Министерство образования Новосибирской области

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

РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ ТИПА PAINT

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения  
для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.200700.010.000ПЗ

Разработал:

студент группы ПР-21.101

Ефименко Е.С.

Проверил:

Санников И.М.

**Содержание**

[Введение 3](#_Toc117112970)

[1 Исследовательский раздел 4](#_Toc117112971)

[1.1 Описание предметной области 4](#_Toc117112972)

[1.2 Образ клиента](#_Toc117112973) 5

[1.3 Сценарии 5](#_Toc117112974)

[1.4 Сбор и анализ прототипов 6](#_Toc117112975)

[2 Проектирование приложения 10](#_Toc117112976)

[2.1 UI/UX дизайн проекта 10](#_Toc117112977)

[2.2 Выбор технологии, языка и среды программирования 12](#_Toc117112978)

[3 Разработка мобильного приложения 13](#_Toc117112979)

[3.1 Взаимодействие с файловой системой 13](#_Toc117112980)

[3.2 Разработка мультимедийного контента 13](#_Toc117112981)

[3.3 Описание используемых плагинов 14](#_Toc117112982)

[3.4 Описание разработанных процедур и функций 16](#_Toc117112982)

[4 Тестирование 20](#_Toc117112983)

[4.1 Протокол тестирования дизайна приложения 20](#_Toc117112984)

[4.2 Протокол тестирования функционала приложения 21](#_Toc117112985)

[Заключение 28](#_Toc117112986)

[Библиография 29](#_Toc117112987)

[Приложение А (обязательное) Техническое задание 30](#_Toc117112988)

Введение

Актуальность данного проекта заключается в растущем спросе на мобильные приложения для рисования и редактирования изображений, обусловленном распространением мобильных устройств и повышенным интересом пользователей к творчеству и самовыражению.

Целью курсового проекта является разработка графического редактора для мобильных устройств, аналогичного известному приложению Paint. Основной задачей проекта является создание удобного и многофункционального редактора, способного удовлетворить потребности пользователей в редактировании и создании графических изображений на мобильных устройствах.

Для достижения поставленной цели необходимо выполнить ряд задач:

* изучение предметной области, включая анализ существующих графических редакторов для мобильных устройств и выявление основных требований пользователей;
* разработка интерфейса и функциональности приложения с учетом полученных данных и собственных инновационных идей;
* написание кода приложения, обеспечивающего его стабильную работу и высокую производительность;
* тестирование готового продукта на соответствие заявленным требованиям и выявление возможных ошибок или несоответствий.

Реализуемый продукт – мобильное приложение, предназначенное для создания или редактирования изображений.

# Исследовательский раздел

## Описание предметной области

Графический редактор – это приложение, которое позволяет пользователю создавать и редактировать графические объекты на своем устройстве. Он может быть использован для создания рисунков, иллюстраций и других видов графики.

Основной функционал редактора включает в себя возможность загрузки изображений из галереи устройства или фотокамеры, а также скачивания и сохранения созданных работ. Пользователи могут легко импортировать фотографии для последующей обработки или создавать изображения с «нуля».

Одной из ключевых возможностей графического редактора является функция заливки цветом. Пользователи могут выбирать из широкой палитры цветов и наносить их на изображения в одно касание, делая процесс закрашивания быстрым и удобным.

Кроме того, редактор обеспечивает доступ к различным фигурам и формам, таким как круги, квадраты, треугольники и другие. Пользователи могут легко добавлять эти фигуры на свои работы, изменять их размеры, поворачивать и стилизовать с помощью различных инструментов.

Обширный набор кистей и инструментов позволяет пользователям экспериментировать с различными эффектами и стилями рисования. Благодаря наличию различных режимов наложения, масштабированию и изменению прозрачности, пользователи могут создавать уникальные и оригинальные произведения искусства.

Графический редактор для мобильного устройства станет незаменимым помощником для всех творческих и креативных людей, позволяя им всегда иметь под рукой удобный инструмент для воплощения своих идей в жизнь.

Графический мобильный редактор обычно предоставляет широкий набор инструментов для редактирования изображений, а также возможность добавлять текст, стикеры, эффекты и многое другое, что делает его удобным инструментом для создания и редактирования графики на ходу.

## Образ клиента

Графический редактор предназначен для людей, которые ценят творчество и выражение через изображения. Приложение ориентировано на широкую аудиторию, начиная от 6 лет и заканчивая взрослыми людьми. Графический редактор предоставляет возможность как создавать свои уникальные произведения и делиться ими с другими пользователями, так и удобно просматривать и редактировать изображения, которые доступны в приложении.

Он подходит как для людей, которые часто работают с графикой и дизайном, так и для тех, кто просто хочет добавить креативные элементы к своим фотографиям. Графический редактор предлагает широкий спектр инструментов и функций, что делает его удобным и доступным для разнообразных потребностей пользователей.

## Сценарии

Мама и дочка собираются создать совместное иллюстрированное руководство для проекта школьного театрального представления. Они используют мобильный графический редактор для создания ярких и креативных афиш и дизайна костюмов.

Парень для создания своих первых пиксель-арт работ. Он рисует простые, но стильные пиксельные изображения, такие как игровые персонажи или абстрактные пейзажи. Он также участвует в онлайн-соревнованиях по пиксель-арту, где использует редактор для создания уникальных и оригинальных работ.

Для своего блога о моде и стиле жизни девушка использует мобильный графический редактор для создания креативных и привлекательных постеров и изображений. Она также создает уникальные обложки для своих видеороликов и превью для своих Instagram Stories, привлекая больше подписчиков своим креативным контентом.

## Сбор и анализ прототипов

Pixlr Editor и Fotor - это два приложения для редактирования фотографий, доступные в Google Play и App Store. Оба этих графических редактора предоставляют широкий набор инструментов для редактирования изображений и фотографий, а также возможности сохранения, обмена и публикации работ.

