Министерство образования Новосибирской области

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

разработка мобильного приложения для определения и фикСации промежутков времени (таймер)

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.200800.010.000ПЗ

Разработал:

студент группы ПР-21.106

Поплавский Д.В.

2023

**Содержание**

[Введение 3](#_Toc136026799)

[1 Исследовательский раздел 4](#_Toc136026800)

[1.1 Описание предметной области 4](#_Toc136026801)

[1.2 Образ клиента 4](#_Toc136026802)

[1.3 Сценарии 4](#_Toc136026803)

[1.4 Сбор и анализ прототипов 5](#_Toc136026804)

[2 Проектирование приложения 13](#_Toc136026805)

[2.1 UI/UX дизайн проекта 13](#_Toc136026806)

[2.2 Выбор технологии, языка и среды программирования 15](#_Toc136026807)

[3 Разработка мобильного приложения 16](#_Toc136026808)

[3.1 Выбор используемого эмулятора 16](#_Toc136026809)

[3.2 Описание разработанных процедур и функций 16](#_Toc136026810)

[Тестирование 20](#_Toc136026811)

[3.3 Протокол тестирования дизайна приложения 20](#_Toc136026812)

[3.4 Протокол тестирования функционала приложения 21](#_Toc136026813)

[Заключение 25](#_Toc136026814)

[Библиография 26](#_Toc136026815)

[Приложение А (обязательное) Техническое задание 27](#_Toc136026816)

Введение

В наше время существует немало различных видов деятельности, в которых требуется точное измерение времени и его запись. Одной из таких деятельностей является спорт, в котором многое завязано на времени. На профессиональных соревнованиях, чтобы определить победителя в какой-либо спортивной дисциплине, часто требуется точно замерять время, особенно, если счёт идёт на доли секунд. Иной подобный вид деятельности – занятия физической культурой в образовательных учреждениях (школах, колледжах, университетах и т.д.). Там преподавателям физической культуры во время сдачи студентами нормативов необходимо измерять это время, чтобы затем выставить оценки, базируясь на критериях времени. Но не только в физическом спорте требуется замер времени. Он ещё необходим и в спорте интеллектуальном, например, в шахматах. В партиях на время оба соперника этим временем ограничены и, чтобы успевать обдумывать и совершать ходы, им нужно замерять время, потраченное на ход.

Целью курсового проекта является разработка мобильного приложения, имеющего функционал таймера.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области темы;
* рассмотрение приложения с точки зрения пользователя для выявления необходимых функций приложения;
* разработка мобильного приложения.

Объект исследования – приложения-таймеры.

Предмет исследования – изучение принципов функционирования и инструментов приложения.

# Исследовательский раздел

## Описание предметной области

В настоящее время существует множество видов деятельности, где необходимо точно измерять время и записывать его. Один из таких видов - спорт, где время играет важную роль. На профессиональных соревнованиях необходимо точно замерять время, чтобы определить победителя в спортивной дисциплине, особенно, если счет идет на доли секунд. Также в занятиях физической культурой в образовательных учреждениях преподавателям необходимо измерять время, чтобы оценить студентов, исходя из критериев времени. Однако, не только в физическом спорте требуется замер времени, он также необходим в интеллектуальном спорте, например, в шахматах, где оба соперника ограничены временем на ходы.

## Образ клиента

Клиентами являются люди, чья деятельность требует точного измерения времени.

Существует несколько видов деятельности, в которых требуется указание определённого промежутка времени. К ним относятся спортсмены, тренеры, преподаватели физической культуры, повара и даже шахматисты.

Средний возраст клиента может разниться в зависимости от рода деятельности. Так, у спортсменов он составляет 20-30 лет, у преподавателей физической культуры – 40-45 лет, у поваров – 30-35 лет.

## Сценарии

Прежде всего, при составлении сценария выявляется цель, которой будут достигать пользователи.

Девушка решила заняться тренировкой выносливости. Чтобы тренировки имели более высокую эффективность, ей необходимо во время выполнения упражнений ставить необходимый промежуток времени. Для этого ей достаточно скачать приложение и воспользоваться таймером, благодаря которому она может ставить на обратный отсчёт отрезки времени и, таким образом, регулировать нагрузку во время тренировок.

С другой стороны, таймер окажется полезным для человека, занимающегося приготовлением. Он может выставить таймер на нужное ему время, и в течение этого времени готовить какую-либо часть блюда. Таймер позволит ему приготовить блюдо вовремя, без риска получить недоготовленное или передержанное блюдо.

## Сбор и анализ прототипов

У данного мобильного приложения имеется множество конкурентов, так как приложения, подобные проекту, легко разработать, и они, благодаря своей простоте, пользуются большим спросом.

Рассмотрим дизайн и функционал приложения, имеющегося в Google Play – Stopwatch.

Stopwatch имеет нейтральное цветовое решение приложения, преобладают белый и синий цвета. На всех экранах отсутствует фоновое изображение.

Далее приведён рисунок 1 с изображением страницы «Секундомер».

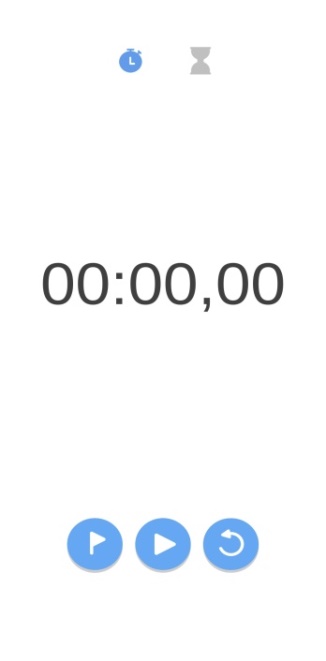


Рисунок 1 – Страница «Секундомер»

На странице есть шапка, информационная часть и функциональное меню в нижней части экрана. Шапку приложения составляют кнопки перехода на таймер/секундомер. В функциональном меню имеются кнопки, выполняющие функции записи секундомера, его запуска и сброса соответственно. Информационная часть состоит из секундомера.

