Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

Разработка мобильной игры-рогалик в жанре экшен

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения  
для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.202500.010.000ПЗ

Разработал:

студент группы ПР-21.102

Вершинин А.И.

**Содержание**

[Введение 3](#_Toc117112970)

[1Исследовательский раздел 3](#_Toc117112971)

[1.1Описание предметной области 4](#_Toc117112972)

[1.2Образ клиента 5](#_Toc117112973)

[1.3Сценарии 5](#_Toc117112974)

[1.4Сбор и анализ прототипов 5](#_Toc117112975)

[2Проектирование приложения 11](#_Toc117112976)

[2.1UI/UX дизайн проекта 11](#_Toc117112977)

[2.2Выбор технологии, языка и среды программирования 16](#_Toc117112978)

[3Разработка мобильного приложения 17](#_Toc117112979)

[3.1Разработка базы данных 17](#_Toc117112980)

[3.2Разработка мультимедийного контента 17](#_Toc117112981)

[3.3Описание используемых плагинов 20](#_Toc117112982)

[3.4Описание разработанных процедур и функций 23](#_Toc117112982)

[4Тестирование 30](#_Toc117112983)

[4.1Протокол тестирования дизайна приложения 30](#_Toc117112984)

[4.2Протокол тестирования функционала приложения 34](#_Toc117112985)

[Заключение 39](#_Toc117112986)

[Библиография 40](#_Toc117112987)

[Приложение А(обязательное) Техническое задание 41](#_Toc117112988)

Введение

В этом месте прописывается актуальность и значимость темы курсового проекта.

Целью курсового проекта является создание мобильного приложения видеохостинга.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области;
* рассмотрение приложения с точки зрения пользователя для выявления функций приложения;
* разработка и реализация дизайна приложения;
* написание кода приложения;
* тестирование полученного продукта.

Объект исследования – мобильное приложение видеохостинга.

Предмет исследования – изучение принципов функционирования и инструментов приложения.

# Исследовательский раздел

## Описание предметной области

Предметная область данного курсового проекта связана с разработкой мобильной игры в жанре экшн-рогалики shoot-emup. Экшен-рогалики представляют собой игры с ролевыми элементами, где пользователь пытается пройти как можно больше уровней без проигрыша. После смерти персонаж теряет значительную часть прогресса, а прохождение начинается с первого уровня. В некоторых играх прогресс закрепляется: например, сюжет игры всё же продвигается или игроку открываются новые игровые механики.

## Образ клиента

Любители сложных мобильных игр всех возрастных категорий. Особенно игра будет заинтересовывать игроков, которые хотят от игры большой сложности и нового опыта. Вот несколько категорий пользователей, которые могут насладиться функциональностью приложения:

Любители инди-проектов: любители малобюджетных игр, где авторы не ограничены рамками и могут придумывать разные механики и стили.

Геймеры, которым нравится сложность: люди, которые любят игры с вызовом, где необходимо пройти сложные уровни.

Фанаты игрового жанра «Rogue-like» : люди, которым нравится проходить игру больше 1 раза и всегда получать новый опыт.

## Сценарии

Александр устал от однообразных triple A проектов , где разработчики должны делать игру чтобы она окупилась . На просторах

Иван любитель сложных проектов, но их сейчас на рынке мало. В один прекрасный день он нашел мобильную игру-рогалик в жанре экшен. Он заинтересовал его уникальным геймплеем и своей сложностью. Иван был удовлетворен.

## Сбор и анализ прототипов

«VampireSurvivors» – это игра в жанре экшн-рогалик, разработанная и изданная инди-разработчиком poncle, где игрок становится "буллет-хеллом" и сражается с многочисленными волнами врагов. Изначально игра была написана с использованием HTML5 и была бесплатно выпущена на itch.io как браузерная игра 31 марта 2021 года

"Soulstonesurvivors" – это рогалик с элементами RPG, в котором вам придётся уничтожать орды монстров и боссов, попутно прокачивая 14 персонажей с разными специальностями и оружием. В процессе каждого забега против монстров вам необходимо будет каждый раз создавать свой собственный билд из 150 доступных умений (дальше разработчики обещают ещё больше).

В GooglePlay и AppStoreсуществует два приложения, выполняющих роль видеохостинга. Первое – YouTube (Американское приложение), второе – RUTUBE (Русское приложение). Оба этих приложения имеют возможность просматривать видеоролики, сохранять понравившиеся и загружать собственные видеоролики с возможность поделиться ими. Рассмотрим эти два приложения подробнее и сравним их.

Оба этих приложения имеют главный экран, на котором собраны видеоролики от разных авторов (рисунок 1).

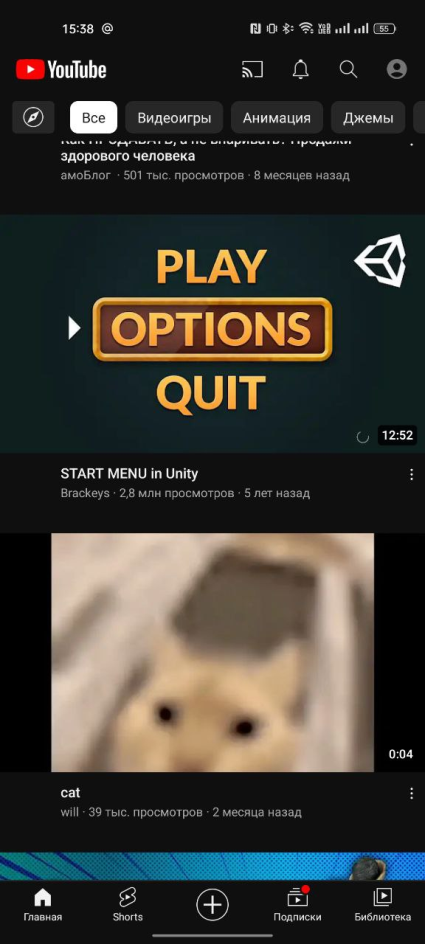
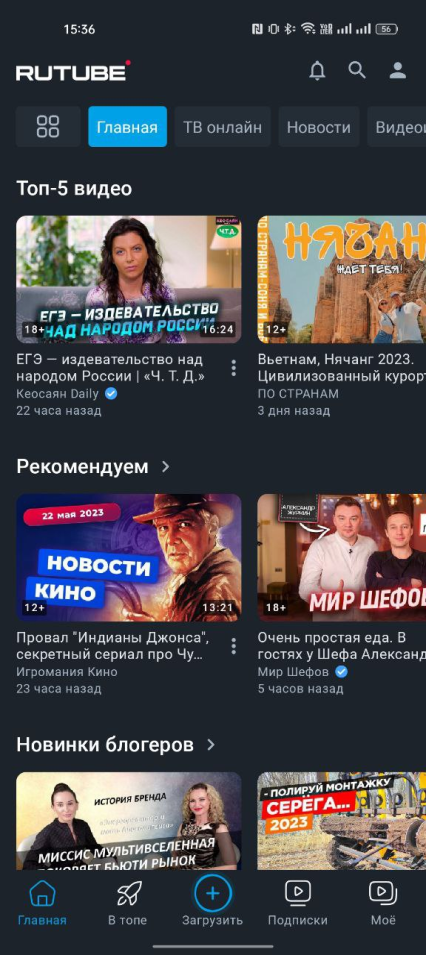


Рисунок 1 – Главный экран приложений

В приложениях на главном экране можно увидеть нижнюю и верхнюю панель, отвечающие за навигацию по приложению. На верхней панели в обоих приложениях почти одинаковые кнопки:

* уведомления;
* поиск;
* профиль.

Нижняя панель отличается только элементами, к которым она перенаправляет, но общими являются:

* главная;
* загрузить;
* подписки.

Вывод видеороликов отличается. В приложении «RUTUBE» - видео выводятся по группам, а в «YouTube» – как общий список рекомендованных видео. Логотип находится в одинаковом месте – в верхнем левом углу.