Оба этих приложения имеют главный экран (рисунок 1).

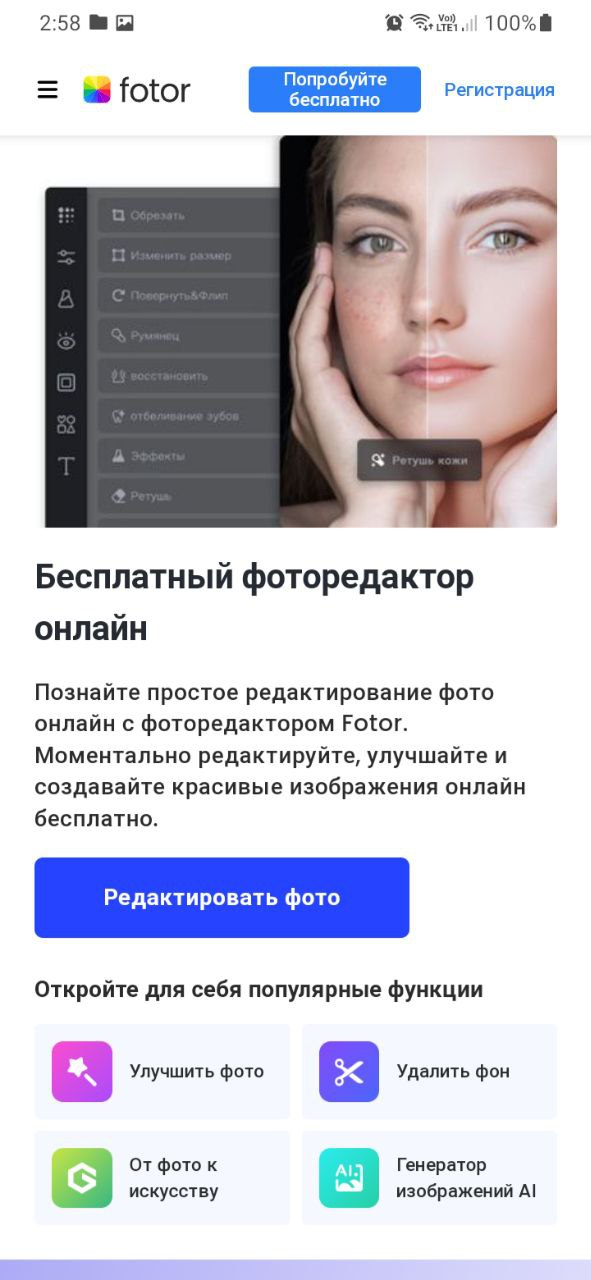
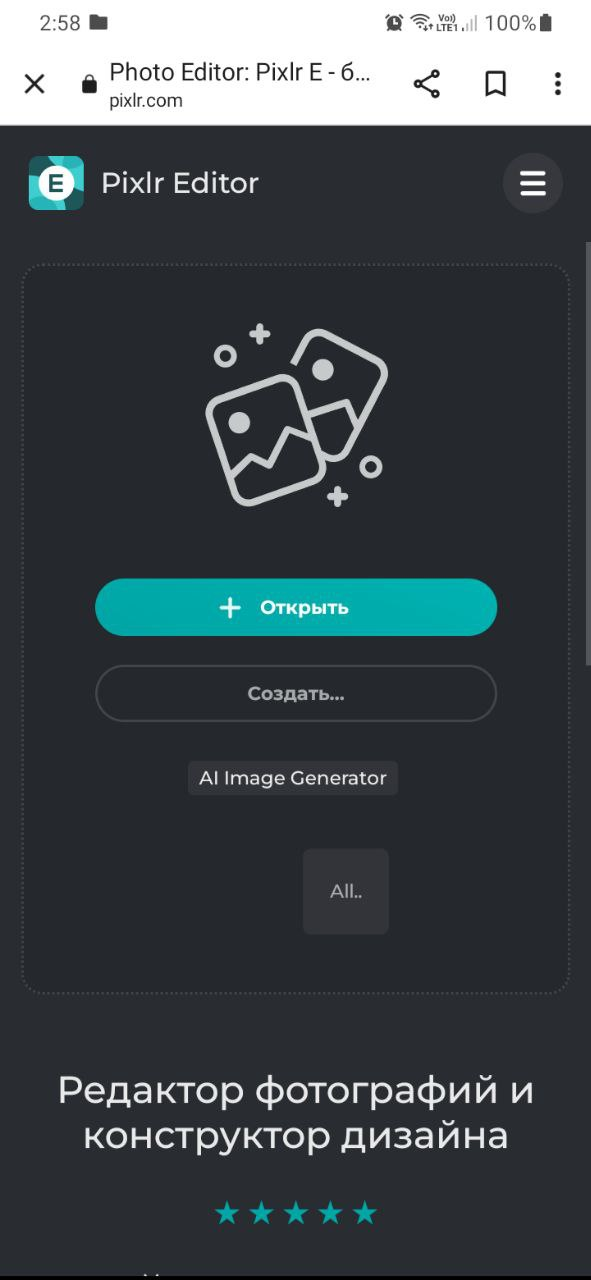


Рисунок 1 – Главный экран приложений

В графических редакторах Pixlr Editor и Fotor также представлены нижняя и верхняя панели, обеспечивающие навигацию по приложению.

Верхняя панель:

* Пользователь может получать уведомления о новых функциях, обновлениях или важных событиях в обоих приложениях.
* Оба редактора предоставляют возможность быстрого поиска инструментов, фильтров и других ресурсов для редактирования изображений.
* Здесь пользователь может найти информацию о своем аккаунте, настройки и доступ к своим сохраненным работам.