На рисунке 2 представлена страница «Таймер».

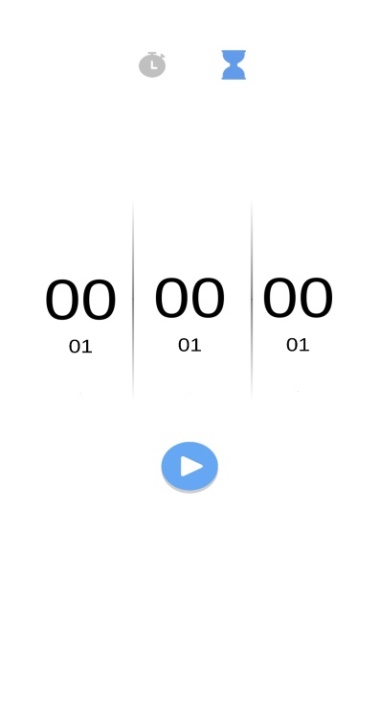


Рисунок 2 – Страница «Таймер»

На данной странице имеются те же основные элементы, что и на предыдущей странице, а именно шапка, информационная часть и функциональное меню. Шапку приложения составляют кнопки перехода на таймер/секундомер. В функциональном меню имеется одна кнопка, запускающая таймер. Информационная часть состоит из элемента выбора времени для таймера.

На рисунке 3 представлена страница «Отсчёт таймера».

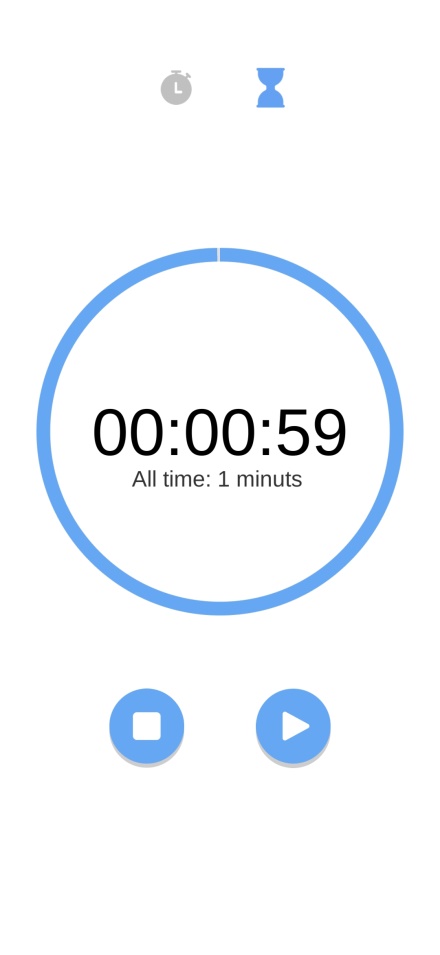


Рисунок 3 – Страница «Отсчёт таймера»

Приложение переходит на эту страницу при запуске таймера. Как и на предыдущих страницах, эта имеет элементы в виде шапки, информационной части и функционального меню. Шапку приложения составляют кнопки перехода на таймер/секундомер. В функциональном меню имеются кнопки, останавливающие и запускающие таймер. Информационная часть состоит из кругового элемента, визуализирующего отсчёт времени, и таймера внутри круга.

Исходя из рассмотрения данного приложения, можно сделать вывод:

Плюсы приложения:

* оснащенность главными функциями;
* цветовая гамма состоит из двух основных цветов.

Минусы приложения:

* отсутствует возможность просмотра сохранённых записей секундомера.

Рассмотрим дизайн и функционал второго приложения, имеющегося в Google Play – Stopwatch Timer Original.

Stopwatch Timer Original имеет тёмное цветовое решение приложения, преобладают тёмно-синий, белый и синий цвета. На всех экранах отсутствует фоновое изображение.

На рисунке 4 представлена страница «Секундомер».

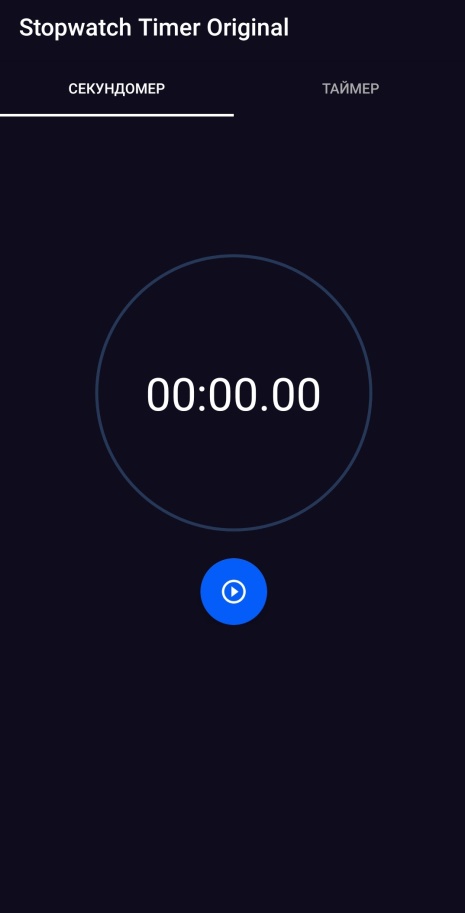


Рисунок 4 – Страница «Секундомер»

На странице есть шапка и информационная часть вместе с функциональным меню по центру экрана. Шапку приложения составляют заголовок с названием приложения, а также плашки перехода на таймер/секундомер. Информационная часть состоит секундомера внутри круга, а под ней расположена кнопка запуска секундомера.

На рисунке 5 изображена страница «Таймер».