Так же в них реализована функция добавления видеороликов в «Понравившиеся» (рисунок 2):

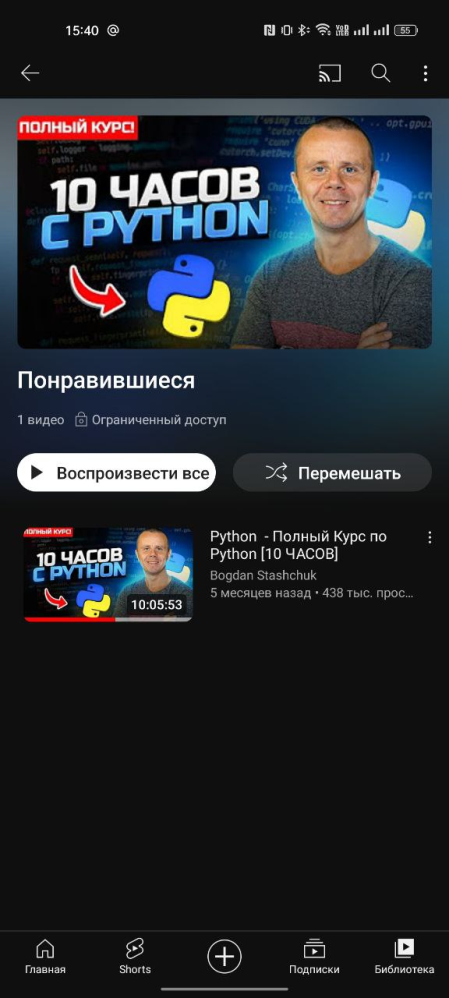


Рисунок 2 – Экран с понравившимися видеороликами

Все добавленные в «Понравившиеся» видеоролики, отображаются на отдельной вкладке. Их так же можно оттуда удалить или посмотреть повторно. Однако в приложении «RUTUBE» на этой вкладке отсутствует верхняя панель. Поэтому в нём нельзя найти нужный видеоролик с помощью поиска, что не очень удобно, если список понравившегося очень большой.

Окно профиля, с возможностью перехода к настройкам и т.п. в этих приложениях сильно отличаются своими функциями (Рисунок 3):

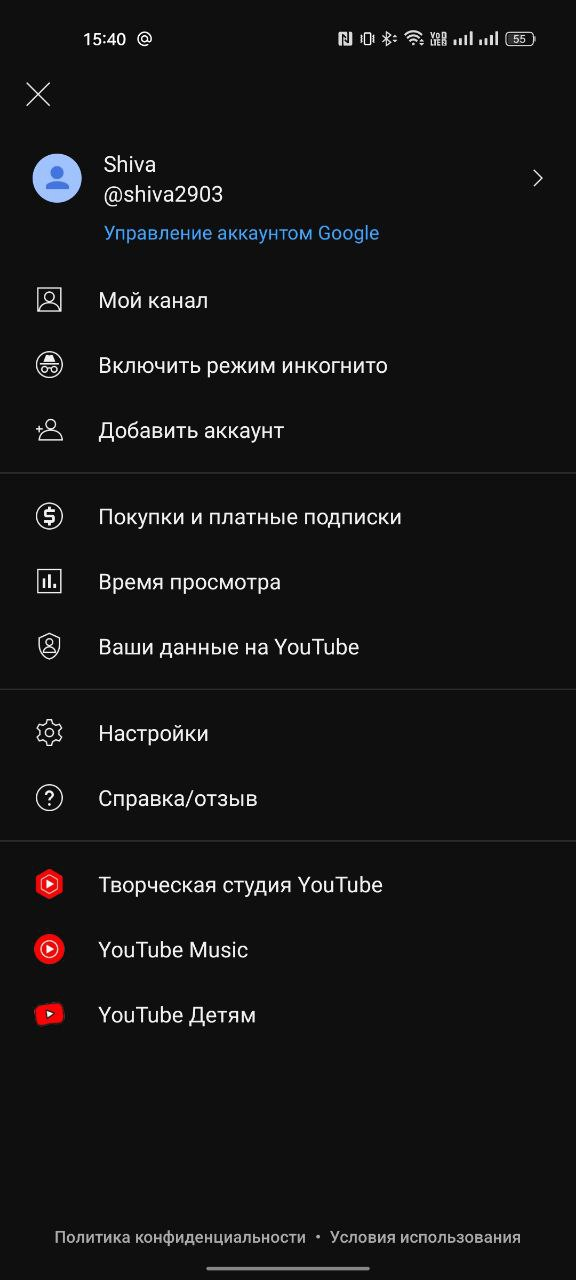
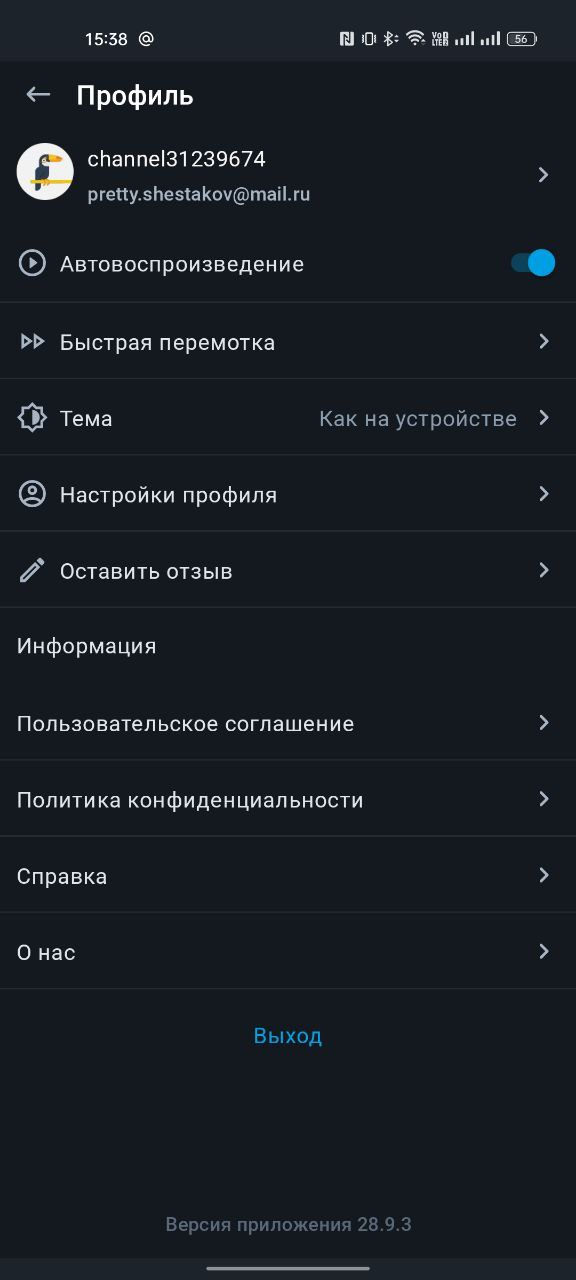


Рисунок 3 – Экран профиля пользователя

У обоих приложений свои страницы профиля, кардинально отличающиеся друг от друга. Из общего у них только два элемента:

* справка;
* переход на канал.

В приложении «YouTube» элементы профиля более сгруппированы. В приложении же «RUTUBE» нет разделения на группы, из-за чего ориентироваться труднее. Настройки, которые на «YouTube»вынесены в отдельную вкладку, в «RUTUBE»находятся прямо в самом окне профиля. Именно поэтому, экран профиля «YouTube», является более удобным для понимания и использования пользователем, чем «RUTUBE».

Сравнение по основным критериям данных приложений представлено в таблице 1.

