Нельзя оставлять активные носители информации (диски и флэшки). Стоит отметить, что порядок выключения составляющих частей ПК отличается от порядка их включения ровно наоборот.

Запуск компьютера происходит по цепочке: общее питание – периферия – системный блок. Выключение, соответственно, начинается с системного блока. Вытаскивать штепсельную вилку необходимо крепко держась за её корпус. Нельзя совершать резких рывков и тем более тянуть за провод. После завершения работы, желательно устранять лишнее статическое напряжение с поверхности электроприборов и проводить влажную уборку рабочего места [5].

## ПРАВИЛА РАЗМЕЩЕНИЯ МОНИТОРА

- монитор находится на расстоянии равной длине руки пользователя. Если диагональ экрана более двадцати дюймов, то дистанция больше;
- глаза находятся на уровне линии, которая на 5 сантиметров ниже верхнего края экрана. Для этого регулируется высота стула и монитора;
- экран находится по центру для того, что бы не нужно было задерживать шею в неестественном положении;
- окна в помещении не должны создавать блики на экране. Свет от окна не ярче, чем свет от ПК;

- чрезмерная контрастность и яркость изображения утомляют зрение, необходимо настраивать эти показатели;
- в офисах, где компьютеры расположены в два ряда, нужно устанавливать защитный промежуточный экран для устранения избыточного облучения.



Если подобная защита отсутствует, то минимальное расстояние от ближайшего экрана – не менее двух метров [5].

### ТРЕБОВАНИЯ К РАБОЧЕМУ МЕСТУ

Минимальная площадь рабочего места для одного человека – 6 м<sup>2</sup>. Свет должен исходить от искусственных и естественных источников. Лампы не отвечиваются от экрана, а избыток солнечных лучей необходимо сдерживать тканевыми шторами. Нежелательно освещать помещение исключительно с помощью потолочного верхнего света.

Недопустимо размещать компьютерные провода рядом с отопительной системой, их изоляция должна быть целостной. Системный блок не должен стоять в нише стола или другом замкнутом пространстве, где нарушена нормальная вентиляция.



Для исключения всевозможных рисков, нужно ответственно относиться ко всем этапам использования компьютера. Пользователь может и должен контролировать весь цикл взаимодействия с техникой. Процесс соблюдения всех этих несложных правил должен быть непрерывным и комплексным [5].

## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы было спроектировано и разработано мобильное приложение по предмету «Основы системного моделирования и проектирования» для ОС Android. В ходе работы были изучены основные методы и технологии разработки мобильных приложений для ОС Android, общие подходы к тестированию приложений, жизненный цикл приложения, методы хранения информации, технология ORM, VK SDK и VK API.

Внедрение мобильного приложения предполагает совершенствование процесса обучения, повышение его эффективности и качества.

В данной выпускной квалификационной работе приводятся теоретические основы разработки мобильного приложения, описывается технология ее создания и разработки. Для достижения цели выпускной квалификационной работы выполнены следующие задачи:

1. Изучен язык программирования Java
2. Изучена технология программирования и архитектура ОС Android.
3. Разработана структура приложения
4. Спроектирован дизайн.
5. Адаптирован дизайн под различные размеры экранов
6. Разработана база данных приложения для SQLite
7. Произведена локализация приложения на английский язык
8. Разработано мобильное приложение для ОС Android.

Подводя итоги данной работы, можно сказать, что было разработано мобильное приложение по предмету «Основы системного моделирования и проектирования» для ОС Android.

Практическая значимость выпускной квалификационной работы заключается в разработке рекомендаций по созданию мобильного приложения.