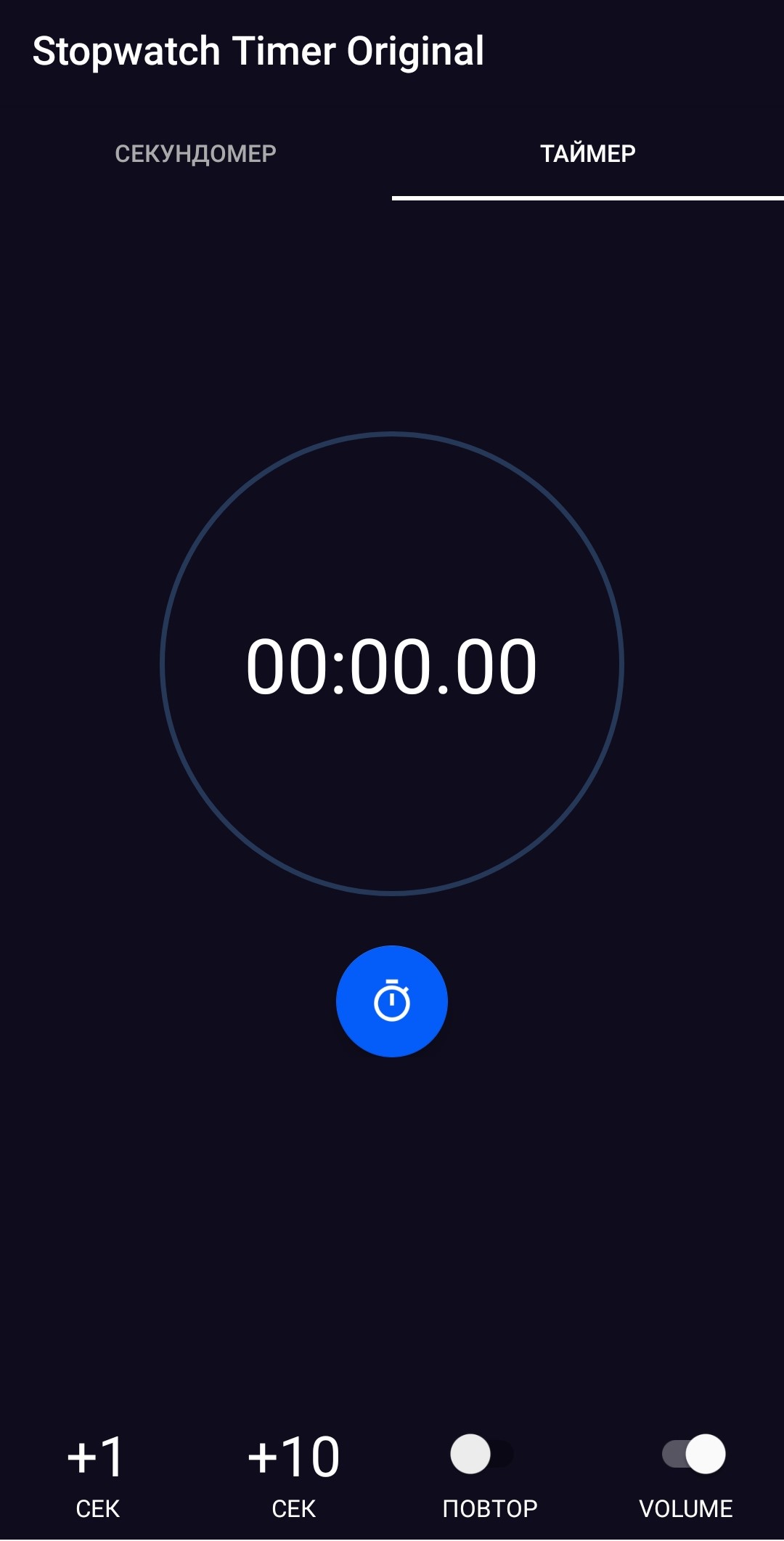


Рисунок 5 – Страница «Таймер»

На этой странице имеются те же элементы, что и на странице «Секундомер», только к ним добавилось ещё и функциональное меню внизу экрана, состоящее из кнопок добавления времени к таймеру (+1 секунда, +10 секунд), кнопки повтора таймера при его окончании и кнопка включения звукового сигнала при окончании таймера.

На рисунке 6 представлено всплывающее окно установки времени таймера.