Таблица 1 – Сравнение приложений

|  |  |  |
| --- | --- | --- |
| Параметры | YouTube | RUTUBE |
| Стоимость | Бесплатно | Бесплатно |
| Основной язык | Английский | Русский |
| Просмотр видеороликов | Есть | Есть |
| Возможность сохранять видеоролики в «Избранное» | Есть | Есть |
| Подробное описание к контенту | Есть | Есть |
| Наличие поиска | Есть | Есть |
| Выбор темы интерфейса | Есть | Есть |
| Список просмотренного | Есть | Есть |
| Оценки видеороликов | Есть | Есть |
| Возможность загружать свои видеоролики | Есть | Есть |
| Возможность продавать свой контент | Есть | Отсутствует |

Рассмотрев пару приложений, выполняющих похожие задачи, было решено написать приложение, которое имело бы простой интерфейс без лишних функций и с акцентом внимания на важных для пользователя элементах, в котором можно смотреть видеоролики и добавлять их в «Избранное». За основу будут взяты разные элементы двух приложений, так как каждое из них имеет как свои плюсы, так и минусы.

ИЛИ

Для разработки мобильного приложения «Mosticsmo» рассмотрим несколько аналогичных приложений для выполнения анализа и статистики финансовой деятельности.Существует большое количество приложений, осуществляющих финансовый учёт и анализ.Так как приложения получают свою популярность за счет своих функций, то при выборе прототипов стоит руководствоваться популярностью приложений.

Для начала рассмотрим первое попавшееся приложение. Интерфейс приложения «Финансы – бюджет & расходы»изображён на рисунке 1.1.

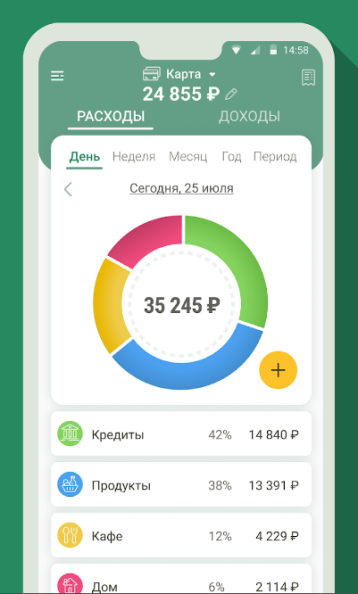


Рисунок 1.1 – интерфейс приложения «Финансы – бюджет & расходы»

Описание продукта: приложение финансы поможет взять под контроль свои финансы, бюджет и деньги, при этом не отнимет много времени. Вам больше не потребуется заглядывать в кошелек или заходить в свой банковский счет, чтобы узнать Ваше текущее финансовое состояние.

Далее рассмотрим приложение «1Money: расходы финансы бюджет» изображённого на рисунке 1.2.



Рисунок 1.2 – интерфейс приложения «1Money: расходы финансы бюджет»

Описание приложения: приложение 1Money покажет вам, на что, на самом деле, Вы тратите деньги. Зачастую, учет финансов занятие не из приятных и требует конкретных навыков и умений, вот почему мы разработали это финансовое приложение. Оно позволит вам легко вести учет расходов и доходов, составлять семейный бюджет, поможет быстрее рассчитаться с долгами, делать сбережения денег и достигать ваши финансовые цели. Ведите учет затрат по-новому!

После рассмотрим приложение «Учет расходов, семейный бюджет» предоставленного на рисунке 1.3.

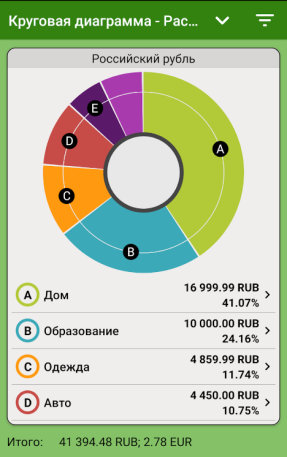


Рисунок 1.3 – интерфейс приложения «Учет расходов, семейный бюджет»

Рассмотрим описание приложения: Finance PM – ваш личный Финансовый менеджер. Основная цель приложения – позволить пользователям контролировать расходы и доходы легко и быстро.

Приложение поможет вам понять, сколько денег уходит на здоровье, на образование, каковы были траты на автомобиль. А также, отложить приличную сумму на отдых или жилье. Оно позволит контролировать расходы и сократить их.

Что пользователь получит, скачав данное приложение?

* внимательное отношение разработчика к пожеланиям и рекомендациям;
* удобный и функциональный интерфейс;
* надежную поддержку;
* управление любым количеством кошельков;
* возможность добавлять, редактировать и удалять операции по кошелькам: доходы, расходы, переводы;
* редактируемое дерево категорий, каждый пользователь настраивает его под себя;
* возможность добавлять новые валюты;
* возможность настраивать периодичные (повторяемые) операции;
* приложение подскажет: сколько денег вы должны кому-то или сколько должны. вам, это позволит избежать неловких ситуаций, когда случайно забываешь вернуть долг;
* защита приложения с помощью пароля, вход по отпечатку пальца;
* возможность синхронизации данных между несколькими устройствами, что необходимо, если вы хотите вести семейный бюджет;
* возможность создания операций по шаблонам;
* импорт и экспорт данных;
* экспорт данных в xls.

Далее рассмотрим последнее приложение «Monefy – Менеджер бюджета», изображенного на рисунке 1.4.

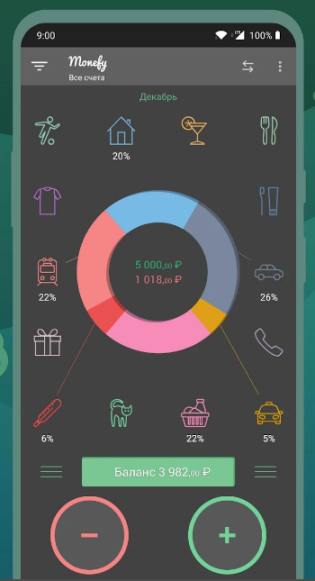


Рисунок 1.4 – интерфейс приложения «Monefy – Менеджер бюджета»

Рассмотрим описание приложения: Вы умеете планировать свой бюджет? С приложением Monefy вести финансы станет просто как никогда! Вы сможете вести учет доходов и расходов в пару кликов. Просто записывайте в приложении все, что покупаете: чашку эспрессо, счет за свет, кредит за авто или покупки из супермаркета. Забудьте про утомительное заполнение форм, вам нужно только зафиксировать сумму. Ваш кошелек от этого только выиграет, ведь экономить куда проще, если можно наглядно посмотреть, на что вы тратите деньги. Monefy – это домашняя бухгалтерия у вас в кармане.

Исходя из рассмотрения аналоговых приложенийможно сделать вывод, что положительными качествами разрабатываемого приложениядолжны быть:

* оснащенность необходимыми функциями;
* интуитивно понятный интерфейс;
* правильная цветовая гамма.

Проанализировав данные приложения и выделив положительные и отрицательные качества, можно сделать вывод, что при разработкепланируется использовать положительные качества.

# Проектирование приложения

## UI/UX дизайн приложения

Дизайн проекта разработан в программе Figma.

Для проекта были определены основные экраны:

* регистрация;
* авторизация;
* главный экран с видеороликами;
* личный кабинет;
* понравившиесявидеоролики;
* загрузка нового видеоролика;
* загруженные видеоролики;
* просмотр видеоролика.

Для тем приложения определены следующие две цветовые схемы. Первая тема реализуется в тёмных тонах, поэтомув ней основными цветамиявляются: серый, черный, белый (Рисунок 4).

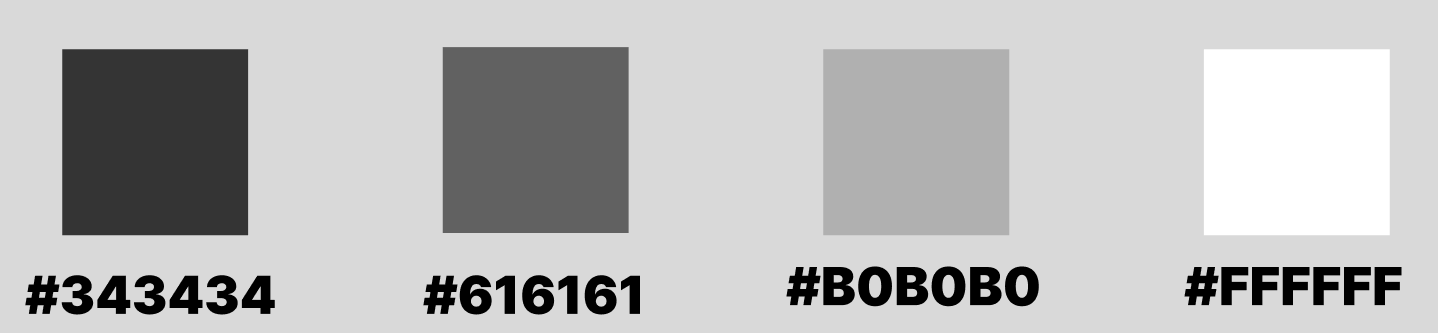


Рисунок 4 – Тёмная цветовая схема

Вторая тема будет светлого цвета: белый и светло-серый (Рисунок 5).



