Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ УЧЁТА И АНАЛИЗА ДОМАШНИХ ФИНАНСОВ

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

МДК 01.03 Разработка мобильных приложений

НАТКиГ.201100.010.000ПЗ

Разработал:

студент группы ПР-20.101

Колпаков Я. А.

2023

**Содержание**

[Введение 5](#_Toc138371568)

[1 Исследовательский раздел 7](#_Toc138371569)

[1.1 Описание предметной области 7](#_Toc138371570)

[1.2 Образ клиента 8](#_Toc138371571)

[1.3 Сценарии 8](#_Toc138371572)

[1.4 Сбор и анализ прототипов 9](#_Toc138371573)

[2 Проектирование приложения 14](#_Toc138371574)

[2.1 UI/UX дизайн проекта 14](#_Toc138371575)

[2.2 Выбор технологии, языка и среды программирования 18](#_Toc138371576)

[3 Разработка мобильного приложения 20](#_Toc138371577)

[3.1 Разработка базы данных 20](#_Toc138371578)

[3.2 Описание используемых плагинов 24](#_Toc138371579)

[3.3 Описание разработанных процедур и функций 24](#_Toc138371580)

[4 Тестирование 29](#_Toc138371581)

[4.1 Протокол тестирования дизайна приложения 29](#_Toc138371582)

[4.2 Протокол тестирования функционала приложения 29](#_Toc138371583)

[Заключение 35](#_Toc138371584)

[Библиография 37](#_Toc138371585)

[Приложение А (обязательное) Техническое задание 38](#_Toc138371586)

[Приложение Б 47](#_Toc138371620)

Введение

Актуальность разработки мобильного приложения обусловлена тем, что учет денежных средств – это важнейшая область бюджетного учета, в ее рамках формируется исчерпывающая информация о состоянии расчетов, обязательств бюджетного учреждения. Денежные средства являются одним из основных элементов имущества субъектов, обеспечивающих ведение хозяйственной, предпринимательской или иной финансовой деятельности. Практически любая операция, так или иначе, влечет за собой изменения в этих активах. Формирование следующих действий финансовой деятельности ссылается на исходную текущую информацию, сформированную для реального пользователя. Поэтому особенно важно в учете правильное оформление информации о денежных средствах.

Цель разработки мобильного приложения для учёта и анализа домашних финансов является контроль за соблюдением доходной и расчетной дисциплины, правильность и эффективность использования денежных средств, обеспечение оптимизированной подачи информации о денежной деятельности в укомплектованном приложении.

Задачами курсового проекта в связи с указанной целью являются:

* изучить всевозможные денежно-расходные виды деятельности бедующего пользователя.
* рассмотреть области применения разрабатываемого мобильного приложения.
* изучить способы представления данных пользователю с помощью графиков и статистики.
* рассмотреть конкурентноспособные приложения для анализа предметной области.
* предусмотреть конфиденциальность информации пользователя.
* предусмотреть сбор данных наличного формата трат.

Объект исследования – пользователь, обладающий какими-либо финансовыми средствами и выполняющий какие-либо операции над данными средствами. Выполняющий роль источника информации для организации данных.

Предметом исследования является организация финансового учета и анализа денежных средств в мобильном приложении «Mosticsmo».

# Исследовательский раздел

## Описание предметной области

В предметной области рассматривается деятельность людей по учету своих доходов и расходов как личных, так и общих внутри семьи. Для учета расходов и доходов предназначены специальные документы, соответственно, по расходам и доходам. Расходы и доходы выполняются пользователями приложения. Расходы выполняются по различным категориям и видам расходов (например, категория «Развлечения», к которой можно отнести вид расходов «Кино»). Доходы происходят за счет выплат зарплат, пенсий, доходов от предпринимательской деятельности и т.п., следовательно, у доходов тоже есть свои категории и виды.

Разрабатываемое мобильное приложение предназначено для одной категории пользователей: члены семьи, и должна выполнять следующие функции:

* учет расходов и доходов;
* планирование будущих расходов и доходов;
* авторизация пользователей;
* возможность создавать виртуальные «кошельки» и «копилки» для учета денежных средств;
* возможность просмотра актуального баланса «кошельков» и «копилок»;
* отчетность по доходам, расходам и их анализу;
* возможность создавать удобные пользователю категории расходов/доходов и относящиеся к ним виды расходов/доходов;
* возможность пересчета записей о расходах из других валют;
* загрузка валютных курсов через интернет.

## Образ клиента

Пользователь, владеющий каким-либо мобильным устройством, позволяющим пользоваться разрабатываемым приложением, а также наличие открытого доступа в интернет. Платёжеспособная личность, имеющая постоянный доход или же иные способы заработка. К группе пользователей можно отнести владельцев каких-либо денежных средств. Пользователь, заинтересованный в использовании разрабатываемого приложения.

## Сценарии

При первом заходе в приложение пользователю предлагают авторизоваться, если у него уже есть аккаунт. Если аккаунта у пользователя нет, то он должен авторизоваться для дальнейшего использования приложения. После регистрации или авторизации пользователь попадает в главное меню с статистикой его расходов.

Пользователь также может перейти в раздел «Накопления», то есть если пользователь имеет какие-либо счета в банке или же акции, то по параметрам, которые он ввел будет высчитываться и выдаваться информация о данных накоплениях.

Если же пользователь предъявит желание о изменении настоек приложения, то он переходит в раздел настроек и ему предоставляется возможность изменить вид своего приложения. Также в настройках он может изменить валюту и другие всевозможные параметры.

Если пользователю пришло уведомление о том, что он давно не вел статистику наличных расчетов, ему предлагается сделать это. Пользователь имеет право выбора и сообщить о отсутствии наличных расчетов или же перейти в раздел «Наличные расчеты». В интерфейсе данной вкладки пользователь должен самостоятельно заполнить категории и указать сферу, где была выполнена оплата.