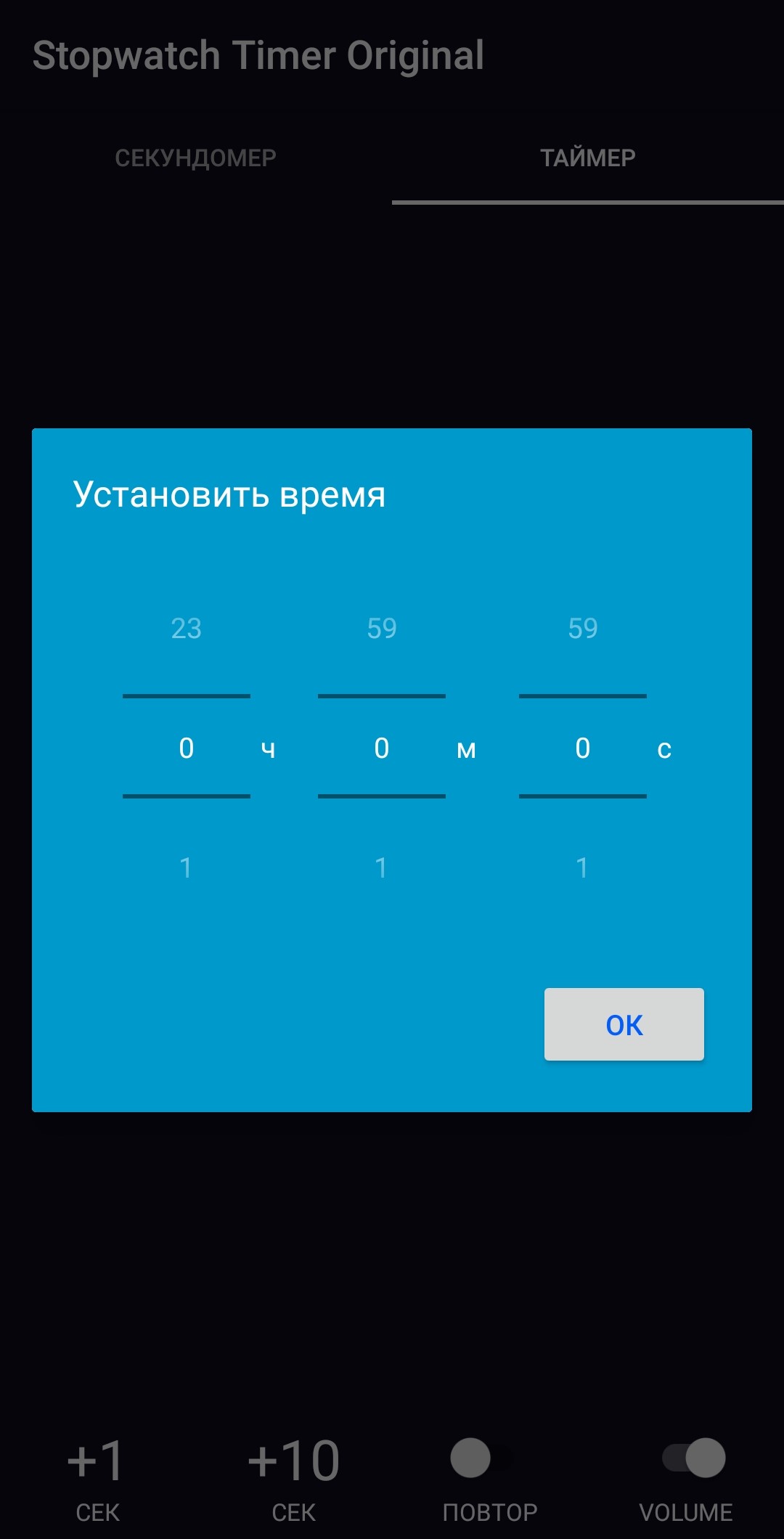


Рисунок 6 – Окно добавления времени к таймеру

Исходя из рассмотрения данного приложения, можно сделать вывод:

Плюсы приложения:

* оснащенность главными функциями;
* цветовая гамма состоит из трёх основных цветов;
* возможность включить/отключить звуковой сигнал.

Минусы приложения:

* слишком простой и минимальный дизайн.

Рассмотрим дизайн и функционал третьего приложения, имеющегося в Google Play – «Секундомер».

Цветовое решение приложения «Секундомер» состоит из белого и насыщенного синего цветов. На всех экранах отсутствует фоновое изображение.

На рисунке 7 изображена страница «Секундомер».

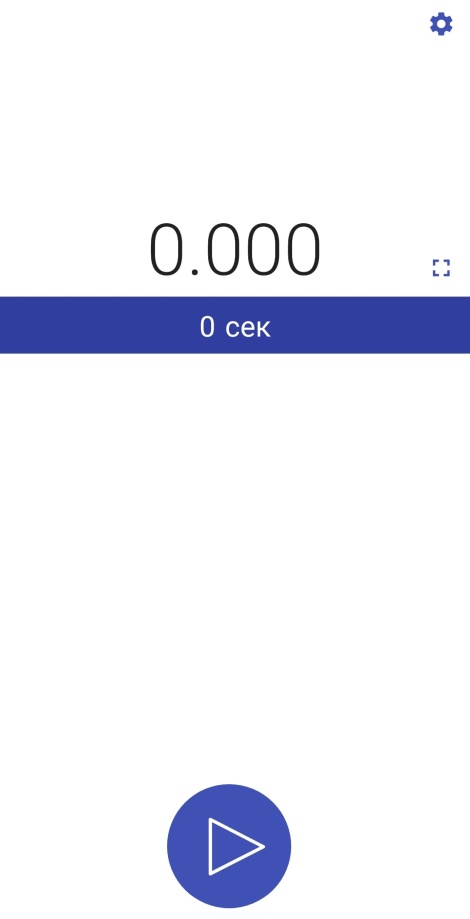


Рисунок 7 – Страница «Секундомер»

На этой странице расположены функциональные меню в верхней и нижней части экрана, а также информационная часть по центру экрана. Верхнее функциональное меню имеет кнопку перехода к настройкам приложения, нижнее – кнопку запуска секундомера. В правом углу информационной части расположена кнопка перехода приложения в альбомный режим.

На рисунке 8 представлена страница «Настройки».

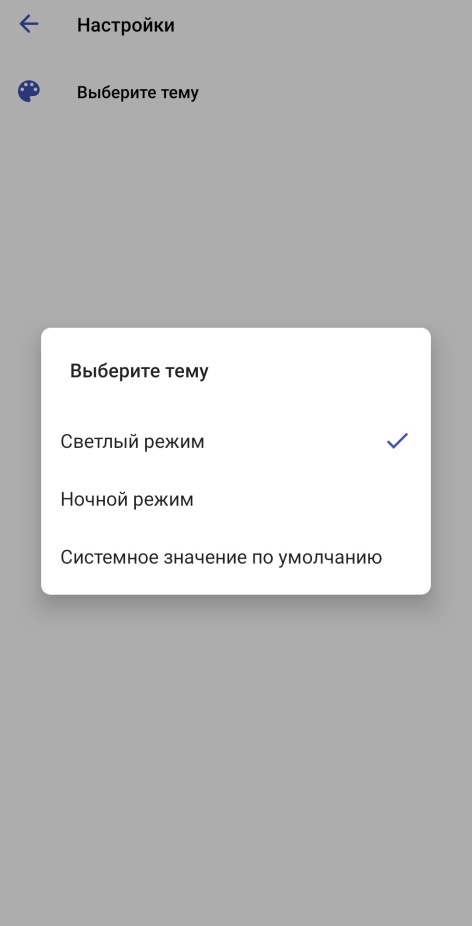


Рисунок 8 – Страница «Настройки»

Данная страница имеет только одну плашку, при нажатии на которую появляется всплывающее окно с опцией выбора темы приложения.