Рисунок 5 – Светлая цветовая схема

Но стоит заметить, что данные цвета, разбавленные изображениями, не выглядят скучно, а весьма лаконично. Необходимость разделения цветовых схем на тёмную и светлую, заключается в том, что пользователю может быть удобнее использовать какую-либо из них в разное время суток.

Логотип приложения общий как для тёмной темы, так и для светлой. Он делится на две половины разных цветов, на которых написано название приложения «VideoHost».

Ниже на рисунке 7 представлен логотип видеохостинга.



Рисунок 6 – Логотип видеохостинга

Цветовая схема логотипа состоит из: красного и белого (Рисунок 7). Это броские цвета, которые могут привлечь внимание пользователя к приложению.

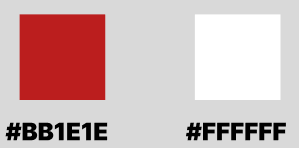


Рисунок 7 – Цветовая схема логотипа

Определившись с цветовой схемой приложения и создав его логотип, был разработан дизайн следующих экранов:

* авторизация (Login);
* регистрация(Registration)
* главный экран (Main)
* профиль (Profile)
* понравившиеся видеоролики (Like)
* загрузка видеороликов (UploadVideo)
* загруженные видеоролики (DoneUpload)
* просмотра видеороликов (Play)

Ниже на рисунке 8 представлен дизайн приложения с тёмной цветовой схемой.





Рисунок 8 – Дизайн приложения с тёмной темой

Ниже на рисунке 9 представлен дизайн приложения со светлой цветовой схемой.





Рисунок 9 – Дизайн приложения со светлой темой

На рисунке 10 показано перемещение пользователя в приложении.

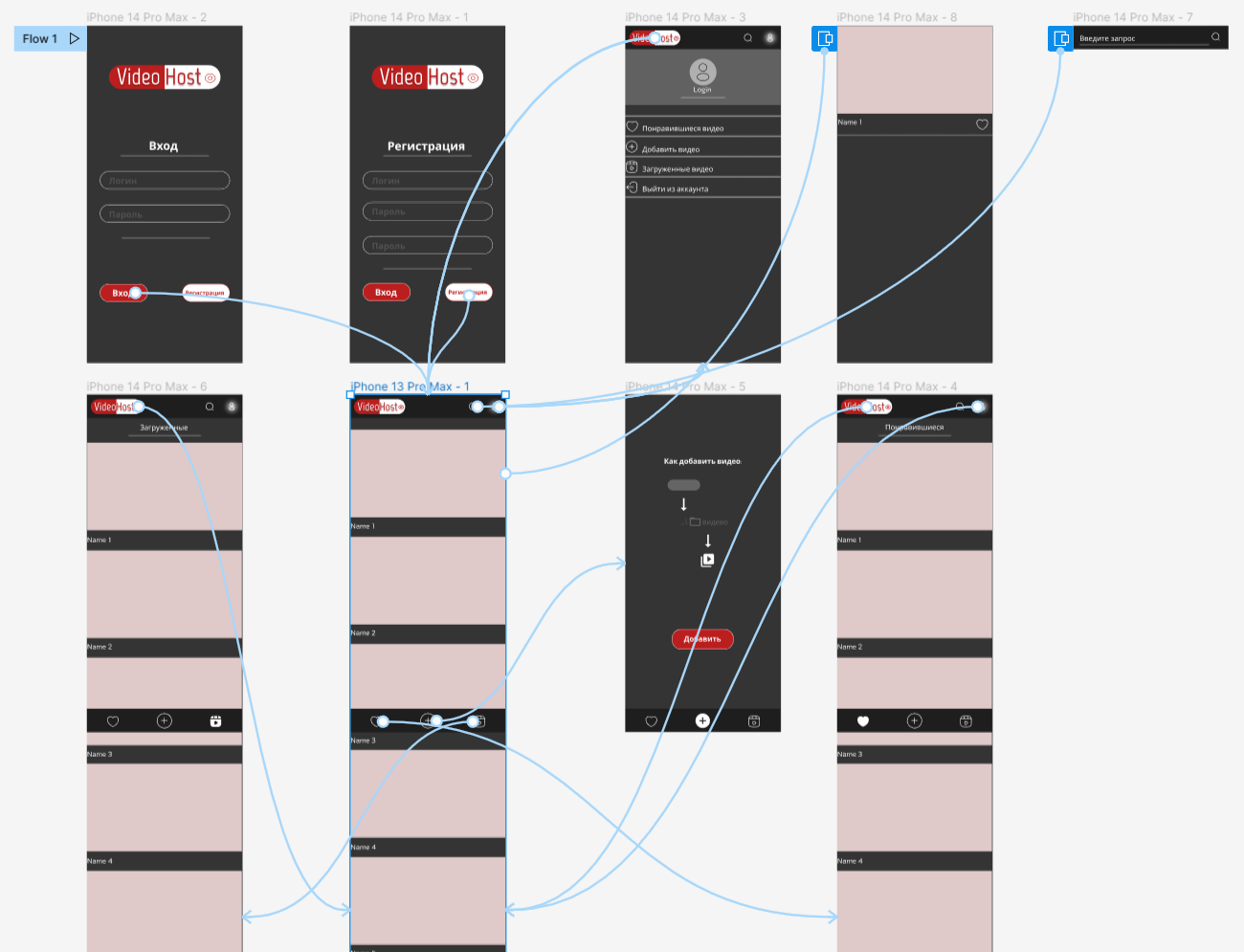


Рисунок 10 – Перемещение пользователя в приложении

При первом запуске приложения пользователь попадает на экран авторизации. Если у него уже есть аккаунт, то он может ввести свои данные и нажать на кнопку «Вход», после чего он попадёт на главный экран. В случае, если у него нет аккаунта, он может нажать на кнопку «Регистрация», которая откроет экран регистрации, где он сможет создать новый аккаунт. С каждого экрана можно перейти в любую часть приложения, однако на рисунке 7 обозначены только действия при нажатии кнопок, это сделано для того, чтобы не нагружать рисунок одинаковыми действиями. При нажатии на кнопку «Понравившееся» пользователь перейдёт на экран со списком видеороликов, которые ему понравились. Нажимая на кнопку «Профиль» откроется экран профиля, на котором пользователь может выбрать другие окна приложения. При нажатии на область превью видеоролика, пользователь начнет его просмотр. Если пользователь нажмёт на кнопку «Загрузить видеоролик», то он перейдёт на окно с его загрузкой. А при нажатии на кнопку «Загруженные видео», откроется список видео, которые загрузил сам пользователь.

## Выбор технологии, языка и среды программирования

Средой программирования выбрана программа AndroidStudio, так как она довольно удобна для создания приложения и является довольно популярной, поэтому в случае возникновения ошибки, легче найти способ её решения. Языком программирования является Kotlin, так как это новый и развивающийся язык программирования, к которому очень часто добавляются новые удобные функции, которые пригодятся во время разработки.

Используемый сервис для базы данных и аутентификации – Firebase. Это бесплатная база данных, которая подходит под поставленную задачу. Она довольно быстрая и легка для понимания. Так же в ней представлены все необходимые функции: Authentication(для регистрации и авторизации пользователей), Realtime Database (для синхронизации данных), Storage (для хранения файлов в системе).