Нижняя панель:

* В этой вкладке пользователь может найти свои последние работы, новые функции или рекомендации от разработчиков.
* Этот раздел предназначен для загрузки новых изображений или фотографий для редактирования.
* Здесь пользователь может увидеть последние обновления, новости или предложения от разработчиков приложения.

Вывод изображений на главном экране также может отличаться. В Pixlr Editor и Fotor пользователь может увидеть свои последние работы или образцы изображений для редактирования. Логотип каждого редактора обычно располагается в одинаковом месте, например, в верхнем левом углу интерфейса (рисунок 2).

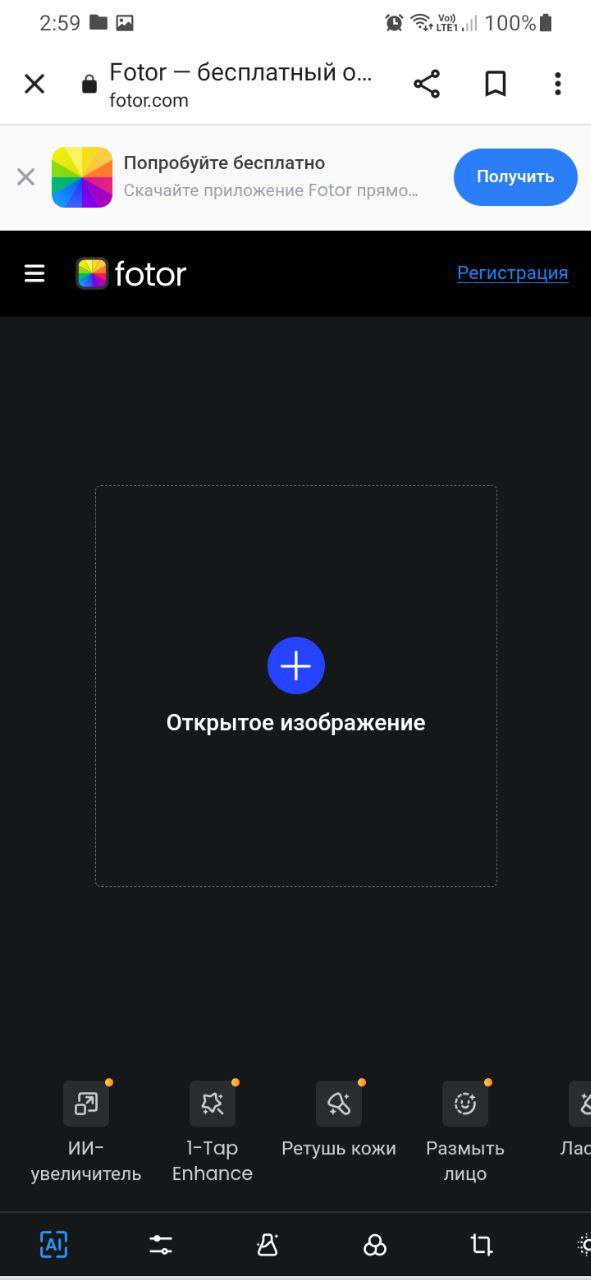
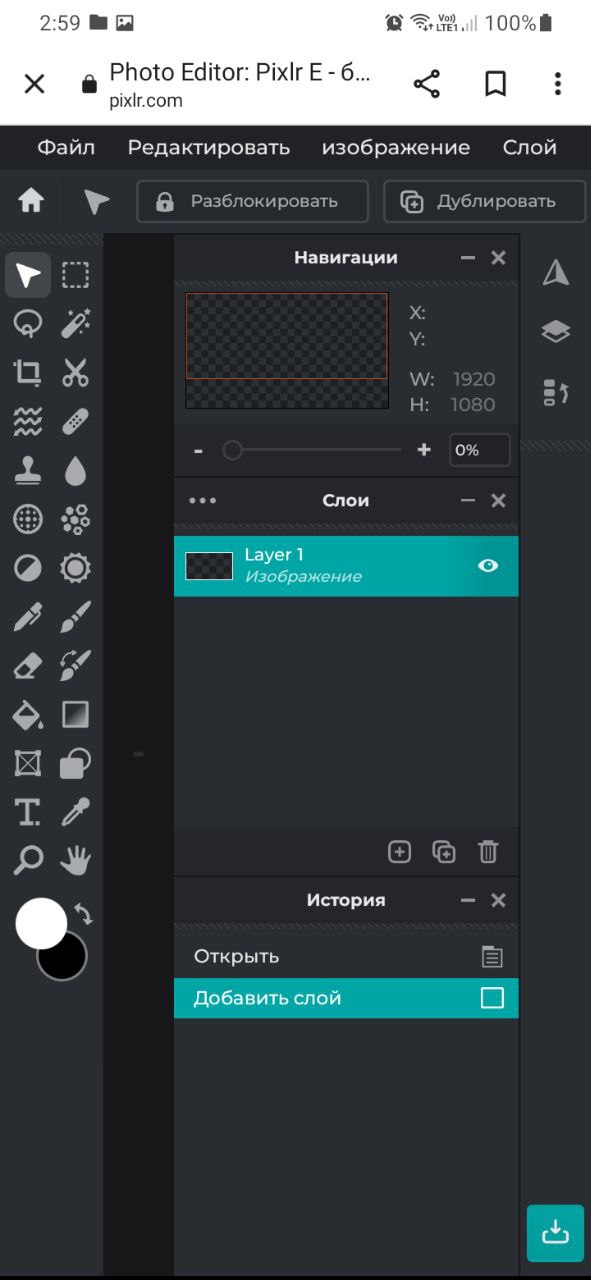


Рисунок 2 – Экраны редактирования

Экраны создания или редактирования могут также предоставлять функциональность авторизации, позволяя пользователям войти в свои аккаунты или зарегистрироваться. Это позволяет сохранять и синхронизировать созданный или отредактированный контент между различными устройствами, обеспечивать доступ к дополнительным функциям и инструментам, а также управлять личными настройками и предпочтениями.