## Список использованной литературы

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. Больше книг об Android - <http://4pda.ru/forum/index.php?showtopic=159318>
3. Блог разработчика Android - <http://androidengineer.ru/>
4. Вопросы и ответы по Java J2ME [Электронный ресурс]. URL: Google Android программирование для мобильных устройств (Голощанов А.Л., 2011, BHV Санкт-Петербург).
5. Курс лекций «Техника безопасности при работе на компьютере».
6. Леонтьев С. Использование мобильных телефонов в разных странах. <http://www.cossa.ru/articles/152/37433/>
7. Разработка приложений для Android (С. Хашими, С. Коматинени, Д. Маклинг, 2011).
8. Топ Обзор. Обзор 10+ облачных хранилищ данных [Электронный ресурс]. 2016. URL: <http://www.topobzor.com/obzor-10-oblachnyx-xranilishh-dannux/.html> (дата обращения: 18.02.2017).
9. Android 2. Программирование приложений для планшетных компьютеров и смартфонов (Рето Майер, Эксмо, 2011).
10. Android Developers [Электронный ресурс]. 2016. URL: <http://developer.android.com> (дата обращения: 05.02.2017).
11. AWS Storage Services Overview. A Look at Storage Services Offered by AWS. Vol. 48, Amazon Web Services, Inc., December 2016.
12. AWS Mobile SDK. Set Up the AWS Mobile SDK for Android [Электронный ресурс]. URL: <http://docs.aws.amazon.com/mobile/sdkforandroid/developerguide/setup.html> (дата обращения: 20.03.2017).
13. AWS Mobile SDK. Store and Retrieve Files with Amazon S3 [Электронный ресурс]. URL: <https://docs.aws.amazon.com/mobile/>

[sdkforandroid/developerguide/s3transferutility.html](#) (дата обращения: 20.03.2017).

14. Bransford M., Douglas J. How People Learn: Brain, mind, experience, and school. Washington (D.C.), 2000.

15. Felker, Donn, and Joshua Dobbs. Android App Development for Dummies. 3rd ed. Vol. 357, Indianapolis, Indiana, Wiley Publishing, Inc., 2011.

16. Firebase. Authentication [Электронный ресурс]. 2016. URL: <https://firebase.google.com> (дата обращения: 05.04.2017).

17. Geddes S. Mobile Learning in the 21st Century: Benefit for Learners. 2004. URL: <http://knowledgetree.flexiblelearning.net.au/edition06/download/geddes.pdf>

18. Kukulska-Hulme A. Mobile Learning for Quality Education and Social Inclusion. UNESCO IITI. M., 2010.

19. Kumar S. Blackboards to Blackberries: Mobile Learning Buzzes across Schools and Universities. 2010. URL: <http://www.learningsolutionsmag.com/authors/315/sesh-kumar>

20. Mobile Learning: What it is, why it matters, and how to incorporate it into your learning strategy. 2008. URL: <http://www.m-learning.org/knowledge-centre/whatismlearning>

21. Rekkedal T., Dye A. Mobile Distance Learning with PDAs // Mobile Learning: Transforming the Delivery of Education and Training / Ed. by M. Ally. 2009. URL: <http://www.aupress.ca/index.php/books/120155>

22. Traxler J. Current State of Mobile Learning // Mobile Learning: Transforming the Delivery of Education and Training. 2009. URL: <http://www.aupress.ca/index.php/books/120155>

## Приложение (код программы)

```
<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=".BoshActivity"
    tools:showIn="@layout/app_bar_bosh">

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

        <Button
            android:id="@+id/idbtnActivity1"
            android:layout_margin="5dp"
            android:layout_width="205dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="@drawable/button"
            android:text="ma`ruzalar"
            android:textColor="#FFFCFCFC"
            android:textSize="15dp"
            android:onClick="btnClickAct1"
            tools:ignore="OnClick"
        />

        <Button
            android:id="@+id/idbtnActivity2"
            android:layout_margin="5dp"
            android:layout_width="205dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="@drawable/button"
            android:text="Mustaqil ish"
            android:textColor="#FFFCFCFC"
            android:textSize="15dp"
            android:onClick="btnClickAct2"
            tools:ignore="OnClick" />

        <Button
            android:id="@+id/idbtnActivity3"
            android:layout_margin="5dp"
            android:layout_width="205dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="@drawable/button"
            android:text="glossariy"
            android:textColor="#FFFCFCFC"
            android:textSize="15dp"
            android:onClick="btnClickAct3"
            tools:ignore="OnClick" />

        <Button
            android:id="@+id/idbtnActivity4"
            android:layout_margin="5dp"
            android:layout_width="205dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
```