# Разработка мобильного приложения

## Разработка базы данных

В качестве разрабатываемой базы данных выбрана облачная база данных Firebase, интегрируемая в AndroidStudio. В ней данные хранятся в формате JSON, то есть, она является NoSQL – базой данных. Пример хранения данных в базе представлен на рисунке 11.



Рисунок 11 – Страница Firebase с информацией о видео

База данных организована в виде древовидной структуры, где каждое видео представлено уникальным идентификатором (ID), который служит вершиной дерева. Каждому ID соответствует набор информации о конкретном видеоролике.

В базе данных хранится вся информация о видеоролике: название, время загрузки и ссылка на него.

## Разработка мультимедийного контента

Весь мультимедийный контент разрабатывался с помощью языка разметки XML. Вёрстка выполнялась по дизайну, разработанному ранее в приложении Figma. Однако, по мере разработки, в дизайн были внесены изменения.

Иконки, кнопки и другие элементы приложения, были импортированы в проект в виде XML-файлов (Рисунок 12). Такой способ хранения уменьшает вес приложения, а также, избавляет от проблем с потерей качества мультимедийного контента. Все ресурсы хранятся в папке «drawable»

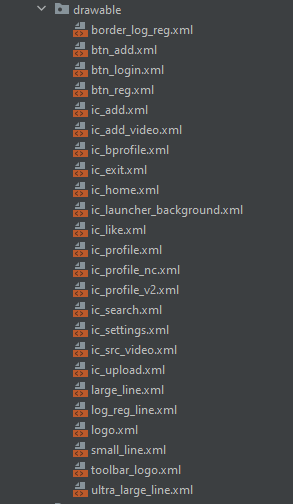


Рисунок 12 – Мультимедийный контент

Так как в приложении используется две цветовые схемы, то необходимо хранить обе палитры в файле colors.xml

Эти цвета используются для описания цветов темы приложения. Темы приложений описываются в двух файлах: themes.xml–для светлой темы, а themes.xml (night) – для тёмной темы.

То, какая тема будет использоваться, пользователь решает сам.  
Кому-то может быть неприятно использовать какую-то определенную тему, поэтому выбор между тёмной и светлой темой – достойная черта приложения. Именно поэтому, в приложение есть выбор, какую тему использовать.

## Описание используемых плагинов

В проекте используются библиотеки с различными компонентами и функциями. Список всех библиотек, а также их описание представлен в таблице 2.

Таблица 2 – Библиотеки и их описание

|  |  |
| --- | --- |
| Библиотека | Описание |
| com.google.firebase:firebase-bom:31.3.0 | Обеспечивает согласованную версию всех Firebase-библиотек |

Продолжение таблицы 2

|  |  |
| --- | --- |
| com.google.firebase:firebase-auth-ktx | Библиотека для аутентификации пользователей через различные методы (электронная почта, социальные сети и др.). |
| com.google.firebase:firebase-  analytics-ktx | Библиотека для сбора и анализа аналитических данных о приложении. |
| androidx.core:core-splashscreen:1.0.1 | Библиотека для создания и управления экраном загрузки (splashscreen) в приложении. |
| androidx.appcompat:appcompat:1.6.1 | Библиотека, обеспечивающая обратную совместимость с новыми функциями и возможностями Android для более старых версий платформы. |
| com.google.android.material:  material:1.9.0 | Библиотека, содержащая MaterialDesign компоненты пользовательского интерфейса. |
| androidx.legacy:legacy-support-  v4:1.0.0 | Библиотека, обеспечивающая обратную совместимость с более старыми версиями Android и предоставляющая поддержку различных функций и компонентов, которые были введены в более новых версиях платформы. |
| androidx.recyclerview:  recyclerview:1.3.0 | Библиотека, предоставляющая RecyclerView - мощный компонент пользовательского интерфейса, который позволяет эффективно отображать списки данных, обеспечивая повторное использование представлений и поддержку различных макетов. |
| androidx.annotation:  annotation:1.6.0 | Библиотека, содержащая аннотации, используемые во время компиляции для предоставления информации о коде, например, для обозначения намерений, проверок на статический анализ или документации. |
| com.google.firebase:firebase-database-ktx:20.2.1 | Библиотека для работы с FirebaseRealtimeDatabase, которая предоставляет возможность хранения и синхронизации данных в реальном времени на облачной платформе Firebase. |

Окончание таблицы 2

|  |  |
| --- | --- |
| com.google.firebase:firebase-storage-ktx:20.2.0 | Библиотека для работы с FirebaseStorage, которая предоставляет возможность хранения и управления медиафайлами (например, изображениями и видео) на облачной платформе Firebase. |
| junit:junit:4.13.2 | Библиотека для написания и запуска модульных тестов в Java. |
| androidx.test.ext:junit:1.1.5 | Библиотека, расширяющая возможности JUnit для тестирования Android-приложений. |
| androidx.test.espresso:  espresso-core:3.5.1 | Библиотека для автоматизированного UI-тестирования Android-приложений с использованием Espresso фреймворка. |
| androidx.navigation:  navigation-fragment-ktx:2.5.3,  androidx.navigation:  navigation-ui-ktx:2.5.3 | Библиотеки, предоставляющие навигационные компоненты для управления переходами и навигацией между фрагментами и активностями в приложении. |
| androidx.lifecycle:  lifecycle-livedata-ktx:2.6.1,  androidx.lifecycle:  lifecycle-viewmodel-ktx:2.6.1 | Библиотеки, обеспечивающие жизненный цикл компонентов Android (таких как активности и фрагменты) и предоставляющие LiveData и ViewModel для удобной работы с данными в приложении. |

Данные библиотеки являются неотъемлемой частью приложения, обеспечивая его правильное функционирование и реализацию всех задуманных возможностей. Без этих библиотек приложение не сможет работать в полной мере, так как они предоставляют необходимые инструменты и функции, которые необходимы для его работы.

## Описание разработанных процедур и функций

В приложении реализованы следующие методы(таблица 3):

Таблица 3 – Методы приложения

|  |  |
| --- | --- |
| Метод | Описание |
| uploadVideoFirebase() | Загрузка видеоролика на сервер по средствам запросов |
| videoPickDialog() | Выбор метода загрузки приложения |
| requestCameraPermission() | Запрос разрешения на использование камеры |
| checkCameraPermissions() | Проверка разрешения на использование камеры |
| videoPickGallery() | Выбор видео из галереи |
| videoPickCamera() | Запись видео с помощью камеры |
| replaceFragment() | Размещение определенного фрагмента на главном экране |
| onBindViewHolder() | Получение информации о видео для его вывода на экране |
| setVideoUrl() | Получение ссылки на видео |
| toggleFavorite() | Добавление видео в избранное |

Так же в приложении есть классы, которые не имеют методов (таблица 4):

Таблица 4 – Классы приложения

|  |  |
| --- | --- |
| Класс | Описание |
| RegisterActivity | Регистрация пользователя в системе |
| ProfileActivity | Окно профиля с выводом токена пользователя |
| LoginActivity | Авторизация в приложение |

Авторизация в приложении реализована по средству запросов в базу данных.

В случае, если пользователю нужно зарегистрироваться, сперва нужно перейти на экран регистрации.

При регистрации, в базу данных запросами заносятся новые данные пользователя.

При переходе на главный экран, пользователю выводятся все видеоролики, загруженные в базу данных. Они выводятся с помощью компонента для отображения элементов списка RecyclerView.