## Сбор и анализ прототипов

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.

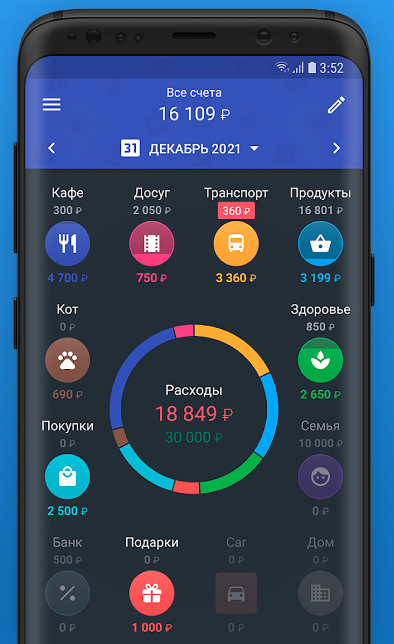


Рисунок 1.2 – интерфейс приложения «1Money: расходы финансы бюджет»

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.

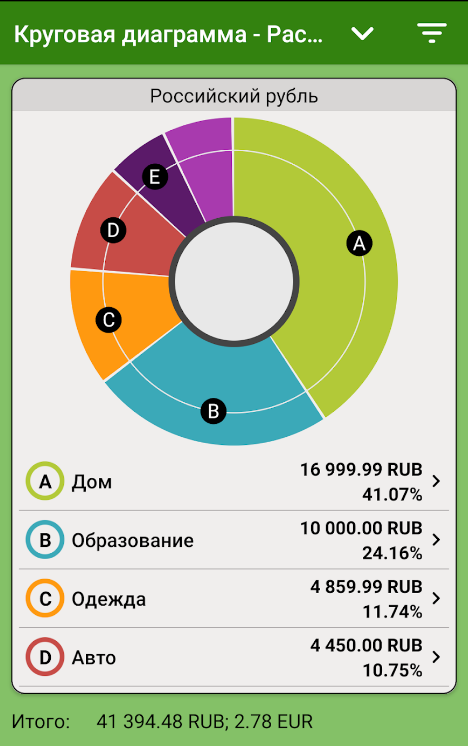


Рисунок 1.3 – интерфейс приложения «Учет расходов, семейный бюджет»

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.

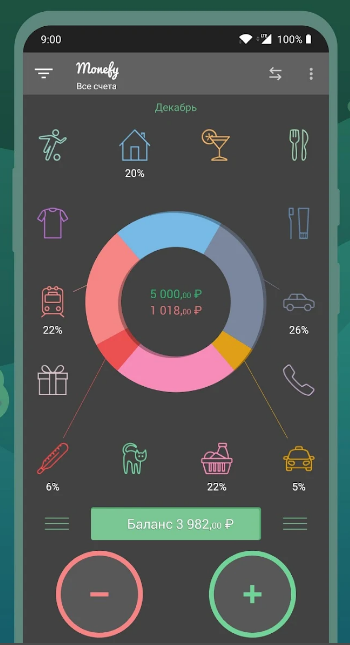


Рисунок 1.4 – интерфейс приложения «Monefy – Менеджер бюджета»

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 приложений можно сделать вывод, что положительными качествами разрабатываемого приложения должны быть:

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

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

# Проектирование приложения

## UI/UX дизайн проекта

Дизайн проекта «Mosticsmo» разработан в программе Figma.

Для проекта были определены основные экраны:

* экран регистрации;
* экран авторизации;
* экран прогрузки в приложение;
* экран данных о финансах пользователя;
* экран панели управления приложения;
* экран добавления финансовых трат.

Цветовая гамма была выбрана посредством подбора цветовой гаммы для более быстрого восприятия информации. Выбирая цвета был сделан упор на контрастность объектов по отношению к фону. Обращая внимание на контрастный объект, пользователь быстрее воспринимает необходимую ему информацию.

Потому было принято решение в разработке макета отдать предпочтение спокойным цветам: бежевый, черный, белый. Но стоит заметить, что данные цвета, разбавленные изображениями, не выглядят скучно, а весьма лаконично.

Ниже на рисунке (Рисунок 2.1) представлен логотип «Mosticsmo».



Рисунок 2.1 – Логотип приложения «Mosticsmo»

На логотипе изображена тематическая приложению иллюстрация.

На рисунке (Рисунок 2.2) ниже представлена страница регистрации.

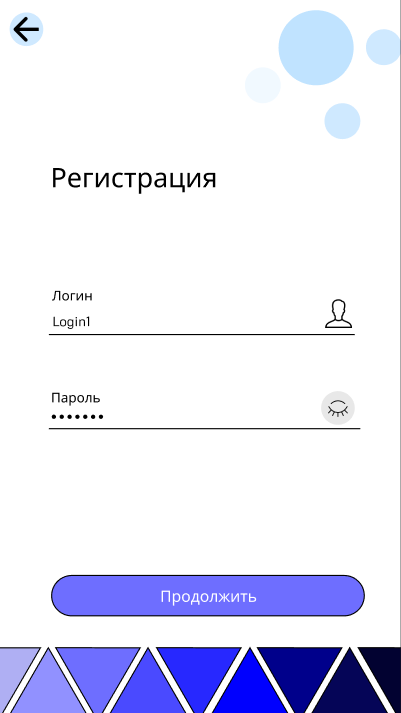


Рисунок 2.2 – Страница «Регистрации»

На данной странице мобильного приложения предоставлена возможность зарегистрироваться новому пользователю, а в случае существования аккаунта у пользователя, он может перейти на страницу входа.