В целом, оба графических редактора предоставляют схожий набор функций и инструментов, но могут отличаться в дизайне интерфейса и организации элементов, что может повлиять на удобство использования и предпочтения пользователей.

Сравнение по основным критериям данных приложений представлено в таблице 1.

Таблица 1 – Сравнение приложений

|  |  |  |
| --- | --- | --- |
| Параметры | Pixlr Editor | Fotor |
| Стоимость | Бесплатно | Бесплатно |
| Основной язык | Английский | Английский |
| Основная цель | Редактирование фотографий | Редактирование фотографий |
| Интерфейс | Интуитивно понятный | Интуитивно понятный |
| Набор инструментов | Разнообразный | Разнообразный |
| Режимы работы | Онлайн | Онлайн |
| Совместимость с устройствами | Веб-приложение | Мобильное приложение, Веб-приложение |
| Сохранение и обмен файлами | Облачное хранилище, Социальные сети | Облачное хранилище, Социальные сети |
| Редактирование снимков | Есть | Есть |
| Создание коллажей | Есть | Есть |
| Добавление эффектов и фильтров | Есть | Есть |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Ретушь и коррекция фотографий | Есть | Есть |
| Изменение размера и обрезка изображений | Есть | Есть |

Рассмотрев несколько приложений для редактирования фотографий, было принято решение разработать новое приложение с упрощенным интерфейсом и основной фокус на ключевых функциях, не перегружая пользователя лишними возможностями. В приложении будет реализована возможность редактирования фотографий с использованием разнообразных инструментов, а также создания коллажей и добавления эффектов.

За основу будут взяты лучшие аспекты интерфейса и функционала из различных приложений, чтобы создать удобное и интуитивно понятное приложение для пользователей. Важными функциями будут просмотр и редактирование фотографий, возможность сохранять результаты работы и обмениваться ими, а также легкий доступ к основным инструментам без избыточных сложностей.

# Проектирование приложения

1. **UI/UX дизайн проекта:**

Дизайн проекта разработан в программе Figma.

Для проекта были определены основные экраны:

* экран с выбором действия;
* экран создания нового изображения;
* экнан открытия существующего изображения.

Для цветовой схемы графического редактора используются основные цвета: белый и светло-серый. Фон приложения оформлен в светлых тонах, таких как белый или светло-серый, что создает легкую и воздушную атмосферу. Панели инструментов и меню выдержаны в тёмных тонах, чтобы контрастировать с основным фоном и обеспечить хорошую видимость текста и иконок. Инструменты рисования и кисти подобраны таким образом, чтобы они хорошо выделялись на светлом фоне и были удобны в использовании. Палитра цветов содержит пастельные оттенки и нежные цвета, что придает приложению лаконичный и эстетичный вид. Область предварительного просмотра оформлена с тёмным фоном, чтобы контрастировать с светлой темой и выделить предпросматриваемые изображения. Все используемые цвета показаны на рисунке 3.

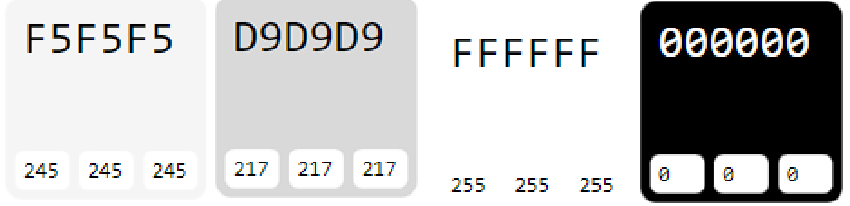


Рисунок 3 – Цветовая схема.

Логотип на белом фоне с черным текстом обладает высокой контрастностью, что обеспечивает легкость в восприятии и читаемости. Этот стильный и современный вариант подходит для брендов, желающих выразить чистоту и минимализм. Классическая черно-белая комбинация придает логотипу универсальный вид, что делает его подходящим для различных цветовых схем и фонов.

Ниже на рисунке 4 представлен логотип графического редактора.



Рисунок 4 – Логотип графического редактора

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

* экран с выбором действия(Start);
* экран создания нового изображения(Create);
* экнан открытия существующего изображения(Open).

Ниже на рисунке 5 представлен дизайн приложения.