Исходя из рассмотрения данного приложения, можно сделать вывод:

Плюсы приложения:

* оснащенность основными функциями;
* цветовая гамма состоит из двух основных цветов;
* возможность изменить тему приложения.

Минусы приложения:

* отсутствие таймера.

В заключении трендами в разработке являются: запуск секундомера, возможность сохранять время, отсчитанное секундомером, выбор времени для таймера, запуск таймера.

# Проектирование приложения

## UI/UX дизайн проекта

Дизайн проекта разработан в программе Figma.

Для проекта определены экраны:

* стартовый экран;
* таймер;
* экран выбора времени таймера.

Цветовая гамма выбрана посредством изучения информации о восприятии цветов. Человеческий глаз лучше воспринимает спокойные цвета белых, синих, голубых оттенков. Приложение не должно раздражать яркими цветами, общая гамма должна быть гармоничной.

Потому принято решение в разработке макета отдать предпочтение спокойным цветам: циановый и белый. Стоит заметить, что данные цвета будут выглядеть лаконично.

На рисунке 9 представлена страница таймера.

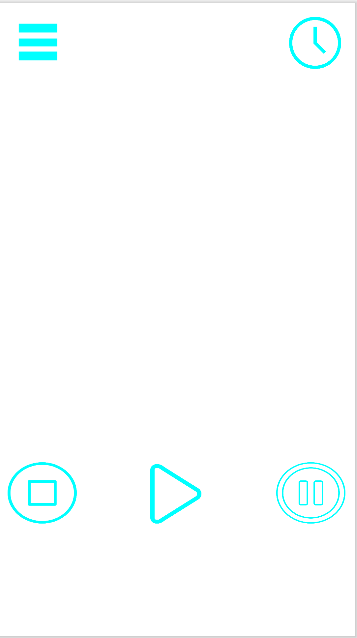


Рисунок 9 – Страница «Таймер»

На рисунке 10 представлена страница выбора времени.



Рисунок 10 – Страница «Выбор времени»

На рисунке 11 представлена страница вывода логов.

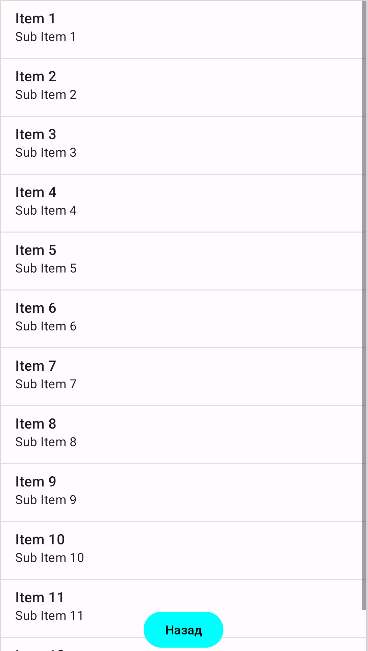


Рисунок 11 – Страница «Логи»

## Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Android Studio. Данная среда программирования выбрана для разработки приложения, так как Android Studio является удобным для разработки проектов, он имеет широкий функционал и инструментарий. Языком программирования выбран язык Java, так как данный язык очень популярен в разработке мобильных приложений. Благодаря своему обширному выбору библиотек, а также понятному синтаксису Java имеет более высокий приоритет, чем другие языки программирования. Также можно отметить, что значительно легче разрабатывать мобильные приложения именно на языке Java, чем на других языках программирования.

Чтобы добавить в проект базу данных и расширить функционал приложения, решено использовать систему управления базами данных SQLite. Данный сервис был выбран по нескольким причинам:

* возможность легко интегрировать сервис в проект;
* доступный набор библиотек для управления базой данных внутри кода;
* высокий отклик базы данных;
* обновление данных без задержек.

Таким образом, выбранный инструментарий облегчит разработку программного продукта и обеспечит его высокое качество.

# Разработка мобильного приложения

## Выбор используемого эмулятора

На рисунке 12 представлено название эмулятора для запуска приложения в Android Studio.

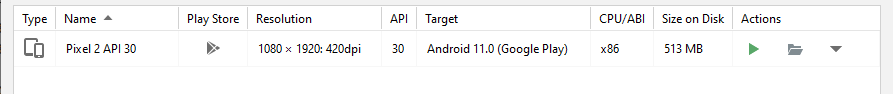


Рисунок 12 – Эмулятор

## Описание разработанных процедур и функций

В приложении разработаны следующие функции:

* выбор времени таймера;
* запуск таймера;
* приостановка таймера;
* остановка и сброс таймера.
* запись логов и их вывод;
* просмотр логов.

В листинге 1 представлен метод setTime, который отвечает за установку введённого времени.

Листинг 1 – метод setTime

public void setTime(View view) {  
 String timeString = timeEditText.getText().toString();  
 Intent intent = new Intent();  
 intent.putExtra("time", timeString);  
 setResult(*RESULT\_OK*, intent);  
 finish();  
}

В листинге 2 представлен метод startTimer, который отвечает за старт отсчёта введённого времени, а также за передачу его в логи.