На рисунке (Рисунок 2.3) ниже представлена страница входа в приложение



Рисунок 2.3 – Страница «Входа»

На данной странице имеется возможность входа в приложение, если пользователь ещё не зарегистрирован, он может перейти к странице регистрации.

На рисунке (Рисунок 2.4) ниже представлена страница показа собранных данных за определённый период.

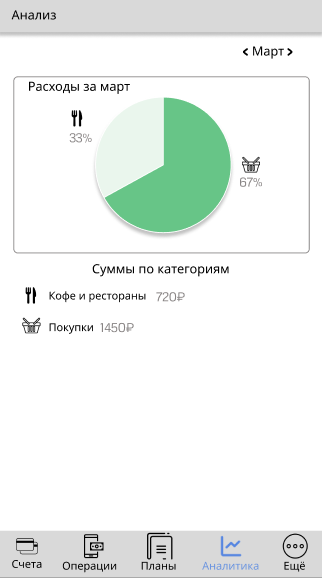


Рисунок 2.4 – Страница «Статистики затрат»

На данной странице клиент имеет возможность просматривать собранные данные за определённый период. Для этого ему достаточно выбрать период, за который были собраны данные и увидеть результат.

На рисунке (Рисунок 2.5) ниже представлена вкладка с счетами пользователя.

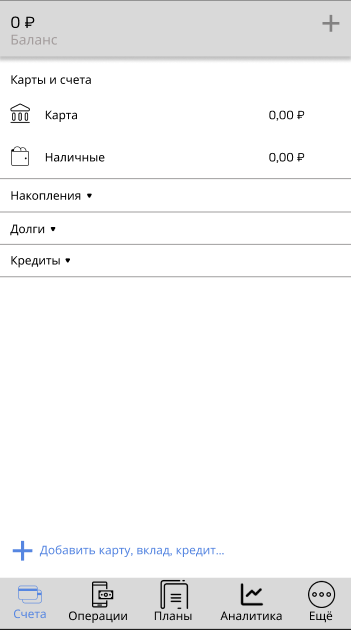


Рисунок 2.5 – Страница «Меню»

На данной странице отображаются все созданные ранее счета, также на данной странице он может создать новый счет.

На рисунке (Рисунок 2.6) ниже представлена страница добавления расходов по категориям вручную.

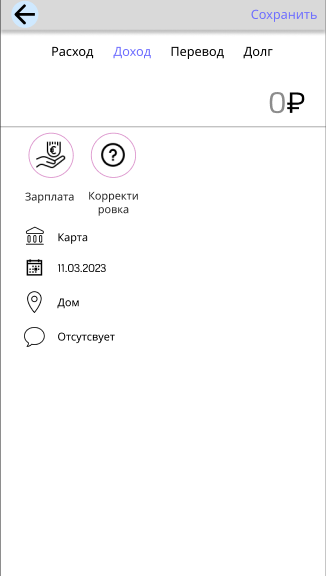


Рисунок 2.6 – Страница «Ручное добавление расходов»

На данной странице клиент имеет возможность вручную добавит денежные расходы, которые были оплачены в наличном виде или картой.

На рисунке (Рисунок 2.6) ниже представлена страница планирования расходов.

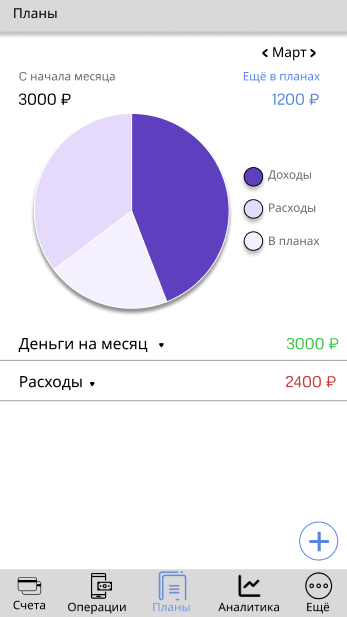


Рисунок 2.6 – Страница «Загрузки приложения»

На данной странице представлена планировать свои бедующие расходы и доходы, также просматривать их в удобной диаграмме отображающая все планы на один месяц.

## Выбор технологии, языка и среды программирования

Для разработки выбрана программа Android Studio версии 2021.3.1.16, которая является одной из самых популярных интегрированных сред разработки (IDE) для создания мобильных приложений под операционную систему Android. Android Studio предлагает мощный набор инструментов, позволяющих разработчикам создавать нативные приложения, оптимизированные под платформу Android.

Одним из преимуществ Android Studio является наличие встроенного эмулятора, который позволяет проверять корректную работу приложения на различных устройствах с разными экранами и соотношениями сторон. Это значительно упрощает процесс тестирования и обеспечивает совместимость приложения с широким спектром устройств.

Android Studio также предоставляет различные средства для упаковки и маркировки кода, что облегчает процесс разработки и поддержки приложения. Богатый функционал IDE включает в себя удобные инструменты для разработки пользовательского интерфейса, включая функцию Drag-n-Drop, позволяющую легко переносить компоненты в среду разработки.

В качестве языка программирования для разработки приложения выбран Java. Java является основным языком для разработки приложений Android, так как большая часть пользовательского интерфейса операционной системы Android разработана на этом языке. Кроме того, Java обладает высокой производительностью и предоставляет мощные инструменты для разработки и отладки приложений.

Использование Android Studio в связке с Java позволяет создавать качественные, эффективные и масштабируемые приложения для операционной системы Android, с удобным интерфейсом и разнообразным функционалом.