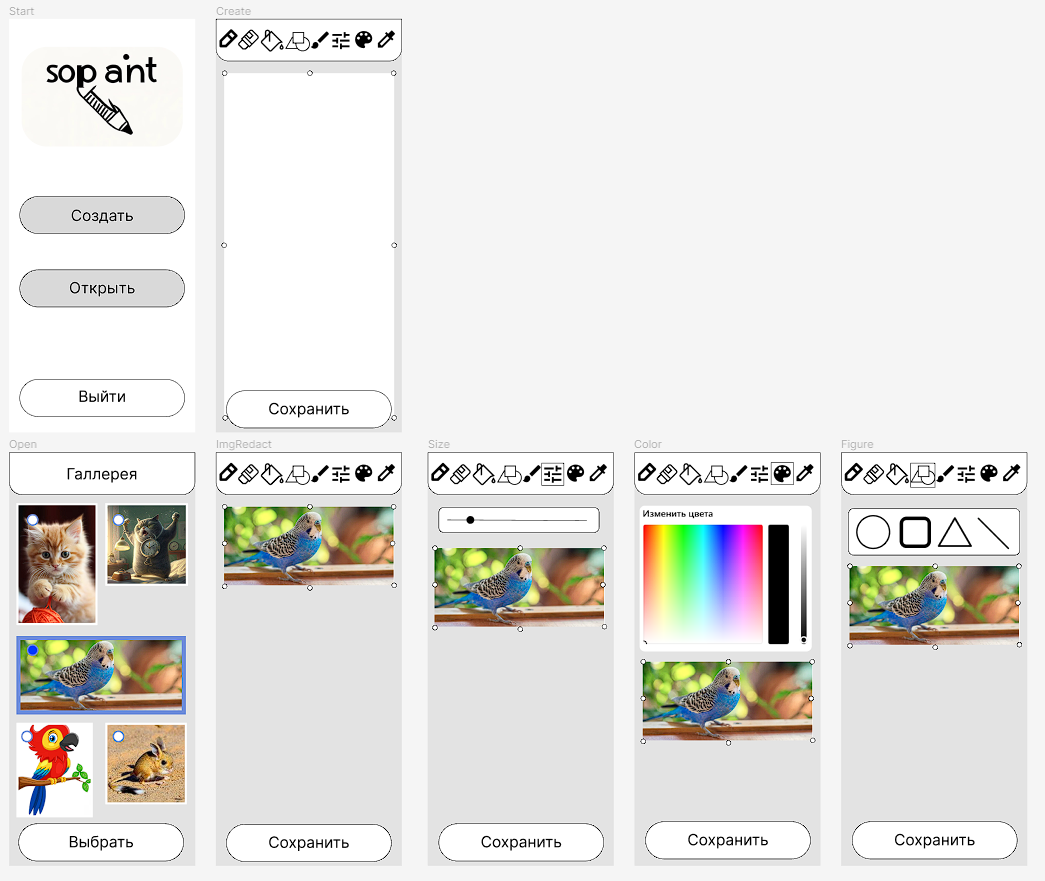


Рисунок 5 – Дизайн приложения

При открытии приложения пользователь увидит начальный экран с выбором действия. Здесь он может выбрать между созданием нового изображения или открытием уже существующего. Если пользователь выбирает создание нового изображения, он попадает на экран создания, где ему доступен пустой холст и набор инструментов для рисования и редактирования. Если же выбрано открытие существующего изображения, открывается экран просмотра изображений, где пользователь может просмотреть список доступных изображений или использовать поиск для нахождения нужного. После выбора изображения открывается экран редактирования, где пользователю предоставляются различные инструменты и функции для изменения изображения. По завершении редактирования пользователь может сохранить изменения, нажав на кнопку "Сохранить", что переведет его на экран сохранения, где он может выбрать параметры сохранения и место для сохранения файла.

1. **Выбор технологии, языка и среды программирования:**

Средой программирования выбрана программа Android Studio, 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.

Работа с файлами в Android Studio может включать в себя множество задач, таких как чтение и запись файлов на внутреннее хранилище устройства, работа с файлами внешнего хранилища, использование различных API для работы с файлами.

**Разработка мобильного приложения**

1. **Взаимодействие с файловой системой:**

Пользователь приложения имеет возможность выбирать файлы из файловой системы устройства с помощью стандартных средств Android, таких как системный файловый диалог или специальное окно выбора файлов в приложении. После выбора файла приложение автоматически осуществляет чтение данных из него, что позволяет просматривать и редактировать информацию, такую как изображения или данные проекта, прямо на экране редактора. Пользователь также может вносить изменения в отображаемые данные, например, рисовать на изображении. После внесения изменений пользователь имеет возможность сохранить их обратно в файловую систему устройства для последующего доступа. Приложение также предоставляет функциональность управления файлами, включая создание новых файлов, удаление существующих и переименование файлов. Для работы с файловой системой приложение запрашивает необходимые разрешения на доступ, такие как разрешения на чтение и запись во внутреннее хранилище устройства или на SD-карт, а также обеспечивает обработку возможных ошибок, связанных с доступом к файлам.

1. **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 6). Такой способ хранения уменьшает вес приложения, а также, избавляет от проблем с потерей качества мультимедийного контента. Все ресурсы хранятся в папке «drawable»

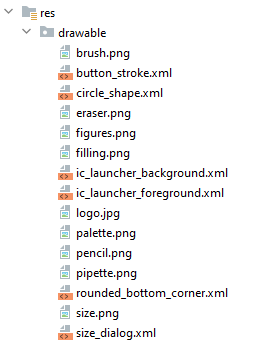


Рисунок 6 – Мультимедийный контент

Приложение использует светлую цветовую схему, которая хранится в файле colors.xml. Эти цвета определяют тему приложения для светлой темы. Пользователь имеет возможность выбрать светлую тему, чтобы наслаждаться ярким и приятным визуальным опытом.

Цвета в файле colors.xml могут быть адаптированы под различные элементы пользовательского интерфейса, такие как фоны, текст, кнопки и т. д. Они обеспечивают консистентность визуального оформления приложения и повышают его привлекательность для пользователей.

1. **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 |
| libs.junit | Библиотека JUnit для модульного тестирования Java-кода. |
| libs.material | Библиотека компонентов интерфейса. |

Продолжение таблицы 2

|  |  |
| --- | --- |
| libs.constraintlayout | Библиотека ConstraintLayout для создания гибких и адаптивных макетов пользовательского интерфейса. |
| libs.appcompat | Библиотека AppCompat для обеспечения обратной совместимости с новыми функциями и возможностями Android для более старых версий платформы. |
| libs.ext.junit | Библиотека расширений для JUnit, используемая для написания тестов на языке Kotlin. |
| libs.espresso.core | Библиотека Espresso Core для написания функциональных тестов пользовательского интерфейса на Android. |
| com.github.QuadFlask:colorpicker:0.0.13 | Это библиотека, которая предоставляет пользовательский интерфейс для выбора цвета в приложении. Она включает в себя различные элементы управления, такие как ползунки для выбора оттенка цвета, текстовое поле для ввода hex-кода цвета и другие. Библиотека проста в использовании и легко интегрируется в приложение, что позволяет пользователям выбирать нужные цвета с удобством. |
| com.github.siyamed:android-shape-imageview:0.9.+@aar | Эта библиотека предоставляет кастомный вид ImageView, который позволяет создавать изображения с различными формами, такими как круг, овал, прямоугольник, треугольник и многое другое. Она также поддерживает установку границы и фона для этих форм, что делает ее полезной для создания интересного и креативного пользовательского интерфейса. |

Вышеперечисленные библиотеки играют ключевую роль в разработке приложения, обеспечивая его функциональность и возможности. Они предоставляют инструменты и функции, необходимые для реализации задуманных идей и обеспечивают правильное функционирование приложения.