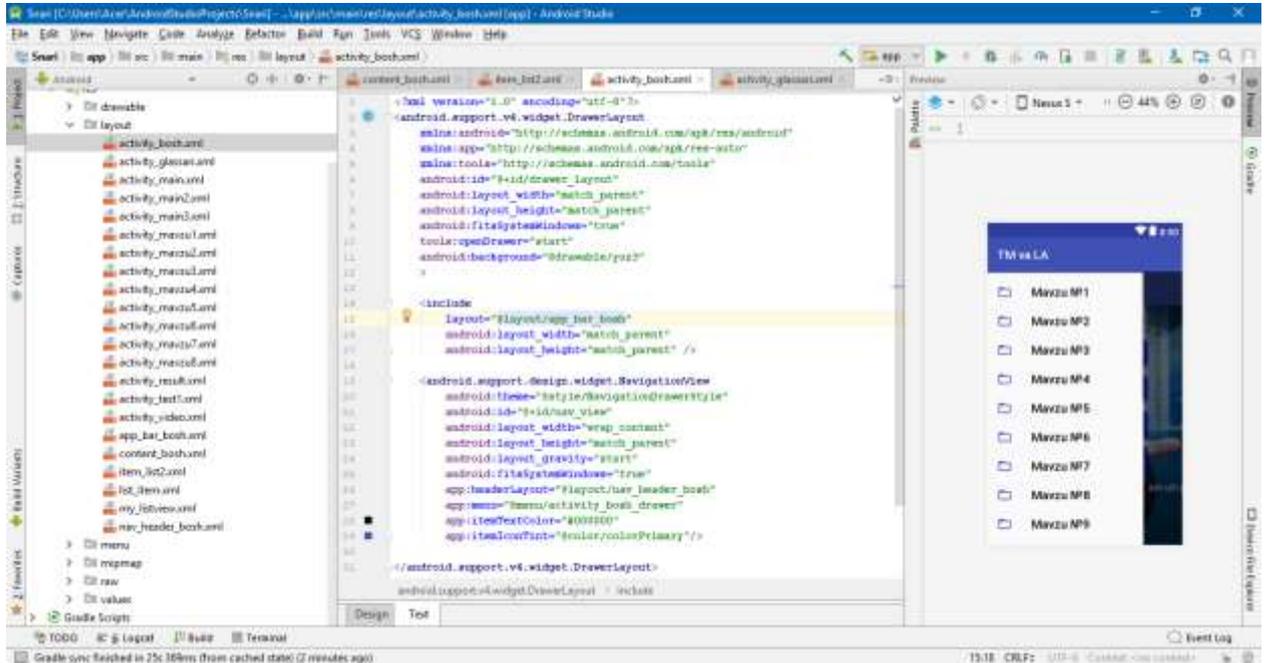
```

        android:background="@drawable/button"
        android:text="Videorolik"
        android:textColor="#FFFCFCFC"
        android:textSize="15dp"
        android:onClick="btnClickAct4"
        tools:ignore="OnClick" />

```

```
</LinearLayout>
```

```
</LinearLayout>
```



```

<?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"
    android:background="@drawable/yuz3"
    >

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

    <android.support.design.widget.NavigationView
        android:theme="@style/NavigationDrawerStyle"
        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"
        app:itemTextColor="#000000"
        app:itemIconTint="@color/colorPrimary" />

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

```

# JAVA

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

import android.content.Intent;
import android.content.res.ColorStateList;
import android.graphics.Color;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.text.SpannableString;
import android.text.style.TextAppearanceSpan;
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;

public class BoshActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener,
    View.OnClickListener {
    Button btn1, btnActivity1, btnActivity2, btnActivity3;
    String key = "id";

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

        btnActivity1 = (Button) findViewById(R.id.idbtnActivity1);
        btnActivity3 = (Button) findViewById(R.id.idbtnActivity3);
        btnActivity2 = (Button) findViewById(R.id.idbtnActivity2);
        btnActivity2.setOnClickListener(this);

        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);

        //navigationView.setItemTextColor(ColorStateList.valueOf(Color.WHITE));

    }

    @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 onNavigationItemSelected(MenuItem item) {
        // Handle navigation view item clicks here.
        int id = item.getItemId();

        if (id == R.id.nav_camera) {
            Intent intent = new Intent(this, Mavzu1Activity.class);
            startActivity(intent);
            // Handle the camera action
        } else if (id == R.id.nav_gallery) {
            Intent intent = new Intent(this, Mavzu2Activity.class);
            startActivity(intent);
        } else if (id == R.id.nav_slideshow) {
            Intent intent = new Intent(this, Mavzu3Activity.class);
            startActivity(intent);
        } else if (id == R.id.nav_manage) {
            Intent intent = new Intent(this, Mavzu4Activity.class);
            startActivity(intent);
        } else if (id == R.id.mavzu1) {
            Intent intent = new Intent(this, Mavzu5Activity.class);
            startActivity(intent);
        } else if (id == R.id.mavzu2) {
            Intent intent = new Intent(this, Mavzu6Activity.class);
            startActivity(intent);
        } else if (id == R.id.mavzu3) {
            Intent intent = new Intent(this, Mavzu7Activity.class);
            startActivity(intent);
        } else if (id == R.id.mavzu4) {
            Intent intent = new Intent(this, Mavzu7Activity.class);

```