Используемый сервис для базы данных и аутентификации – Firebase. Firebase – платформа специализирующаяся на созданий баз данных для мобильной разработки и веб-приложений. Узкая специализированные данного сервиса, понятность и простота работы с данной платформой позволяет уменьшить затраты времени на разработку и подключение базы данных с разрабатываемым мобильным приложением. Возможности Firebase позволяют реализовать все поставленные задачи.

# Разработка мобильного приложения

## Разработка базы данных

К проекту подключён сервис «Firebase» посредством помощника «Firebase» в программе «Android Studio» и прописывания кода в файлах проекта.

С помощью «Authentication Firebase» была реализована аутентификация пользователей. При регистрации пользователь вводит данные, которые следом заносятся в базу данных. При входе в свой аккаунт, введенные данные пользователя сверяются с базой и при успешной проверке, приложение пропускает пользователя на следующий экран.

В ходе анализа выявлены сущности: пользователи, планы пользователя, ПИН-коды, пользовательские локации, операции пользователя, счета пользователя. Диаграмма базы данных изображена на рисунке 3.1.

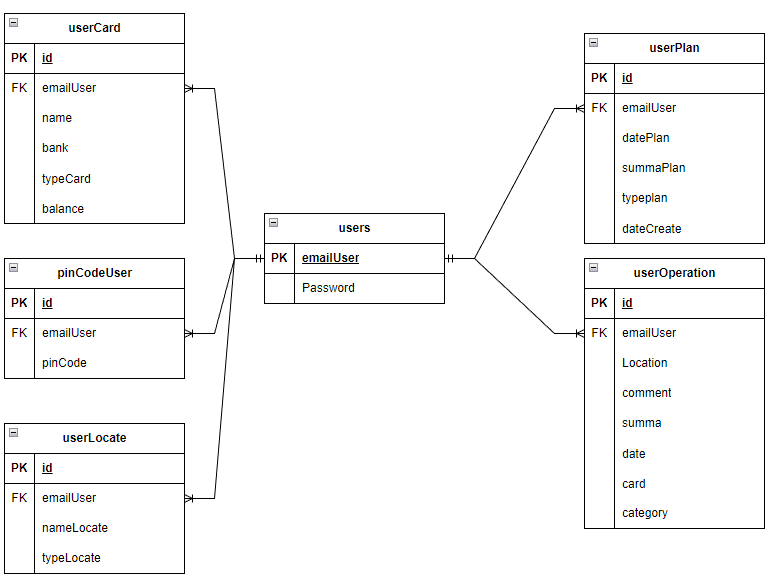


Рисунок 3.1 – ER-диаграмма «Mosticsmo»

Таблица «userPlans» предназначена для хранения планов пользователя на свои денежные средства. Таблица в себя включает:

* дату плана;
* почта пользователя;
* сумма плана;
* дата создания плана;
* тип плана.

Таблица «userPinCode» предназначена для хранения кода доступа пользователя в приложение. Таблица включает в себя почту пользователя и его личный ПИН-код доступа в приложение.

Таблица «userLocate» предназначена для хранения локаций пользователя. Таблица в себя включает:

* почту пользователя;
* название локации;
* тип локации.

Таблица «userOperation» предназначена для хранения операций пользователя. Таблица в себя включает:

* почту пользователя;
* комментарий;
* название локации;
* сумма операции;
* название источника оплаты;
* категорию операции;
* дату проведения операции;
* тип операции.

Таблица «userCard» предназначена для хранения счетов пользователя. Таблица в себя включает:

* почту пользователя;
* банк;
* название счёта;
* сумма счёта;
* вид счёта;
* валюта счёта.

Таблица «Users» хранит в себе почту и пароль пользователей, которая заполняется при каждом новом зарегистрированном пользователе автоматически.

## Описание используемых плагинов

Ниже на рисунке 3.7 изображено название эмулятора для запуска приложения в Android Studio.

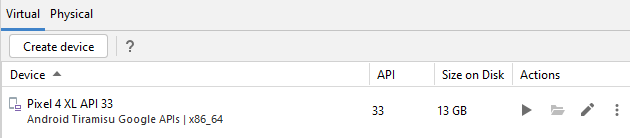


Рисунок 3.7 – Список эмуляторов

Для разработки были использованы встраиваемые плагины, которые создаются вместе с проектом, а также дополнительный плагины, которые нужны для функциональности приложения и изображены на рисунке 3.7.



Рисунок 3.7 – Используемые плагины

## Описание разработанных процедур и функций

В приложении разработаны следующие функции:

* переход между страницами с помощью навигационного меню;
* создание новых записей в базе данных;
* создание диаграммы анализа данных;
* вход по биометрическим данным;
* получение актуальных цен валют.

Ниже на листинг 3.1 представлен метод, который реализует в навигационном меню основной активности переход к фрагментам при помощи класса FragmentManager, в случае выбора пунктов «Счета», «Операции», «Аналитика», «Ещё» и «Планы».

Листинг 3.1 – Метод для реализации навигационного меню

public class MainActivity extends AppCompatActivity {  
  
 ActivityMainBinding binding;  
 FirebaseAuth mAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 binding = ActivityMainBinding.*inflate*(getLayoutInflater());  
 setContentView(binding.getRoot());  
 replaceFragment(new CardFragment());  
 mAuth = FirebaseAuth.*getInstance*();  
  
 binding.bottomNavigationView.setOnItemSelectedListener(item -> {  
 switch (item.getItemId()){  
 case R.id.*bankcards*:  
 replaceFragment(new CardFragment());  
 break;  
 case R.id.*topuppayment*:  
 replaceFragment(new OperationFragment());  
 break;  
 case R.id.*paperwork*:  
 replaceFragment(new PlanFragment());  
 break;  
 case R.id.*graph*:  
 replaceFragment(new AnalysisFragment());  
 break;  
 case R.id.*about*:  
 replaceFragment(new MoreFragment());  
 break;  
 }  
 return true;  
 });  
 }  
 void replaceFragment(Fragment fragment){  
 FragmentManager fragmentManager = getSupportFragmentManager();  
 FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  
 fragmentTransaction.replace(R.id.*frame\_layout*,fragment);  
 fragmentTransaction.commit();  
 }

Ниже на листинг 3.2 представлен класс, который реализует создание новых записей в базе данных. Пользователь вводит данные, и они создаются новой записью в базе данных.