Листинг 6 – Вывод видеороликов

private fun loadVideosFromFirebase() {  
 videoArrayList = ArrayList()  
 // Получаем ссылку на базу данных Firebase  
 val ref = FirebaseDatabase.getInstance("https://videohost-project-default-rtdb.europe-west1.firebasedatabase.app").getReference("Videos")  
 ref.addValueEventListener(object : ValueEventListener{  
  
 override fun onDataChange(snapshot: DataSnapshot) {  
 // Очищаем список видео перед загрузкой новых данных  
 videoArrayList.clear()  
 // Проходим по всем дочерним элементам снимка данных  
 for (ds in snapshot.*children*){  
 // Получаем объект видео из снимка данных  
 val modelVideo = ds.getValue(ModelVideo::class.*java*)  
 // Добавляем видео в список  
 videoArrayList.add(modelVideo!!)  
 }  
 // Сортируем список видео в порядке убывания по временной метке  
 videoArrayList.*sortByDescending* **{ it**.timestamp **}**// Создаём адаптер для отображения видео  
 adapterVideo = AdapterVideo(requireContext(), videoArrayList, firebaseAuth)  
 val videosRv: RecyclerView = *view*!!.findViewById(R.id.*videosRv*)  
 videosRv.*adapter* = adapterVideo  
 }  
 // В случае ошибки при получении данных из базы данных Firebase, выводим сообщение об ошибке  
 override fun onCancelled(error: DatabaseError) {  
 Toast.makeText(*context*, "Database error: " + error.*message*, Toast.*LENGTH\_SHORT*).show()  
 }  
 })  
}

В случае, если пользователь решит добавить видео в избранное, то с помощью запроса в базе данных создастся новая запись.

Добавленные в избранное видеоролики, выводятся на другом экране, по средствам двух запросов в базу данных.

При добавлении видеоролика пользователь должен дать разрешение на использование камеры или галереи.Нижепредставленкодзапросаправ.

//Запросправнакамеру  
privatefunrequestCameraPermission(){  
ActivityCompat.requestPermissions(  
this,  
cameraPermission,  
CAMERA\_REQUEST\_CODE  
 )  
}  
  
privatefuncheckCameraPermissions():Boolean{  
 //Проверяетналичиеправ  
valresult1 = ContextCompat.checkSelfPermission(  
this,  
android.Manifest.permission.*CAMERA*) == PackageManager.*PERMISSION\_GRANTED*returnresult1  
}

// Переопределениеметодадляобработкирезультатовзапросаразрешений  
override fun onRequestPermissionsResult(  
 requestCode: Int,  
 permissions: Array<out String>,  
 grantResults: IntArray  
) {  
 when(requestCode){  
 CAMERA\_REQUEST\_CODE ->  
 if (grantResults.*isNotEmpty*()) {  
 val cameraAccepted = grantResults[0] == PackageManager.*PERMISSION\_GRANTED*if (cameraAccepted) { // Разрешениенаиспользованиекамерыполучено  
 //Приняли  
 videoPickCamera()  
 } else{  
 Toast.makeText(this, "Праваподтверждены", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 }  
  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults)  
}

После получения прав доступа пользователь может загрузить видеоролик.

# Тестирование

## Протокол тестирования дизайна приложения

Тестирование дизайна приложения проводится на самом минимальном (AndroidSDK 24) и на более позднем (AndroidSDK 33) с различной диагональю экранов для проверки разметки страниц и вёрстки приложения.

Примеры проверок отображения элементов на экране представлены на рисунках 13–17.

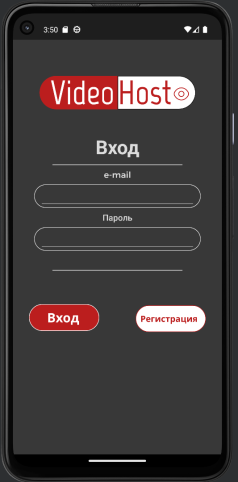


Рисунок 13 – Экраны авторизации

Элементы интерфейса в обоих случаях отображаются корректно.

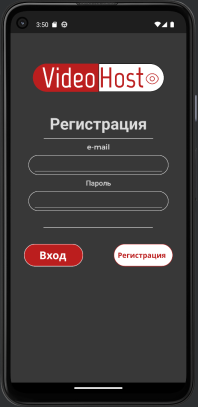


Рисунок 14 – Экраны авторизации

Экран авторизации так же отображается корректнона обоих устройствах, все элементы интерфейса расположены на своих местах.

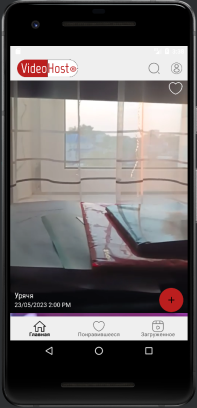


Рисунок 15 – Главные экраны

Главный экран тоже отображается корректно. Верхняя и нижняя панель нормальных размеров, элементы на них не сдвинуты. Логотип так же находится на своём месте. Кнопки избранного привязаны к видеопроигрывателю.

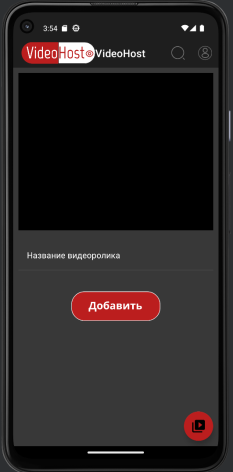
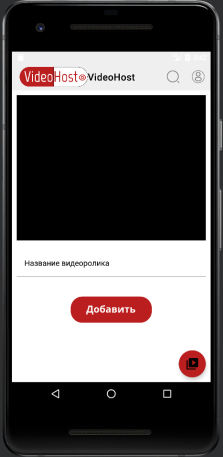


Рисунок 16 – Экран добавления видеоролика

На экране добавления видеоролика так же не обнаружено ошибок. Масштаб элементов сохранен. Верхняя панель находится на своём месте и имеет нормальные размеры.

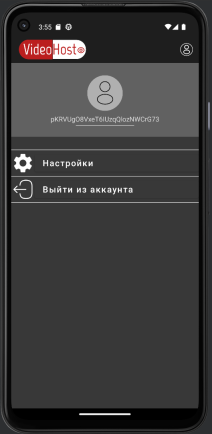
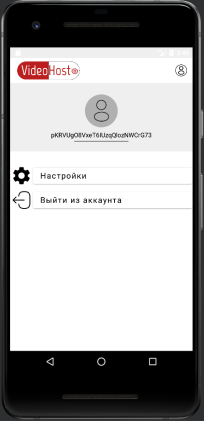


Рисунок 17 – Экраны профиля

Элементы экрана профиля отображаются корректно. Фотография пользователя не поменяла свои масштабы. Элементы меню профиля находятся на своих местах. Текст не растянут и не выходит за пределы экрана.

ИЛИ

Для проведения тестирования дизайна приложение было выбрано два устройства с разным разрешением экрана.

На рисунке 15 отображены экраны меню. Элементы меню отображаются корректно на обоих устройствах.

Изображение выглядит как текст, письмо

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рисунок 15– экраны меню

Ниже на рисунке 16 отображен дизайн выбора уровня на устройствах с разным разрешением. Элементы отображаются корректно, кнопки не съезжают, элементов главного экрана меню не видно при большом разрешении.



Рисунок 16 – экран выбора уровня

На рисунке 17 отображены настройки игры, в обоих случаях видны все элементы и текст выходит за пределы экрана.



Рисунок 17 – окно выбора уровня

На рисунке 18 отображен экран уровня. Таймер отображается корректно в обоих случаях.

Изображение выглядит как диаграмма

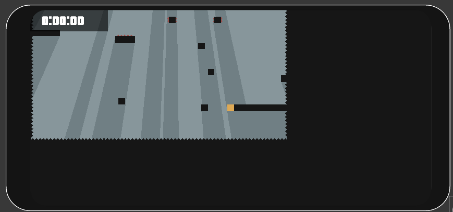
Автоматически созданное описание

Рисунок 18 – экран уровня

Здесь должен быть небольшое заключение по UIтестированию

## Протокол тестирования функционала приложения

Так же необходимо проверить функционал приложения. Для этого, для каждой функции были разработаны TestCase.

В таблице 5 представлено тестирование функции вывода видеоролика на экран.

Таблица 5 – Тестирование функции вывода видео на экран

|  |  |  |
| --- | --- | --- |
| **Название**: | VideoHost | |
| **Функция**: | Вывод видео на экран | |
| **Действие** | **Ожидаемыйрезультат** | **Результаттеста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  |
|  |  |  |
| **Шагитеста (positive)**: |  | |
| Дождаться загрузки главного экрана | Окно для вывода видеороликов открылось | пройден |

В таблице 6 представлено тестирование функции добавления видео в «Понравившиеся».