```

        startActivity(intent);
    } else if (id == R.id.mavzu5) {
        Intent intent = new Intent(this, Mavzu7Activity.class);
        startActivity(intent);
    } else if (id == R.id.mavzu6) {
        Intent intent = new Intent(this, Mavzu7Activity.class);
        startActivity(intent);
    } else if (id == R.id.mavzu7) {
        Intent intent = new Intent(this, Mavzu7Activity.class);
        startActivity(intent);
    } else if (id == R.id.mavzu8) {
        Intent intent = new Intent(this, Mavzu7Activity.class);
        startActivity(intent);
    } else if (id == R.id.mavzu9) {
        Intent intent = new Intent(this, Mavzu7Activity.class);
        startActivity(intent);
    } else if (id == R.id.mavzu10) {
        Intent intent = new Intent(this, Mavzu7Activity.class);
        startActivity(intent);
    } else if (id == R.id.mavzu11) {
        Intent intent = new Intent(this, Mavzu7Activity.class);
        startActivity(intent);
    }

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

public void btnClickAct1(View view) {
    Intent i = new Intent(this, Main3Activity.class);
    startActivity(i);
}

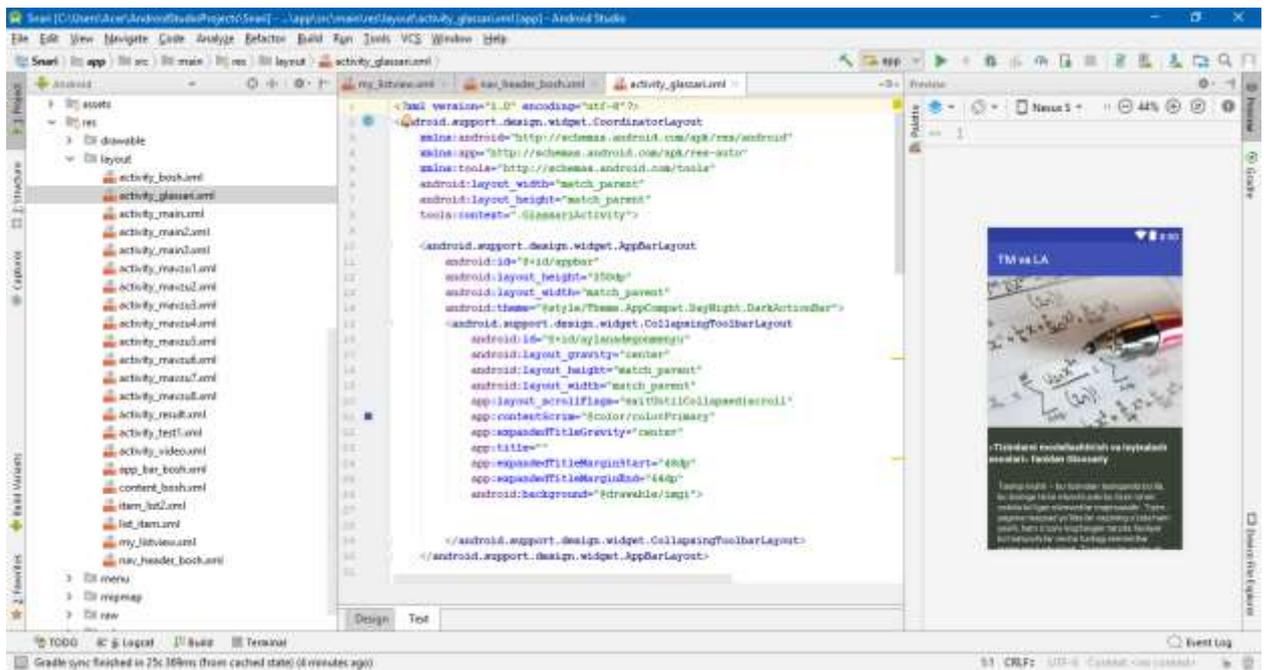
public void btnClickAct3(View view) {
    Intent i = new Intent(this, GlassariActivity.class);
    startActivity(i);
}

public void btnClickAct4(View view) {
    Intent i = new Intent(this, VideoActivity.class);
    startActivity(i);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {

        case R.id.idbtnActivity2:
            Intent intent = new Intent(BoshActivity.this,
ResultActivity.class);
            intent.putExtra(key, "12");
            startActivity(intent);
            break;
    }
}
}}

```



# XML