Листинг 3.2 – Класс создания новой записи о счёте

void SaveCard(){  
 *//TODO Сделать добавление нового счёта, сделать документ с логикой для хранения данных для каждого пользователя* String nameTitle = namecard.getText().toString();  
 Map<String, Object> userMoneySave = new HashMap<>();  
 userMoneySave.put("email", user.getEmail());  
 userMoneySave.put("namemoney", nameTitle);  
 userMoneySave.put("typecard", typeCard.getSelectedItem().toString());  
 userMoneySave.put("typevalut", typeV.getSelectedItem().toString());  
 userMoneySave.put("typebank", typeBank.getSelectedItem().toString());  
 userMoneySave.put("scoremoney",ScoreCard.getText().toString());  
  
 db.collection("userScore")  
 .add(userMoneySave)  
 .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {  
 @Override  
 public void onSuccess(DocumentReference documentReference) {  
 Toast.*makeText*(AddNewMoneyCardActivity.this, "Счёт был успешно добавленн.", Toast.*LENGTH\_SHORT*).show();  
 finish();  
 }  
 })  
 .addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Log.*w*(*TAG*, "Error adding document", e);  
 }  
 });  
}

Ниже на листинг 3.3 представлен метод, который отвечает за создание диаграммы анализа данных на основе хранимых данных пользователя в базе данных записывая их в лист в виде объектов AnalyticModel и создание диаграммы.

Листинг 3.3 – Метод для создания диаграммы анализа данных

private void initializeData(){  
 mAuth = FirebaseAuth.*getInstance*();  
 user = mAuth.getCurrentUser();  
 db = FirebaseFirestore.*getInstance*();  
 db.collection("userOperation")  
 .whereEqualTo("email", user.getEmail()).orderBy("yearoperation").orderBy("monthoperation")  
 .get()  
 .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  
 @Override  
 public void onComplete(@NonNull Task<QuerySnapshot> task) {  
 if (task.isSuccessful()) {  
 for (QueryDocumentSnapshot document : task.getResult()) {  
 datePlan = document.getLong("dateoperation");  
 mouthPlan = document.getLong("monthoperation");  
 yearPlan = document.getLong("yearoperation");  
 typePlan = document.getString("typecategoty");  
 SummaPlan = document.getLong("summa");  
  
 plan = new DateModel(mouthPlan, yearPlan);  
 plans.add(new AnalysticModel(datePlan,mouthPlan,yearPlan,SummaPlan,typePlan));  
 if(dates.size()==0){  
 dates.add(plan);  
 }else{  
 int y = 0;  
 for(int i = 0; i<dates.size(); i++){  
 if(dates.get(i).month != plan.month && dates.get(i).year != plan.year){  
 y++;  
 if (y== dates.size()){  
 dates.add(plan);}}}}}} else {  
 Log.*d*("User", "Error getting documents: ", task.getException());  
 }  
 if(dates.size() != 0){  
 ListPlan.setText(MouthSet(dates.get(0).month)+","+dates.get(0).year);  
 SetData(Integer.*parseInt*(String.*valueOf*(dates.get(0).month)), Integer.*parseInt*(String.*valueOf*(dates.get(0).year)));}  
 }  
 });  
}

Ниже на листинг 3.4 представлен метод, который отвечает за вход по биометрическим данным на основе хранимых данных пользователя на его устройстве. При помощи плагина разрешается доступ для получения системных данных и вызова проверки данных сканером.

Листинг 3.4 – Метод для входа по биометрическим данных

BiometricManager biometricManager = BiometricManager.*from*(this);  
 switch (biometricManager.canAuthenticate())  
 {  
 case BiometricManager.*BIOMETRIC\_ERROR\_NO\_HARDWARE*:  
 break;  
 case BiometricManager.*BIOMETRIC\_ERROR\_HW\_UNAVAILABLE*:  
 break;  
 case BiometricManager.*BIOMETRIC\_ERROR\_NONE\_ENROLLED*:  
 break;  
 }  
 Executor executor = ContextCompat.*getMainExecutor*(this);  
 biometricPrompt = new BiometricPrompt(PinCode.this, executor, new BiometricPrompt.AuthenticationCallback() {  
 @Override  
 public void onAuthenticationError(int errorCode, @NonNull CharSequence errString) {  
 super.onAuthenticationError(errorCode, errString);  
 }  
  
 @Override  
 public void onAuthenticationSucceeded(@NonNull BiometricPrompt.AuthenticationResult result) {  
 super.onAuthenticationSucceeded(result);  
 startActivity(new Intent(PinCode.this, MainActivity.class));  
 o++;  
 finish();  
 }  
  
 @Override  
 public void onAuthenticationFailed() {  
 super.onAuthenticationFailed();  
 }  
 });  
  
 promptInfo = new BiometricPrompt.PromptInfo.Builder().setTitle("Приложите палец к сканеру")  
 .setDescription("чтобы разблокировать Mosticsmo").setDeviceCredentialAllowed(true)  
 .build();  
 biometricPrompt.authenticate(promptInfo);  
}

В приложении Б представлен метод (листинг 1), который отвечает за получение актуальных цен валют с сайта по продажам валют путём вытягивания с сайта данных. При помощи плагина для парсинга получаем данные о странице и вытягиваем нужную информации о валютах.