Таблица 6 – Тестирование функции добавление видео в «Понравившиеся».

|  |  |  |
| --- | --- | --- |
| **Название**: | VideoHost | |
| **Функция**: | Добавление видео в «Понравившиеся» | |
| **Действие** | **Ожидаемыйрезультат** | **Результаттеста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  |
|  |  |  |
| **Шагитеста (positive)**: |  | |
| Дождаться загрузки главного экрана | Окно для вывода видеороликов открылось | пройден |
| Нажать на кнопку добавления видео в избранное | В базу данных отправлен запрос на добавление видео в избранное | пройден |
| Перейти на вкладку «Понравившиеся» | Добавленный ролик отобразится в списке | пройден |

В таблице 7 представлено тестирование функции загрузки видео в базу данных.

Таблица 7 – Тестирование функции загрузки видео

|  |  |  |
| --- | --- | --- |
| **Название**: | VideoHost | |
| **Функция**: | Загрузка видео | |
| **Действие** | **Ожидаемыйрезультат** | **Результаттеста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  |
|  |  |  |
| **Шагитеста (positive)**: |  | |
| Перейти на вкладку добавления видеоролика | Окно корректно загрузилось | пройден |
| Выбрать видеоролик из галереи или снять новый на камеру | Приложение получило файл видеоролика и информацию о нём | пройден |
| Ввести название видеоролика | Приложение получило название видеоролика | пройден |
| Нажать на кнопку «Загрузить» | Файл видеоролика отправлен в базу данных | пройден |
|  |  |  |
|  |  |  |
| **Шагитеста (negative)**: |  |  |
| Перейти на вкладку добавления видеоролика | Окно корректно загрузилось | пройден |
| Не выбирать видеоролик | Приложение получит код ошибки | пройден |
| Не вводить название видео | Приложение получит код ошибки | пройден |
| Нажать на кнопку «Загрузить» | Появится всплывающее сообщение с определенной ошибкой | пройден |

В таблице 8 представлено тестирование функции регистрации нового пользователя.

Таблица 8– Тестирование функции регистрации пользователя

|  |  |  |
| --- | --- | --- |
| **Название**: | VideoHost | |
| **Функция**: | Регистрация нового пользователя | |
| **Действие** | **Ожидаемыйрезультат** | **Результаттеста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «VideoHost» | Приложение открылось корректно |  |
|  |  |  |
| **Шагитеста (positive)**: |  | |
| Перейти на вкладку добавления регистрации | Окно корректно загрузилось | пройден |
| Заполнить обязательные поля корректными данными | Успешное заполнение обязательных полей | пройден |
| Нажать на кнопку «Регистрация» | Приложение проверило введенные данные и зарегистрировало пользователя | пройден |
| **Шагитеста (negative)**: |  |  |
| Перейти на вкладку добавления регистрации | Окно корректно загрузилось | пройден |
| Не заполнять обязательные поля или ввести некорректные данные | Успешное заполнение обязательных полей | пройден |
| Нажать на кнопку «Регистрация» | Появится всплывающее сообщение с определенной ошибкой | пройден |

В таблице 9 представлено тестирование функции авторизации пользователя.

Таблица 9– Тестирование функции авторизации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название**: | | VideoHost | | |
| **Функция**: | | Авторизация | | |
| **Действие** | **Ожидаемыйрезультат** | | | **Результаттеста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | | | |
| Запустить приложение «VideoHost» | Приложение открылось корректно | |  | |
|  |  | |  | |
| **Шагитеста (positive)**: |  | | | |
| Заполнить обязательные поля корректными данными | Успешное заполнение обязательных полей | | пройден | |
| Нажать на кнопку «Войти» | Приложение проверило введенные данные и авторизировало пользователя | | пройден | |
|  |  | |  | |
|  |  | |  | |
| **Шагитеста (negative)**: |  | |  | |
| Не заполнять обязательные поля или ввести некорректные данные | Успешное заполнение обязательных полей | | пройден | |
| Нажать на кнопку «Войти» | Появится всплывающее сообщение с определенной ошибкой | | пройден | |

Разработанные TestCaseдемонстрируют корректную работу функций приложений. Так же в процессе тестирования не выявлено ошибок в дизайне приложения и его логике. Каждый экран прошёл проверку на разных API на корректное отображение элементов соответствующих экранов.

Заключение

В ходе разработки мобильного приложения для видеохостингаучтены предпочтения и потребности пользователей, в формате коротких видеороликов. Функция ведения списка просмотренного контента успешно реализована, позволяя пользователям отслеживать свои предпочтения и легко вернуться к уже просмотренному контенту.

Интерфейс приложения разработан с учетом удобства использования и минимального количества действий для достижения пользовательской цели. Основные экраны спроектированы таким образом, чтобы не перегружать пользователей лишней информацией, и добавлены полезные функции, такие как просмотр трейлеров и возможность создания собственных списков просмотренного и избранного контента.

Важным элементом концепции приложения является возможность просмотра коротких видеороликов, которая позволяет пользователям сократить время в дороге на работу, учебу или в перерывах между делами. Это удобный способ получать информацию на ходу.

Поддержка списка избранного контента также оказалась значимой для пользователей, предоставляя удобный способ отслеживания и доступа к контенту, который им особенно нравится.

Важным этапом в разработке приложения было тестирование дизайна, которое подтвердило его корректность и успешное функционирование. Это гарантирует, что разработанное приложение соответствует поставленным целям и требованиям.

В целом разработка мобильной версии видеохостинга позволила создать функциональное и удобное приложение, учитывающее потребности пользователей. Проделанная работа успешно достигла поставленной цели курсового проекта и является основой для дальнейшего развития и улучшения приложения.

Библиография

Нормативно-правовые акты:

1. ГОСТ Р 2.105-2019. ЕСКД. Общие требования к текстовым документам. – Москва: Стандартинформ, 2019. – 36 с

Электронные ресурсы:

1. AndroidDevelopers [Электронный ресурс]. – Документация AndroidStudio. – URL: https://developer.android.com/docs/ (дата обращения: 25.11.2022)
2. JavaHTMLParser [Электронный ресурс]. – Вытягивание данных с страницы. – URL: https://jsoup.org/ (дата обращения: 10.03.2023)
3. FirebaseDocuments [Электронный ресурс]. – Документация Firebase. Вытягивание данных с базы данных – URL: https://firebase.google.com/support (дата обращения: 25.02.2023)
4. GeeksForGeeks [Электронный ресурс]. – Как добавить круглые графики в ваше android приложение? – URL: https://www.geeksforgeeks.org/how-to-add-a-pie-chart-into-an-android-application(дата обращения: 20.04.2022)
5. GeeksForGeeks [Электронный ресурс]. – Как добавить отпечаток пальца в ваше android приложение? – URL: https://www.geeksforgeeks.org/how-to-add-fingerprint-authentication-in-your-android-app/ (дата обращения: 20.12.2022)
6. Material Design [Электронныйресурс]. – Bottom Navigation. – URL: https://material.io/components/bottom-navigation/android (дата обращения: 25.11.2022)
7. Figma[Электронный ресурс]. – Прототипирование проектов. – URL: https://www.figma.com/ (дата обращения: 25.11.2022)

Можно использовать и такие источники:

1 Yasno [Электронный ресурс]: Мобильный интерфейс: разработка дизайна приложений и проектирование макетов на примерах – Режим доступа к руководству: <https://yasno.mobi/blog/mobilnyy-interfeys-razrabotka-dizayna-prilozheniy-i-proektirovanie-maketov-na-primerakh/>

2 DevelopersAndroid [Электронный ресурс]: Полная документация по Android–Режим доступа к руководству: <https://developer.android.com/>

3 METANIT [Электронный ресурс]: Документация по языку программирования Kotlin – Режим доступа к руководству: <https://metanit.com/kotlin/>

4 Webkyrs [Электронный ресурс]: Создание базы данных Firebase – Режим доступа к руководству: <https://webkyrs.info/page/kak-sozdat-bazu-dannykh-firebase-realtime-database>

5 Epochta [Электронный ресурс]: как составить портрет клиента – Режим доступа: <https://www.epochta.ru/blog/portret-klientov>

**Приложение А**

(обязательное)

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

РАЗРАБОТКА Мобильной версии видеохостинга

Техническое задание

НАТКиГ.202000.010.000ПЗ

Выполнил:

Студент группыПР-21.106

Шестаков И.В.