```

<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    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=".GlassariActivity">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/appbar"
        android:layout_height="250dp"
        android:layout_width="match_parent"
        android:theme="@style/Theme.AppCompat.DayNight.DarkActionBar">
        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/aylanadegonmenu"
            android:layout_gravity="center"
            android:layout_height="match_parent"
            android:layout_width="match_parent"
            app:layout_scrollFlags="exitUntilCollapsed|scroll"
            app:contentScrim="@color/colorPrimary"
            app:expandedTitleGravity="center"
            app:title=""
            app:expandedTitleMarginStart="48dp"
            app:expandedTitleMarginEnd="64dp"
            android:background="@drawable/img1">

            </android.support.design.widget.CollapsingToolbarLayout>
        </android.support.design.widget.AppBarLayout>

        <android.support.v4.widget.NestedScrollView
            android:layout_height="match_parent"
            android:layout_width="match_parent"
            app:layout_behavior="@string/appbar_scrolling_view_behavior">

            <LinearLayout
                android:background="#384336"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center"
        android:clipToPadding="false">
<android.support.v7.widget.CardView
    android:background="#2ce212"
    android:id="@+id/btn1"
    android:foreground="?android:attr/selectableItemBackground"
    android:clickable="true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="1dp">
<LinearLayout
    android:background="#384336"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="fill_horizontal">

    <TextView
        android:textSize="16sp"
        android:layout_gravity="center_horizontal"
        android:textColor="#fffefe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:layout_marginTop="20sp"
        android:text="«Tizimlarni modellashtirish va
loyixalash asoslari» fanidan Glossariy"/>

    <TextView
        android:padding="15dp"
        android:layout_gravity="center_horizontal"
        android:textColor="#fffefe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle=""
        android:layout_marginTop="10dp"
        android:text="@string/s"/>

</LinearLayout>
</android.support.v7.widget.CardView>
</LinearLayout>
</android.support.v4.widget.NestedScrollView>
</android.support.design.widget.CoordinatorLayout>

```

## JAVA

```

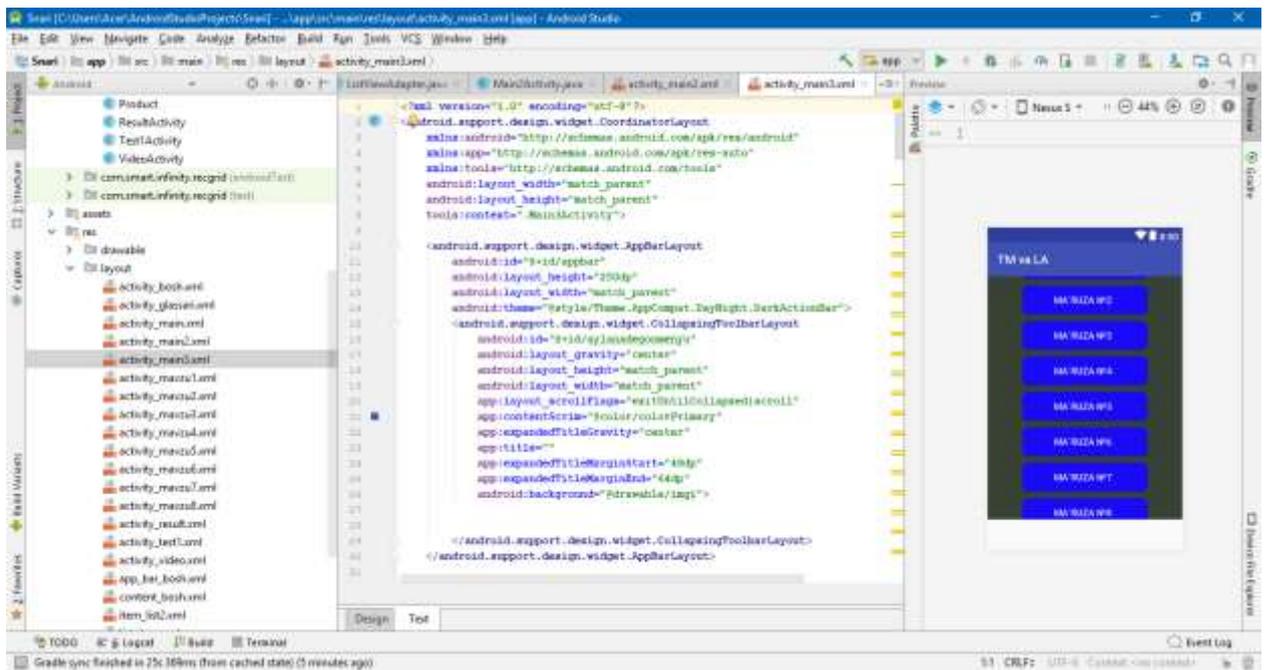
package com.smart.infinity.recgrid;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class GlassariActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_glassari);
    }
}

```



# XML