# Тестирование

## Протокол тестирования дизайна приложения

Для проведения тестирования дизайна приложение были выбраны устройства Samsung M21 c разрешением экрана 2340×1080 и Pixel 4 XL c разрешением экрана 3040×1440.

Проверка была проведена на:

* оптимальный размер кнопок;
* читабельный размер шрифта;
* приятное глазу цветовое сочетание;
* корректное расположение элементов.

В результате тестирования дизайна все пункты были пройдены.

## Протокол тестирования функционала приложения

Для тестирования функционала были разработаны тест-кейсы, каждый из которых описывает небольшую последовательность действий с одним конкретным результатом. Благодаря тест-кейсам всегда известно, как и что протестировать оптимальным количеством проверок, и не забыть о нюансах, так как записан каждый шаг. Разработанные тест-кейсы представлены в таблицах от 4.1 до 4.9.

Таблица 4.1 – Test-case 1

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Дымовое тестирование | | |
| Функция: | Запуск приложения | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
|  | |  |  |
| Шаги теста: | |  |  |
| Нажатие на ярлык приложения | | Корректное открытие приложения с выводом разрешений для функционирования | пройден |

Продолжение таблицы 4.1

|  |  |  |
| --- | --- | --- |
| Постусловие: |  |  |
| Продолжение работы | Приложение работает и функционирует | пройден |

Таблица 4.2 – Test-case 2

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Работа навигационного меню | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Нажатие на кнопку меню | | Появление навигационного меню | пройден |
| Шаги теста: | |  |  |
| Нажатие на кнопку меню «Счета» | | Перемещение на страницу счёта | пройден |
| Нажатие на кнопку меню «Операции» | | Перемещение на страницу с операциями | пройден |
| Нажатие на кнопку меню «Планы» | | Перемещение на страницу планов | пройден |
| Нажатие на кнопку меню «Аналитика» | | Перемещение на страницу с аналитикой | пройден |
| Нажатие на кнопку меню «Ещё» | | Перемещение на страницу с дополнительными пунктами | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4.3 – Test-case 3

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Отображение ранее созданных записей счетов | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Нажатие на кнопку «Счета» в навигационном меню | | Перемещение на страницу с счётами | пройден |
| Шаги теста: | |  |  |

Продолжение таблицы 4.3

|  |  |  |
| --- | --- | --- |
| Просмотр списка записей | Отображение сделанных ранее записей экрана | пройден |
| Постусловие: |  |  |
| Продолжение работы | Приложение работает и функционирует | пройден |

Таблица 4.4 – Test-case 4

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Просмотр операций, сделанных ранее | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Нажатие на кнопку «Операции» в навигационном меню | | Перемещение на страницу с записями операций | пройден |
| Шаги теста: | |  |  |
| Нажатие на сделанную ранее запись | | Открытие окна вызова системных приложений. Открытие выбранного приложения с корректным отображением выбранной записи операций | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4.5 – Test-case 5

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Просмотр операций, сделанных ранее | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Нажатие на кнопку «Операции» в навигационном меню | | Перемещение на страницу с записями операций | пройден |
| Шаги теста: | |  |  |

Продолжение таблицы 4.5

|  |  |  |
| --- | --- | --- |
| Нажатие на сделанную ранее запись операции | Открытие окна вызова системных приложений. Открытие выбранного приложения с корректным отображением выбранной записи операций | пройден |
| Постусловие: |  |  |
| Продолжение работы | Приложение работает и функционирует | пройден |

Таблица 4.6 – Test-case 6

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Создание новой операции | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Нажатие на кнопку «Операции» в навигационном меню | | Перемещение на страницу с записями операций | пройден |
| Шаги теста: | |  |  |
| Выбор типа операции «Расход» | | Сохранение данного параметра. | пройден |
| Ввод суммы операции | | Сохранение данного параметра. | пройден |
| Выбор счёта оплаты в поле ввода | | Сохранение данной настройки. | пройден |
| Выбор даты операции | | Сохранение данной настройки. | пройден |
| Выбор места проведения операции | | Сохранение данной настройки | пройден |
| Ввод комментария в поле ввода | | Сохранение данной настройки. | пройден |
| Нажатие кнопки «Добавить» | | Сохранение новой записи в базе данных. | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4.7 – Test-case 7

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Просмотр стоимости валют | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Нажатие на кнопку «Ещё» в навигационном меню | | Перемещение на страницу с дополнительными пунктами | пройден |
| Шаги теста: | |  |  |
| Нажатие на пункт «Курс Валю» | | Открытие окна вызова системных приложений. Открытие окна с валютами. | пройден |
| Ожидание загрузки показателей | | Корректный вывод значений курса валют. | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4.8 – Test-case 8

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Вход по отпечатку пальца | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Шаги теста: | |  |  |
| Выбор способа для входа «Отпечаток» | | Вывод соответствующего виджета для проверки отпечатка пальца. | пройден |
| Прикладывание пальца к сканеру | | Проверка схождение биометрических данных. Переход на основной экран приложения | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Таблица 4.9 – Test-case 9

|  |  |  |  |
| --- | --- | --- | --- |
| Название: | Критическое тестирование | | |
| Функция: | Редактирование планов | | |
| Действие | | Ожидаемый результат | Результат теста:   * пройден * провален * заблокирован |
| Предусловие: | |  |  |
| Запуск приложения | | Корректная работа приложения | пройден |
| Нажатие на кнопку «Планы» в навигационном меню | | Перемещение на страницу с планами | пройден |
| Шаги теста: | |  |  |
| Выбор пункта существующего плана | | Открытие окна вызова системных приложений. Открытие окна данными плана. | пройден |
| Внесение изменений в план | | Отображение введённых данных. | пройден |
| Нажатие на кнопку «Сохранить» | | Сохранение изменений в записи базы данных. | пройден |
| Постусловие: | |  |  |
| Продолжение работы | | Приложение работает и функционирует | пройден |

Заключение

В результате работы над проектом было разработано мобильное приложение, выполняющие роль учёта и анализа домашних финансов, работающего на операционной системе Android.