Листинг 2 – метод startTimer

public void startTimer(View view) {  
 if (isRunning) {  
 *// Таймер уже запущен, нет необходимости второй раз* return;  
 }  
  
 String timeString = timeEditText.getText().toString();  
 if (timeString.isEmpty()) {  
 *// Проверка на пустой ввод* Toast.*makeText*(this, "Введите время", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 long setTimeInMillis;  
 try {  
 int time = Integer.*parseInt*(timeString);  
 if (time <= 0) {  
 *// Проверка на недопустимое значение времени* Toast.*makeText*(this, "Введите положительное значение времени", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 setTimeInMillis = time \* 1000;  
 } catch (NumberFormatException e) {  
 *// Проверка на некорректный формат ввода* Toast.*makeText*(this, "Введите корректное значение времени", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 if (isPaused) {  
 timeLeftInMillis = pauseOffsetInMillis;  
 isPaused = false;  
 } else {  
 timeLeftInMillis = setTimeInMillis;  
 *//Записываем время в БД* TimerHistoryActivity.TimerDatabaseHelper dbHelper = new TimerHistoryActivity.TimerDatabaseHelper(this);  
 dbHelper.insertTimer(timeLeftInMillis);  
 }  
  
 isRunning = true;  
  
 countDownTimer = new CountDownTimer(timeLeftInMillis, 1000) {  
 private long startTimeInMillis;  
  
 @Override  
 public void onTick(long millisUntilFinished) {  
 timeLeftInMillis = millisUntilFinished;  
  
 long elapsedTimeInMillis = System.*currentTimeMillis*() - startTimeInMillis;  
 long timeLeftPlusDelayInMillis = timeLeftInMillis + 1000;  
 long timeLeftPlusDelayAndElapsedTimeInMillis = timeLeftPlusDelayInMillis + elapsedTimeInMillis;  
  
 if (timeLeftPlusDelayAndElapsedTimeInMillis <= 0) {  
 timeLeftInMillis = 0;  
 onFinish();  
 } else {  
 updateTimerText();  
 }  
 }  
  
 @Override  
 public void onFinish() {  
 isRunning = false;  
 mediaPlayer.start();  
 updateTimerText();  
 }  
 }.start();  
}

В листинге 3 представлен метод pauseTimer, который приостанавливает обратный отсчёт таймера.

Листинг 3 – метод startTimer

public void pauseTimer(View view) {  
 if (!isRunning) {  
 *// Таймер уже остановлен* return;  
 }  
  
 countDownTimer.cancel();  
 pauseOffsetInMillis = timeLeftInMillis;  
 isPaused = true;  
 isRunning = false;  
}

В листинге 4 представлен метод resetTimer, который останавливает обратный отсчёт таймера и сбрасывает его на нулевое значение.

Листинг 4 – метод resetTimer

public void resetTimer(View view) {  
 if (isRunning || isPaused) {  
 countDownTimer.cancel();  
 isRunning = false;  
 isPaused = false;  
 }  
  
 timeLeftInMillis = 0;  
 pauseOffsetInMillis = 0;  
 updateTimerText();  
 mediaPlayer.stop();  
 mediaPlayer.prepareAsync();  
}

В листинге 5 представлен класс TimerDatabaseHelper, создающий базу данных SQLite. Он включает в себя метод onCreate, в котором создаётся таблица timer\_history для записи логов, метод getAllTimers для получения значений времени, установленных пользователем, а также метод insertTimer, записывающий значение времени в таблицу timer\_history.

Листинг 5 – класс TimerDatabaseHelper

public static class TimerDatabaseHelper extends SQLiteOpenHelper {  
 private static final String *DATABASE\_NAME* = "timer\_history.db";  
 private static final int *DATABASE\_VERSION* = 1;  
  
 public TimerDatabaseHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL("CREATE TABLE timer\_history (\_id INTEGER PRIMARY KEY AUTOINCREMENT, time INTEGER);");  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 *// ...* }  
  
 public void insertTimer(long timeInMillis) {  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put("time", timeInMillis/1000);  
 db.insert("timer\_history", null, values);  
 db.close();  
 }  
  
 public Cursor getAllTimers() {  
 SQLiteDatabase db = getReadableDatabase();  
 return db.rawQuery("SELECT \* FROM timer\_history", null);  
 }  
}

В листинге 6 представлен метод onCreate класса TimerHistoryActivity, в котором производится вывод логов в ListView.

Листинг 6 – метод onCreate класса TimerHistoryActivity

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_timer\_history*);  
  
 Button backButton = findViewById(R.id.*backButton*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 timerHistoryListView = findViewById(R.id.*timerHistoryListView*);  
 dbHelper = new TimerDatabaseHelper(this);  
  
 Cursor cursor = dbHelper.getAllTimers();  
 String[] fromColumns = {"time"};  
 int[] toViews = {android.R.id.*text1*};  
 SimpleCursorAdapter adapter = new SimpleCursorAdapter(this, android.R.layout.*simple\_list\_item\_1*, cursor, fromColumns, toViews, 0);  
 timerHistoryListView.setAdapter(adapter);  
}

# Тестирование

## Протокол тестирования дизайна приложения

Для проведения тестирования дизайна приложения выбрано устройство Xiaomi Redmi Note 7 c разрешением экрана 2340×1080.

На рисунке 13, представлено приложение на устройстве Xiaomi Redmi Note 7.