Без этих библиотек приложение не смогло бы реализовать все свои функции и предоставить пользователю удобный и привлекательный пользовательский опыт. Они являются неотъемлемой частью разработки приложения, обеспечивая его успешное выполнение и повышая его привлекательность для пользователей.

1. **Описание разработанных процедур и функций:**

В приложении реализованы следующие методы (таблица 3):

Таблица 3 – Методы приложения

|  |  |
| --- | --- |
| Метод | Описание |
| onCreate(Bundle savedInstanceState) | Метод, который вызывается при создании активности. В этом методе инициализируются элементы интерфейса и устанавливаются слушатели для кнопок. |
| onActivityResult(int requestCode, int resultCode, Intent data) | Метод, который вызывается при получении результата от другой активности, запущенной с помощью. |
| startActivityForResult() | Метод используется для получения выбранного изображения из галереи. |
| onClick(View v) | Метод, который вызывается при щелчке на кнопке. В этом методе определяется поведение для каждой из кнопок (создание нового рисунка, открытие изображения из галереи, выход из приложения). |

Продолжение таблицы 3

|  |  |
| --- | --- |
| showSizePanel() | Пользовательский метод, который создает и отображает диалоговое окно для выбора размера рисования. |
| setPencilColor(int color) | Этот метод устанавливает цвет карандаша. |
| setToolSize(float currentSize) | Этот метод устанавливает размер инструмента (карандаша, ластика или кисти). |
| createPaint(DrawingTool tool, float size) | Этот метод создает и возвращает объект Paint, используемый для рисования линий в зависимости от выбранного инструмента (карандаш, ластик или кисть) и размера инструмента. |

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

Таблица 4 – Классы приложения

|  |  |
| --- | --- |
| Класс | Описание |
| DrawingView | Этот класс является подклассом View и представляет пользовательский вид, предназначенный для рисования на экране. Он отвечает за обработку касаний, рисование линий, кругов и других фигур на канве, а также управление цветом и размером инструментов (карандаш, ластик, кисть). |
| DrawingTool | Это перечисление, определяющее типы рисовальных инструментов, такие как карандаш, ластик и кисть. |

Продолжение таблицы 4

|  |  |
| --- | --- |
| PathHistory | Этот внутренний статический класс представляет историю рисования. Он содержит объект Path, который представляет собой контур фигуры, нарисованной пользователем, и объект DrawingTool, указывающий на тип инструмента, используемого для рисования этой фигуры. |
| SizeDialogFragment | Этот класс является подклассом DialogFragment и представляет диалоговое окно выбора размера для рисования. Внутри метода onCreateDialog() создается и настраивается диалоговое окно с использованием AlertDialog.Builder. Он содержит слушателя для SeekBar, чтобы отслеживать изменения размера, а также метод applySize(), который применяет выбранный размер к инструменту рисования (DrawingView). |
| Start | Этот класс является подклассом AppCompatActivity и представляет активити стартового экрана приложения. В методе onCreate() устанавливаются обработчики нажатия на кнопки для создания нового рисунка, открытия изображения из галереи и завершения работы приложения. Также переопределен метод onActivityResult(), который вызывается после выбора изображения из галереи. |
| Create | Этот класс представляет собой активность (Activity) Android-приложения и отвечает за создание и управление процессом создания изображения в графическом редакторе. Он содержит логику для отображения пользовательского интерфейса, обработки событий нажатия на кнопки и сохранения рисунка. |

В графическом редакторе, отсутствие авторизации и базы данных не является необычным, особенно если приложение ориентировано на простоту использования и редактирование изображений на устройстве пользователя. Вместо этого, редактор может предоставлять возможность создавать, редактировать и сохранять изображения локально на устройстве.

Листинг 1 – Логика создания пустого изображения



При нажатии на эту кнопку, пользователь переходит на экран для создания нового изображения. Экран содержит редактор, где пользователь может рисовать или вносить изменения в изображение.

Создание пустого изображения в этом контексте означает инициализацию редактора с пустым холстом или изображением без какого-либо содержимого. Это позволяет пользователю начать работу с редактором с чистого листа, чтобы создать новое изображение или рисунок.

# Тестирование

## Протокол тестирования дизайна приложения

Тестирование дизайна приложения проводится на минимальном (Android SDK 30) и на более позднем (Android SDK 33) с различной диагональю экранов для проверки разметки страниц и вёрстки приложения.

Примеры проверок отображения элементов на экране представлены на рисунках 7–8.

Рисунок 7 – Стартовые экраны

Элементы интерфейса в обоих случаях отображаются корректно.

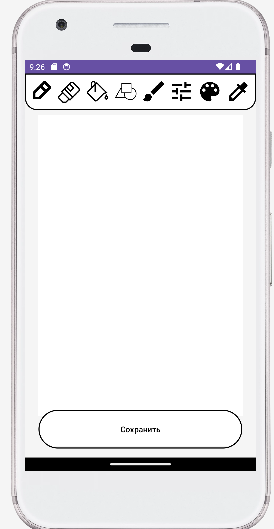


Рисунок 8 – Экраны создания изображений

Экран создания изображений так же отображается корректно на обоих устройствах, все элементы интерфейса расположены на своих местах.