Данное приложение позволяет осуществить добавление записей операций, добавление планов, добавление счетов. Также данное приложение может анализировать ваши траты и планировать будущие операции, при этом демонстрируя все показатели в графике. К всему этому данное приложение позволяет просматривать актуальный курс валют. И самое главное позволяет сохранять данные на удалённом сервере базы данных, не занимая лишнего пространства на устройстве.

В ходе выполнения проекта были решены следующие задачи:

* проведены исследования и изучены различные методы и инструменты для анализа и учёта данных на устройствах Android. Это позволило определить наиболее эффективные и надежные принципы для работы с денежными средствами, которые могут быть использованы в приложении;
* проведен анализ целевой аудитории приложения. Исследование включало определение потребностей и предпочтений пользователей, их основные характеристики и требования;
* проведен обзор существующих приложений, связанных с финансовым учётом. Этот анализ позволил выявить лучшие практики, улучшить дизайн и функциональность приложения;
* разработана структура приложения, определяющая его основные модули, функциональные блоки и связи между ними. Это позволило логически организовать приложение и обеспечить эффективное взаимодействие между различными его частями;
* проведено изучение методов взаимодействия с удалённой базой данных Firebase, необходимой для сохранения и ведения финансового учёта. Это позволило разработать соответствующую функциональность, чтобы пользователи могли легко управлять своими финансами;
* выполнен этап разработки самого приложения, включая создание кода, реализацию функциональности и интеграцию всех компонентов. Результатом был готовый программный продукт, способный помогать пользователю вести финансовый учёт и анализ.

Во время разработки возникли трудности с взаимодействием базы данных с приложением, которые не давали корректно выводить данные пользователя, а также проблема с вытягиванием цен на валюты. Данные проблемы были решены путем прочтения документации о работе библиотеки для вытаскивания данных из HTML-страницы, а также документации для работы с Firebase и применения теоретических данных практически.

В дальнейшем планируется совершенствовать приложение и добавить такие функции, как автоматический учёт финансов, учёт снятия денежных средств с имеющихся счетов, учёт финансов разными валютами одновременно и отслеживание стоимости валют.

Библиография

Нормативно-правовые акты:

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

Электронные ресурсы:

1. Android Developers [Электронный ресурс]. – Документация Android Studio. – URL: https://developer.android.com/docs/ (дата обращения: 25.11.2022)
2. Java HTML Parser [Электронный ресурс]. – Вытягивание данных с страницы. – URL: https://jsoup.org/ (дата обращения: 10.03.2023)
3. Firebase Documents [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 [Электронный ресурс]. – 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)

# Приложение А

(обязательное)

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С.Галущака»

Техническое задание

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ УЧЁТА И АНАЛИЗА ДОМАШНИХ ФИНАНСОВ

НАТКиГ.201100.010.000ПЗ

Выполнил:

студент группы ПР-20.101

Колпаков Я.А.

2023

**Содержание**

[1 Введение 3](#_Toc120216214)7

[1.1 Наименование программы 3](#_Toc120216215)7

[1.2 Краткая характеристика области применения 3](#_Toc120216216)7

[2 Основание для разработки 3](#_Toc120216217)7

[2.1 Основание для проведения разработки 3](#_Toc120216218)7

[2.2 Наименование и условное обозначение темы разработки 3](#_Toc120216219)7

[3 Назначение разработки 3](#_Toc120216220)7

[3.1 Функциональное назначение 3](#_Toc120216221)7

[3.2 Эксплуатационное назначение 38](#_Toc120216222)

[4 Требования к программе или программному изделию 38](#_Toc120216223)

[4.1 Требования к функциональным характеристикам 38](#_Toc120216224)

[4.2 Требования к надежности. 39](#_Toc120216228)

[4.3 Условия эксплуатации 40](#_Toc120216240)

[4.4 Требования к составу и параметрам технических средств 40](#_Toc120216244)

[4.5 Требования к информационной и программной совместимости 40](#_Toc120216245)

[4.6 Требования к маркировке и упаковке 41](#_Toc120216250)

[4.7 Требования к транспортировке и хранению. 41](#_Toc120216251)

[4.8 Специальные требования 41](#_Toc120216252)

[5 Требования к программной документации 41](#_Toc120216253)

[6 Технико-экономические показатели 42](#_Toc120216255)

[6.1 Ориентировочная экономическая эффективность 42](#_Toc120216256)

[6.2 Предполагаемая потребность 42](#_Toc120216257)

[7 Стадии и этапы разработки 42](#_Toc120216259)

[7.1 Стадии разработки 42](#_Toc120216260)

[7.2 Этапы разработки 42](#_Toc120216261)

[8 Порядок контроля и приемки 43](#_Toc120216262)

[8.1 Виды испытаний 43](#_Toc120216263)

[8.2 Общие требования к приемке работы 43](#_Toc120216264)

# 1 Введение

* 1. **Наименование программы**

Мобильное приложение «Mosticsmo».

**1.2 Краткая характеристика области применения**

Приложение для рядовых пользователей – заинтересованных в учёте и ведении расходов своих личных финансов самостоятельно.

**2** **Основание для разработки**

**2.1 Основание для проведения разработки**

Основанием для проведения разработки является Протокол № Уч-34/1 от 7 марта 2023 года.