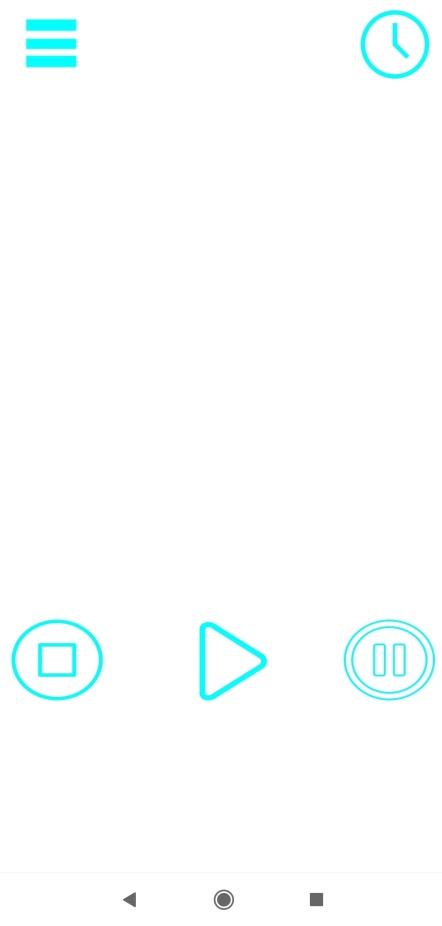


Рисунок 13 – приложение на устройстве Xiaomi Redmi Note 7

Проверка проведена на:

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

## Протокол тестирования функционала приложения

Для тестирования функционала разработаны TestCase.

Test-Case №1

Название тест кейса: Выбор времени для таймера

Предусловия тест кейса: приложение открыто на экране выбора времени таймера

Шаги тест кейса:

* + шаг тест кейса №1: нажать на панель ввода текста с текстовой подсказкой «Время в секундах»;
  + шаг тест кейса №2: ввести любое не отрицательное число;
  + шаг тест кейса №3: нажать на кнопку «ОК»;
  + шаг тест кейс №4: нажать на кнопку «Старт».

Ожидаемый результат тест кейса:

Программа отловит указанное пользователем время и запишет его в TextView, который при нажатии кнопки «Старт» отобразится на экране и в нём начнёт идти обратный отсчёт в секундах в том количестве, которое указал пользователь.

Фактический результат тест кейса:

Программа отлавливает указанное пользователем время, записывает его в TextView, тот при нажатии кнопки «Старт» отображается на экране и в нём идёт обратный отсчёт в секундах в том количестве, которое указал пользователь.

Статус кейса: выполнено.

Test-Case №2

Название тест кейса: Проверка на ввод пустого значения

Предусловия тест кейса: приложение открыто

Шаги тест кейса:

* + шаг тест кейса №1: нажать на панель ввода текста с текстовой подсказкой «Введите время в секундах»;
  + шаг тест кейса №2: не вводить никакое число, а сразу закрыть цифровую панель ввода;
  + шаг тест кейса №3: нажать на кнопку «ОК»;
  + шаг тест кейс №4: нажать на кнопку «Старт».

Ожидаемый результат тест кейса:

После нажатия пользователем кнопки «Старт» TextView не отобразится на экране и отсчёт не пойдёт, а вместо этого программа выдаст всплывающее сообщение «Введите время».

Фактический результат тест кейса:

После нажатия пользователем кнопки «Старт» TextView не отображается на экране и отсчёт не идёт, программа выдаёт всплывающее сообщение «Введите время».

Статус кейса: выполнено.

Test-Case №3

Название тест кейса: Проверка на ввод отрицательного значения

Предусловия тест кейса: приложение открыто

Шаги тест кейса:

* + шаг тест кейса №1: нажать на панель ввода текста с текстовой подсказкой «Введите время в секундах»;
  + шаг тест кейса №2: ввести любое отрицательное число;
  + шаг тест кейса №3: нажать на кнопку «ОК»;
  + шаг тест кейс №4: нажать на кнопку «Старт».

Ожидаемый результат тест кейса:

После нажатия пользователем кнопки «Старт» TextView не отобразится на экране и отсчёт не пойдёт, а вместо этого программа выдаст всплывающее сообщение «Введите положительное значение времени».

Фактический результат тест кейса:

Цифровая панель ввода не даёт пользователю возможность ввести отрицательное значение, т.к. при нажатии на кнопку «-» ничего не происходит.

Статус кейса: провалено.

Test-Case №4

Название тест кейса: Проверка функционала

Предусловия тест кейса: выполнен Test-Case №1

Шаги тест кейса:

* + шаг тест кейса №1: нажать на кнопку «Пауза»;
  + шаг тест кейса №2: нажать на кнопку «Старт»;
  + шаг тест кейса №3: нажать на кнопку «Сброс».

Ожидаемый результат тест кейса:

После нажатия пользователем кнопки «Пауза» отсчёт времени в TextView остановится; после нажатия кнопки «Старт» обратный отсчёт продолжится с той секунды, на которой был остановлен. Затем, после нажатия кнопки «Сброс» отсчёт в TextView обнулится и перестанет идти.

Фактический результат тест кейса:

После нажатия пользователем кнопки «Пауза» отсчёт времени в TextView останавливается; после нажатия кнопки «Старт» обратный отсчёт продолжается с той секунды, на которой был остановлен. Затем, после нажатия кнопки «Сброс» отсчёт в TextView обнуляется и перестаёт идти.

Статус кейса: выполнено.

Test-Case №5

Название тест кейса: Проверка функционала звукового сигнала

Предусловия тест кейса: выполнен Test-Case №1, звук в качельке громкости смартфона увеличен на несколько пунктов

Шаги тест кейса:

* + шаг тест кейса №1: дождаться окончания обратного отсчёта.

Ожидаемый результат тест кейса:

После окончания обратного отсчёта в TextView программа воспроизведёт звуковой сигнал, уведомляя тем самым пользователя об окончании обратного отсчёта.

Фактический результат тест кейса:

После окончания обратного отсчёта в TextView программа воспроизводит звуковой сигнал, уведомляя тем самым пользователя об окончании обратного отсчёта.

Статус кейса: выполнено.

Test-Case №6

Название тест кейса: Проверка вывода логов таймера

Предусловия тест кейса: выполнен Test-Case №1

Шаги тест кейса:

* + шаг тест кейса №1: нажать на кнопку «Логи»;

Ожидаемый результат тест кейса:

После нажатия пользователем кнопки «Логи» приложение перейдёт на экран логов, куда выводятся все значения времени, установленные пользователем до этого.