UI-тестирование играет ключевую роль в обеспечении качества программного обеспечения и удовлетворении потребностей пользователей. Регулярное и комплексное тестирование пользовательского интерфейса помогает выявить и устранить проблемы до их выхода в продакшн, что в конечном итоге способствует созданию более удобных и привлекательных приложений для пользователей.

## Протокол тестирования функционала приложения

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

В таблице 5 представлено тестирование функции создания нового изображения.

Таблица 5 – Тестирование функции создания нового изображения

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Создание нового изображения | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Нажать на кнопку "Создать" | Переход на экран создания нового рисунка | Пройден |

В таблице 6 представлено тестирование функции карандаша.

Таблица 6 – Тестирование функции карандаша

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Карандаш | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Выбрать инструмент "Карандаш" | Переход на экран создания нового рисунка | Пройден |
| Нажать на поле для рисования | Появляется карандаш на экране, готовый к рисованию | Пройден |
| Начать рисовать | Линия начинает появляться на экране под курсором | Пройден |
| Закончить рисование | Рисование завершено, линия сохранена на экране | Пройден |

В таблице 7 представлено тестирование функции ластика.

Таблица 7 – Тестирование функции ластика

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Ластик | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |

Продолжение таблицы 7

|  |  |  |
| --- | --- | --- |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Выбрать инструмент "Ластик" | Переход на экран создания нового рисунка | Пройден |
| Нажать на область, которую нужно стереть | Область выделяется для удаления | Пройден |
| Начать стирать | Линии в области стираются | Пройден |
| Закончить стирать | Стерание завершается | Пройден |

В таблице 8 представлено тестирование функции отрисовки фигур.

Таблица 8 – Отрисовка фигур

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Отрисовка фигур | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Выбрать инструмент "Фигуры" | Переход на экран создания нового рисунка | Пройден |
| Нажать на поле для отрисовки фигуры | Фигура появляется на экране в выбранной позиции | Пройден |
| Подтвердить отрисовку фигуры | Фигура сохраняется на экране | Пройден |

В таблице 9 представлено тестирование функции заливки.

Таблица 9 – Функция заливки

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Заливка | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |

Продолжение таблицы 9

|  |  |  |
| --- | --- | --- |
| Выбрать инструмент "Заливка" | Переход на экран создания нового рисунка | Пройден |
| Нажать на поле для заливки | Область, по которой произведено нажатие, заполняется цветом заливки | Пройден |
| Подтвердить заливку | Область, внутри которой произведено нажатие, заполняется цветом заливки | Пройден |
| Перекрасить заливку в другой цвет | Цвет заливки изменяется на выбранный | Пройден |

В таблице 10 представлено тестирование функции кисти.

Таблица 10 – Функция кисти

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Кисть | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Выбрать инструмент "Кисть" | Переход на экран создания нового рисунка | Пройден |
| Нажать на поле для рисования | Появляется кисть на экране, готовая к рисованию | Пройден |
| Начать рисовать | Линия начинает появляться на экране под курсором | Пройден |
| Закончить рисование | Рисование завершено, линия сохранена на экране | Пройден |
| Перекрасить линию в другой цвет | Цвет линии изменяется на выбранный | Пройден |

В таблице 11 представлено тестирование функции изменения размера.

Таблица 11 – Функция изменения размера

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Изменение размера | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |

Продолжение таблицы 11

|  |  |  |
| --- | --- | --- |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Выбрать инструмент "Кисть" | Переход на экран создания нового рисунка | Пройден |
| Нажать на иконку размера | Появляется скроллбар для выбора размера кисти | Пройден |
| Изменить размер кисти в скроллбаре | Размер кисти меняется соответственно выбранному значению | Пройден |
| Начать рисовать | Линия начинает появляться на экране под курсором | Пройден |
| Закончить рисование | Рисование завершено, линия сохранена на экране | Пройден |

В таблице 12 представлено тестирование функции палитры.

Таблица 12 – Функция палитры

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Палитра | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Выбрать инструмент "Палитра" | Переход на экран выбора цвета | Пройден |
| Выбрать цвет из палитры | Выбранный цвет отображается на экране | Пройден |
| Начать рисовать | Линия начинает появляться на экране под курсором, используя выбранный цвет | Пройден |
| Закончить рисование | Рисование завершено, линия сохранена на экране с выбранным цветом | Пройден |

В таблице 13 представлено тестирование функции пипетки.

Таблица 13 – Функция пипетки

|  |  |
| --- | --- |
| **Название**: | SoPaint |
| **Функция**: | Пипетка |

Продолжение таблицы 13

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Выбрать инструмент "Пипетка" | Переход на экран выбора цвета | Пройден |
| Навести пипетку на область с нужным цветом | Выбранный цвет изображается в палитре | Пройден |
| Нажать на кнопку "Выбрать" | Выбранный цвет активируется | Пройден |
| Начать рисовать | Линия начинает появляться на экране под курсором, используя выбранный цвет | Пройден |
| Закончить рисование | Рисование завершено, линия сохранена на экране с выбранным цветом | Пройден |

В таблице 14 представлено тестирование функции сохранения.

Таблица 14 – Функция сохранения

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Сохранение | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Нажать на кнопку "Сохранить" | Появляется всплывающее окно с параметрами сохранения (например, формат, разрешение, имя файла) | Пройден |
| Выбрать желаемые параметры сохранения | Параметры сохранения выбраны пользователем | Пройден |
| Нажать на кнопку "Сохранить" в окне | Изображение сохраняется в соответствии с выбранными параметрами в указанном месте | Пройден |
| Проверить, что изображение сохранено | Изображение сохранено в указанном месте с выбранными параметрами | Пройден |

В таблице 15 представлено тестирование функции выхода.