2023

**Содержание**

Введение 43

1 Назначения разработки 44

2 Требования к мобильному приложению 45

2.1 Требования к функциональным характеристикам 45

2.2 Требования к надёжности 45

2.3 Условия эксплуатации 45

2.4 Требования к составу и параметрам технических средств 45

2.5 Требования к информационной и программной совместимости 46

2.6 Требования к защите информации 46

2.7 Требования к маркировке и упаковке 46

3 Требования к программной документации 47

4 Технико-экономические показатели 48

5 Стадии и этапы разработки 49

6 Порядок контроля и приёмки 50

**Введение**

Настоящее техническое задание распространяется на разработку мобильного приложения «Разработка мобильной версии видеохостинга», используемого для просмотра видеороликов, ведения списков загруженных видеороликов и понравившегося.

Наименование приложения: «VideoHost».

Краткая характеристика области применения: мобильное приложение предоставляет возможность просмотра видеороликов, загруженных на видеохостинг, а также ведения списка загруженных видеороликов самого пользователя и наиболее понравившегося ему контента.

Основанием для проведения разработки является Протокол №6 от 21 февраля 2022 года.

Наименование темы разработки − «Разработка мобильной версии видеохостинга».

Условное обозначение темы разработки – «VideoHost».

**1 Назначение разработки**

Основное назначение приложения заключается в:

* обеспечении удобного и понятного интерфейса приложения для пользователя;
* предоставление большой библиотеки видеороликов.

Лица, которые могут работать с данной системой:

администратор – управляет полностью всем приложением, следит за его работоспособностью, обновляет информацию в базе данных приложения;

пользователь приложения – может просматривать видеоролики загруженные на видеохостинг, а также вести списки понравившихся и загруженных им видеороликов.

**2 Требования к мобильному приложению**

**2.1 Требования к функциональным характеристикам**

Требования к составу выполняемых функций:

* просмотр видеороликов;
* просмотр списка загруженных пользователем видеороликов;
* просмотр списка понравившихся видеороликов;

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация стабильного интернет-соединения.

Приложение должно контролировать входную информацию:

* соблюдение типов данных при заполнении полей.

**2.3 Условия эксплуатации**

Пользователь должен иметь практические навыки использования мобильного устройства под управлением операционной системы Android.

**2.4 Требования к составу и параметрам технических средств**

Для работы приложения необходимо мобильное устройство с установленной операционной системой Android не ниже версии 7.0.

**2.5 Требования к информационной и программной совместимости**

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования Java.

**2.6 Требования к защите информации**

Доступ к информации БД предоставляется только администратору базы данных.

**2.7** **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

**3 Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* пояснительная записка.

**4 Технико-экономические показатели**

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

**5 Стадии и этапы разработки**

Таблица 1 – Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Этапы разработки КП** | **Сроки выполнения** | **Отчётность** |
| 1 | Определение цели и задач, объекта и предмета исследования | 13.03.2023 | Пояснительная записка |
| 2 | Описание предметной области | 08.04.2023 | Пояснительная записка |
| 3 | Выбор технологии, языка и средыпрограммирования | 10.04.2023 | Пояснительная записка |
| 4 | Оформление технического задания | 15.04.2023 | Техническое задание |
| 5 | Проектирование UI/UXдизайна | 17.04.2023 | Спецификации программного обеспечения |
| 6 | Разработка мобильного приложения | 22.04.2023 | Схема структурная системы и спецификации компонентов |
| 7 | Разработка базы данных | 02.05.2023 | Программный  продукт |
| 8 | Отладка и тестирование приложения | 13.05.2023 | Тексты программных компонентов |
| 9 | Оформление документации | 22.05.2023 | Программная  документация |
| 10 | Защита | 27.05.2023 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

* техническое задание;
* пояснительная записка;
* программный продукт;

# Приложение Б

Метод для получения актуальных цен валют

private void GetWeb(){  
try {  
doc = Jsoup.*connect*("https://novosibirsk.bankiros.ru/currency").get();  
Elements tables = doc.getElementsByTag("tbody");  
*//Выбортаблицы*Element our\_table = tables.get(0);  
Elements elements\_from\_tavble = our\_table.children();  
*//Выборэлементастроки*for(int i = 2; i <9; i++) {  
Element valut = elements\_from\_tavble.get(i);  
Elements valut\_element = valut.children();  
String valutname = valut\_element.get(0).text();  
Element coastbuy = valut\_element.get(1);  
Elements coastbuy\_element = coastbuy.children();  
Element coastsell = valut\_element.get(2);  
Elements coastsell\_element = coastsell.children();  
Element sell = coastsell\_element.get(0);  
Elements co = sell.children();  
Element buy = coastbuy\_element.get(0);  
Elements cos = buy.children();  
Pattern decimalNumPattern = Pattern.*compile*("[0-9]+(.[0-9]+)?");  
Matcher matcher;  
switch (valutname){  
case "USD":  
 matcher = decimalNumPattern.matcher(coastbuy\_element.get(0).text());  
while (matcher.find()) {  
usbuy = cos.get(0).text();  
 }  
 matcher = decimalNumPattern.matcher(coastsell\_element.get(0).text());  
while (matcher.find()) {  
ussell = co.get(0).text();  
 }  
ussell = coastsell\_element.get(0).text().substring(0,5);  
break;  
case "EUR":  
 matcher = decimalNumPattern.matcher(coastbuy\_element.get(0).text());  
while (matcher.find()) {  
erbuy = cos.get(0).text();  
 }  
 matcher = decimalNumPattern.matcher(coastsell\_element.get(0).text());  
while (matcher.find()) {  
ersell = co.get(0).text();  
 }  
break;  
case "CNY":  
 matcher = decimalNumPattern.matcher(coastbuy\_element.get(0).text());  
while (matcher.find()) {  
cnbuy = cos.get(0).text();  
 }  
 matcher = decimalNumPattern.matcher(coastsell\_element.get(0).text());  
while (matcher.find()) {  
cnsell = co.get(0).text();  
 }  
break;  
case "BYN":  
 matcher = decimalNumPattern.matcher(coastbuy\_element.get(0).text());  
while (matcher.find()) {  
bybuy = cos.get(0).text();  
 }  
 matcher = decimalNumPattern.matcher(coastsell\_element.get(0).text());  
while (matcher.find()) {  
bysell = co.get(0).text();  
 }  
break;  
case "GBP":  
 matcher = decimalNumPattern.matcher(coastbuy\_element.get(0).text());  
while (matcher.find()) {  
gbbuy = cos.get(0).text();  
 }  
 matcher = decimalNumPattern.matcher(coastsell\_element.get(0).text());  
while (matcher.find()) {  
gbsell = co.get(0).text();  
 }  
break;  
case "CHF":  
 matcher = decimalNumPattern.matcher(coastbuy\_element.get(0).text());  
while (matcher.find()) {  
chbuy = cos.get(0).text();  
 }  
 matcher = decimalNumPattern.matcher(coastsell\_element.get(0).text());  
while (matcher.find()) {  
chsell = co.get(0).text();  
 }  
break;  
case "AED":  
 matcher = decimalNumPattern.matcher(coastbuy\_element.get(0).text());  
while (matcher.find()) {  
aebuy = cos.get(0).text();  
 }  
 matcher = decimalNumPattern.matcher(coastsell\_element.get(0).text());  
while (matcher.find()) {  
aesell = co.get(0).text();  
 }  
break;  
 }  
 }  
 }catch(IOException exception){  
 exception.printStackTrace();  
 }