```

<?xml version="1.0" encoding="utf-8" ?>
<android.support.design.widget.CoordinatorLayout
    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=".Main3Activity">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/appbar"
        android:layout_height="250dp"
        android:layout_width="match_parent"
        android:theme="@style/Theme.AppCompat.DayNight.DarkActionBar">
        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/aylanadegonmenyu"
            android:layout_gravity="center"
            android:layout_height="match_parent"
            android:layout_width="match_parent"
            app:layout_scrollFlags="exitUntilCollapsed|scroll"
            app:contentScrim="@color/colorPrimary"
            app:expandedTitleGravity="center"
            app:title=""
            app:expandedTitleMarginStart="48dp"
            app:expandedTitleMarginEnd="64dp"
            android:background="@drawable/img1">

            </android.support.design.widget.CollapsingToolbarLayout>
        </android.support.design.widget.AppBarLayout>

        <android.support.v4.widget.NestedScrollView
            android:layout_height="match_parent"
            android:layout_width="match_parent"
            app:layout_behavior="@string/appbar_scrolling_view_behavior">

            <LinearLayout
                android:background="#384336"

```

```

android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:gravity="center"
android:clipToPadding="false">
<android.support.v7.widget.CardView
    android:background="#2ce212"
    android:id="@+id/btn1"
    android:foreground="?android:attr/selectableItemBackground"
    android:clickable="true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="1dp">
<LinearLayout
    android:background="#384336"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="fill_horizontal">

<Button
    android:id="@+id/idbtnActivity1"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №1"
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity2"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma` ruza №2"
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity3"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №3"
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity4"
    android:layout_margin="5dp"

```

```

        android:layout_width="205dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:background="@drawable/button"
        android:text="        ma`ruza №4        "
        android:textColor="#FFFCFCFC"
        android:textSize="15dp"
        android:onClick="btnClickAct1"
        tools:ignore="OnClick"
    />
<Button
    android:id="@+id/idbtnActivity5"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №5        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
    />
<Button
    android:id="@+id/idbtnActivity6"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №6        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
    />
<Button
    android:id="@+id/idbtnActivity7"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №7        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
    />
<Button
    android:id="@+id/idbtnActivity8"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №8        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
    />
<Button

```

```

        android:id="@+id/idbtnActivity9"
        android:layout_margin="5dp"
        android:layout_width="205dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:background="@drawable/button"
        android:text="        ma`ruza №9"
        android:textColor="#FFFCFCFC"
        android:textSize="15dp"
        android:onClick="btnClickAct1"
        tools:ignore="OnClick"
    />
<Button
    android:id="@+id/idbtnActivity10"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №10"
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity11"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma ruza №11"
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity12"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №12"
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity13"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma ruza №13"
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"

```

```

/>
<Button
    android:id="@+id/idbtnActivity14"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №14        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity15"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №15        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity16"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №16        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity17"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №17        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"
    android:onClick="btnClickAct1"
    tools:ignore="OnClick"
/>
<Button
    android:id="@+id/idbtnActivity18"
    android:layout_margin="5dp"
    android:layout_width="205dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/button"
    android:text="        ma`ruza №18        "
    android:textColor="#FFFCFCFC"
    android:textSize="15dp"

```

```
        android:onClick="btnClickAct1"
        tools:ignore="OnClick"
    />

    </LinearLayout>
    </android.support.v7.widget.CardView>
</LinearLayout>
</android.support.v4.widget.NestedScrollView>
</android.support.design.widget.CoordinatorLayout>
```