**2.2 Наименование и условное обозначение темы разработки**

Наименование темы разработки – «Разработка мобильного приложения для учета домашних финансов».

**3** **Назначение разработки**

**3.1 Функциональное назначение**

Функциональным назначением программы является предоставление пользователю вести свои расходы, планировать их, вести отчетность по тратам в различных категориях, отслеживать курс валют, получать доступ к приложению с любого устройства, где установлено данное приложение.

**3.2 Эксплуатационное назначение**

Программа должна использоваться на устройствах с мобильной операционной системой Android. Конечными пользователями программы могут являться владельцы смартфонов, интересующиеся функционалом своего устройства.

**4** **Требования к программе или программному изделию**

**4.1** **Требования к функциональным характеристикам**

**4.1.1 Требования к составу выполняемых функций**

Приложение должно обеспечивать возможность выполнения перечисленных ниже функций:

* учет расходов и доходов;
* анализ финансов;
* хранение информации об счетах пользователя;
* организация хранения данных пользователя удалённо.

Для получения результата будет необходимо воспользоваться приложением для ознакомления с представленным функционалом

**4.1.2 Требования к организации входных и выходных данных**

Отсутствуют, пользователь вправе вводить необходимую для него информацию.

Данные пользователя должны хранятся на удалённом сервере базы данных.

**4.1.3 Требования к временным характеристикам**

Требования к временным характеристикам определяются относительно памяти устройства.

**4.2 Требования к надежности**

**4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы**

Надёжность программного средства должна соответствовать следующим требованиям:

* использование лицензионного программного обеспечения;
* организация бесперебойного питания оборудования пользователя;
* защита от ошибочных действий пользователей на уровне пользовательского интерфейса;
* обеспечение обнаружения и диагностики ошибок с выдачей соответствующих сообщений пользователю на уровне пользовательского интерфейса.

**4.2.2 Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

**4.2.3 Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

**4.3 Условия эксплуатации**

**4.3.1 Климатические условия эксплуатации**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

**4.3.2 Требования к численности и квалификации персонала**

Количество персонала, требуемого для работы программы, должно составлять одну штатную единицу – конечный пользователь программы (оператор). Конечный пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

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

Для работы приложения на устройстве требуется: объем свободной памяти не менее 3 ГБ, версия устройства Android 10 и выше.

**4.5 Требования к информационной и программной совместимости**

**4.5.1 Требования к исходным кодам и языкам программирования**

Исходные коды программы должны быть реализованы на языке Java. В качестве интегрированной среды разработки программы должна быть использована среда Android Studio Dolphin 2021.3.1.

**4.5.2 Требования к программным средствам, используемым программой**

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы.

**4.5.3 Требования к защите информации и программ**

Требования к защите информации и программ не предъявляются.

**4.6 Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

**4.7 Требования к транспортировке и хранению.**

Проект приложения поставляется apk файлом.

Программная документация поставляется в электронном и   
печатном виде.

**4.8 Специальные требования**

Программное обеспечение должно иметь дружественный интерфейс, который должен быть защищен от неправильных действий пользователей и снабжен системой подсказок.

**5** **Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание;
* текст программы;
* описание программы;
* пояснительная записка;

# 6 Технико-экономические показатели

## 6.1 Ориентировочная экономическая эффективность

Ориентировочная экономическая эффективность не рассчитывается.

## 6.2 Предполагаемая потребность

Предполагаемая потребность обусловливается тем фактом, что функция захвата и записи видео с экрана является популярной и помогает пользователю облегчить взаимодействие с устройством.

**7** **Стадии и этапы разработки**

**7.1 Стадии разработки**

Разработка должна быть проведена в три стадии:

1. разработка технического задания;
2. разработка пояснительной записки;
3. рабочее проектирование;
4. защита.

## 7.2 Этапы разработки

Этапы разработки представлены в таблице 1.

Таблица 7.1 — Стадии разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название этапа | Срок, даты | Отчётность |
| 1 | 2 | 3 | 4 |
| 1 | Исследование предметной области | 25.03.2023-28.03.2023 | Пояснительная записка |
| 2 | Выбор моделей, описывающих предметную область | 01.04.2023-03.04.2023 | Пояснительная записка |
| 3 | Разработка технического задания | 03.04.2023-05.04.2023 | Техническое задание |
| 4 | Изучение Java | 01.04.2023-  18.04.2023 | Пояснительная записка |

Продолжение таблицы 7.1

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Анализ требований и уточнение спецификаций | 14.04.2023 | Спецификации программного обеспечения |
| 6 | Проектирование структуры мобильного приложения, проектирование компонентов (технический проект) | 15.04.2023-29.04.2023 | Схема структурная системы и спецификации компонентов |
| 7 | Кодирование клиентской части | 01.05.2023-20.06.2023 | Программный  Продукт |
| 8 | Тестирование компонентов.  Сборка и комплексное тестирование | 20.06.2023 | Тексты программных компонентов |
| 9 | Разработка программной документации | 20.06.2023 – 21.06.2023 | Программная  документация |
| 10 | Защита | 26.06.2023 |  |

**8** **Порядок контроля и приемки**

**8.1 Виды испытаний**

Видом испытания является защита курсового проекта.

**8.2 Общие требования к приемке работы**

Общими требованиями к приёмке являются:

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

# Приложение Б

Листинг 1 – Метод для получения актуальных цен валют

private void GetWeb(){  
 try {  
 doc = Jsoup.*connect*("https://novosibirsk.bankiros.ru/currency").get();  
 Elements tables = doc.getElementsByTag("tbody");  
 *//Выбор таблицы* Element our\_table = tables.get(0);  
 Elements elements\_from\_tavble = our\_table.children();  
 *//Выбор элемента строки* 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();  
 }