Таблица 15 – Функция выхода

|  |  |  |
| --- | --- | --- |
| **Название**: | SoPaint | |
| **Функция**: | Выход | |
| **Действие** | **Ожидаемый** **результат** | **Результат** **теста**:  пройден  провал  заблокирован |
| **Предусловие**: |  | |
| Запустить приложение «SoPaint» | Приложение открылось корректно |  |
|  |  |  |
| **Шаги** **теста (positive)**: |  | |
| Нажать на кнопку "Выход" | Появляется всплывающее окно с запросом подтверждения выхода | Пройден |
| Подтвердить выход | Приложение закрывается | Пройден |
| Проверить, что приложение закрыто | Приложение полностью закрывается и пользователь возвращается на рабочий стол или экран с приложениями | Пройден |

Разработанные TestCase подтверждают эффективную работу ключевых функций приложения. В процессе тестирования не выявлено ошибок в логике. Кроме того, дизайн и визуальное представление каждого экрана были проверены на соответствие заданным стандартам и корректность отображения элементов. Общая структура приложения и его логика также были подвержены проверке, и они демонстрируют надлежащую работу.

Заключение

В ходе разработки мобильного графического редактора были учтены предпочтения и потребности пользователей, ориентированных на создание и редактирование изображений с минимальными усилиями. Реализованная функция быстрого доступа к инструментам и настроек редактора позволяет пользователям легко и интуитивно взаимодействовать с приложением.

Интерфейс редактора был разработан с акцентом на удобство использования и минимальное количество шагов для достижения пользовательских целей. Основные инструменты и функции редактирования размещены таким образом, чтобы обеспечить максимальную доступность и эффективность их использования.

Особое внимание уделено функции редактирования изображений с помощью различных инструментов, таких как карандаш, ластик, кисть, фигуры и заливка. Пользователям предоставляется возможность создания и редактирования графических элементов, а также управления размером кисти и выбором цвета из палитры.

Функция сохранения и экспорта изображений реализована интуитивно понятно, позволяя пользователям сохранять свои работы в различных форматах и делиться ими с другими.

Тестирование дизайна и функциональности редактора подтвердило его корректность и эффективность работы. Это гарантирует соответствие приложения поставленным целям и требованиям пользователей.

В целом, разработка мобильного графического редактора обеспечила создание функционального и удобного приложения, которое удовлетворяет потребности пользователей в редактировании изображений. Полученные результаты являются 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.

Электронные ресурсы:

1. Android Developers [Электронный ресурс]. – Документация Android Studio. – URL: https://developer.android.com/docs/ (дата обращения: 25.04.2024)
2. Java HTML Parser [Электронный ресурс]. – Вытягивание данных с страницы. – URL: https://jsoup.org/ (дата обращения: 10.04.2024)
3. Metenit [Электронный ресурс]. – Ресурсы Color и установка цвета. – URL: https://metanit.com/ (дата обращения: 10.04.2024)
4. Figma [Электронный ресурс]. – Прототипирование проектов. – URL: https://www.figma.com/ (дата обращения: 20.01.2024)

**Приложение А**

(обязательное)

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ ТИПА PAINT

Техническое задание

НАТКиГ.200700.010.000ПЗ

Выполнил:

Студент группы ПР-21.101

Ефименко Е.С.

2024

**Содержание**

Введение 31

1 Назначения разработки 32

2 Требования к мобильному приложению 32

2.1 Требования к функциональным характеристикам 32

2.2 Требования к надёжности 32

2.3 Условия эксплуатации 32

2.4 Требования к составу и параметрам технических средств 33

2.5 Требования к информационной и программной совместимости 33

2.6 Требования к защите информации 33

2.7 Требования к маркировке и упаковке 33

3 Требования к программной документации 33

4 Технико-экономические показатели 35

5 Стадии и этапы разработки 35

6 Порядок контроля и приёмки 36

**Введение**

Настоящее техническое задание распространяется на разработку мобильного приложения «Разработка графического редактора для мобильных устройств типа Paint», используемого для создания или редактирования изображений.

Наименование приложения: «SoPaint».

Краткая характеристика области применения: мобильное приложение предоставляет возможность создавать или редактировать изображения.

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

Наименование темы разработки − «Разработка графического редактора для мобильных устройств типа Paint».

Условное обозначение темы разработки – «SoPaint».

**1 Назначение разработки**

Основное назначение приложения заключается в:

* обеспечении удобного и понятного интерфейса приложения для пользователя;
* предоставление возможности создания и редактирования изображений с помощью различных инструментов.

Лица, которые могут работать с данной системой:

администратор – управляет полностью всем приложением, следит за его работоспособностью и могут выполнять техническую поддержку;

пользователь приложения – могут создавать, редактировать и сохранять изображения, а также использовать различные инструменты редактирования.

**2 Требования к мобильному приложению**

**2.1 Требования к функциональным характеристикам**

Требования к составу выполняемых функций:

* создание нового изображения;
* редактирование изображение;
* сохранение и экспорт изображений;

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* корректной работы интерфейса и инструментов редактора.
* обработки ошибок и исключений для предотвращения сбоев приложения.

**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 10.0.

**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 сохранённым изображениям имеет только пользователь.

**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 | 24.02.2024 | Пояснительная записка |
| 2 | Описание предметной области | 25.02.2024 | Пояснительная записка |
| 3 | Выбор технологии, языка и среды программирования | 27.02.2024 | Пояснительная записка |
| 4 | Оформление технического задания | 29.02.2024 | Техническое задание |
| 5 | Проектирование UI/UX дизайна | 02.03.2024 | Спецификации программного обеспечения |
| 6 | Разработка мобильного приложения | 07.03.2024 | Схема структурная системы и спецификации компонентов |
| 7 | Разработка базы данных | 16.03.2024 | Программный  продукт |
| 8 | Отладка и тестирование приложения | 18.03.2024 | Тексты программных компонентов |
| 9 | Оформление документации | 20.03.2024 | Программная  документация |
| 10 | Защита | 29.04.2024 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

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