Фактический результат тест кейса:

После нажатия пользователем кнопки «Логи» приложение переходит на экран логов. Все логи корректно выводятся в ListView, определённый на этом экране.

Статус кейса: выполнено.

Заключение

В ходе работы исследована предметная область, а также прототип приложения. В результате работы разработано мобильное приложение для определения и фиксации промежутков времени (таймер).

В мобильном приложении реализованы следующие функции:

* + установка значения времени для таймера;
  + запуск таймера;
  + приостановка таймера;
  + сброс таймера с остановкой.
  + запись логов и их вывод;
  + просмотр логов.

В ходе работы выполнены следующие поставленные задачи:

* исследовать предметную область проекта;
* определить функционал приложения;
* разработать дизайн приложения;
* разработать прототип приложения;
* реализовать функционал приложения;
* провести тестирование приложения.

В дальнейшем планируется совершенствовать приложение и к имеющемуся функционалу добавить новый.

Библиография

1. WordsOfWonders [электронный ресурс]: Опрос по предметной области: https://wordsofwonders.ru/est-vopros-100-k-1-komu-nuzhen-sekundomer/
2. Android Developers [электронный ресурс]: Документация Android Studio: https://developer.android.com/docs/
3. Help Center [электронный ресурс]: Документация Java: <https://docs.oracle.com/en/java/>
4. SQLite [электронный ресурс]: Документация SQLite: https://www.sqlite.org/docs.html
5. StartAndroid [электронный ресурс]: Уроки по Android: https://startandroid.ru/ru/uroki/vse-uroki-spiskom.html
6. Git [электронный ресурс]: Документация Git: https://git-scm.com/
7. Figma [электронный ресурс]: Прототипирование проектов: https://www.figma.com/

Приложение А

Техническое задание

Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж

имени Б.С. Галущака»

разработка мобильного приложения для определения и фикСации промежутков времени (таймер)

НАТКиГ.200800.010.000ПЗ

Выполнил:

студент группы

ПР-21.106

Поплавский Д.В.

2023

**Содержание**

[Введение 25](#_Toc136026799)

[1 Назначение разработки 26](#_Toc136026800)

[2 Требования к мобильному приложению 27](#_Toc136026805)

[2.1 Требования к функциональным характеристикам 27](#_Toc136026806)

[2.2 Требования к надёжности 28](#_Toc136026807)

[2.3 Требования к составу и параметрам технических средств 2](#_Toc136026807)8

[2.4 Требования к информационной и программной совместимости 2](#_Toc136026807)8

[2.5 Требования к маркировке и упаковке 2](#_Toc136026807)8

[3 Требования к программной документации 2](#_Toc136026808)8

[4 Технико-экономические показатели 2](#_Toc136026808)8

[5 Стадии и этапы разработки 2](#_Toc136026808)9

6 Порядок контроля и приёмки 30

**Введение**

Настоящее техническое задание распространяется на разработку информационной системы «Разработка мобильного приложения для определения и фиксации промежутков времени (таймер)», используемого для установки таймера на определённое время.

Наименование приложения: «Timer».

Краткая характеристика области применения: приложение таймер – полезное оснащение для телефона того, кому необходимо замерять время. Пользователь может установить таймер на определённое время и запустить отсчёт таймера.

Наименование темы разработки – «Разработка мобильного приложения для определения и фиксации промежутков времени (таймер)».

Условное обозначение темы разработки – «Мобильное приложение - Timer».

1. **Назначение разработки**

Основное назначение мобильного приложения заключается в возможности запуска таймера.

Лица, которые могут работать с данной Системой:

* пользователь – устанавливает время для таймера, запускает/ставит на паузу/сбрасывает таймер, просматривает логи.

**2 Требования к мобильному приложению**

**2.1 Требования к функциональным характеристикам**

Требования к составу выполняемых функций:

* установка отсчёта времени для таймера;
* запуск/пауза/сброс таймера;
* запись и вывод логов.

**2.2 Требования к надёжности**

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

* организация бесперебойного питания оборудования пользователя;
* использование лицензионного программного обеспечения.

Время восстановления после отказа, вызванного сбоем электропитания технических средств, не фатальным сбоем ОС или файловой системы, не должно превышать 15 минут при соблюдении условий эксплуатации технических и программных средств и правильной настройки операционной системы.

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

Для работы приложения на устройстве требуется: объем свободной памяти не менее 3 ГБ, версия устройства Android 11 и выше.

**2.4 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.

**2.5** **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;
* проектную документацию.

**4 Технико-экономические показатели**

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

**5 Стадии и этапы разработки**

Таблица 1 – Стадии разработки

|  |  |  |
| --- | --- | --- |
| Название этапа | Срок, даты | Отчётность |
| Исследование предметной области | 25.01.2023-28.01.2023 | Пояснительная записка |
| Выбор моделей, описывающих предметную область | 01.02.2023-03.02.2023 | Пояснительная записка |
| Разработка технического задания | 03.02.2023-05.02.2023 | Техническое задание |
| Изучение Java | 01.03.2023-  18.03.2023 | Пояснительная записка |
| Анализ требований и уточнение спецификаций | 14.03.2023 | Спецификации программного обеспечения |
| Проектирование структуры мобильного приложения, проектирование компонентов (технический проект) | 15.03.2023-29.03.2023 | Структурная схема системы и спецификации компонентов |
| Кодирование клиентской части | 01.04.2023-20.05.2023 | Программный  Продукт |
| Тестирование компонентов.  Сборка и комплексное тестирование | 20.05.2023 | Тексты программных компонентов |
| Разработка программной  документации | 24.05.2023 – 26.05.2023 | Программная  документация |
| Защита | 29.05.2023 |  |

**6 Порядок контроля и приёмки**

Виды испытаний – защита курсового проекта.

Общее требования к приёмке:

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