

**ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»**

**НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ «ШАГ В БУДУЩЕЕ,  
МОСКВА»**

1198
------

*регистрационный номер*

**Информатика и системы управления**

*название факультета*

**Программное обеспечение ЭВМ и информационные технологии**

*название кафедры*

**Разработка андроид приложения для автоматизированной проверки  
устной части ЕГЭ по английскому**

*название работы*

**Автор:**

***Пырко Алексей Владиславович***

*фамилия, имя, отчество*

**ГБОУ “школа №1584”, 11 класс**

*наименование учебного заведения, класс*

**Москва - 2019**

## Оглавление

<b>Введение:</b> .....	3
Проблема проекта .....	3
Цель работы.....	3
Актуальность.....	3
<b>Критерии результативности проекта</b> .....	4
<b>Концепция</b> .....	4
<b>Определение доступных ресурсов</b> .....	4
<b>План выполнения проекта</b> .....	5
<b>Реализация плана</b> .....	5
<b>Оценка эффективности и результативности</b> .....	6
<b>В перспективах</b> .....	6
<b>Литература</b> .....	7
<b>Приложение</b> .....	8-15

## **Введение:**

В 21 веке появились информационные технологии, которые упрощают и автоматизируют процессы в науке, здравоохранении и других сферах. В образовании этот процесс начался недавно, но у него есть перспективы развития: технологии распознавания речи и андроид приложения могут оптимизировать работу специалистов. В своей работе я использовал эти технологии для автоматизации проверки устной части ЕГЭ по английскому языку.

## **Проблема проекта:**

Ряд неудобств при подготовке обучающихся к устной части ЕГЭ по английскому:

- 1) Отсутствие возможности получения обратной связи при самостоятельной подготовке к ЕГЭ;
- 2) Учащиеся не всегда могут разобрать устную часть с учителем;
- 3) Необходимость специализированного программного обеспечения для записи ответа;

## **Цель работы:**

Автоматизировать проверку устной части ЕГЭ по английскому методами распознавания речи с помощью андроид приложения.

## **Актуальность:**

На данный момент аналогов подобной системы не существует, кроме проекта EGEEnglish.ru, в партнёрстве с которым и была подготовлена данная работа (отзыв руководителя проекта можно прочитать в пдф файле)

### **Критерии результативности проекта:**

- 1) Работоспособность (необходимо обеспечить стабильную работу приложения)
- 2) Скорость работы (необходимо минимизировать потери времени пользователем)
- 3) Функциональность (обеспечение функциональности сайта EGEEnglish.ru)
- 4) Доступность (доступ к приложению в Play market)
- 5) Качество распознаваемой речи (эффективность распознавания)

### **Концепция:**

Проект будет осуществлён посредством взаимодействия андроид приложения и API<sup>1</sup> сайта EGEEnglish.ru. Предполагается использование приложения школьниками для подготовки к устной части ЕГЭ по английскому.

### **Определение доступных ресурсов:**

Для реализации проекта требуется:

- 1) Язык программирования и среда разработки (IDE);
- 2) Телефон с операционной системой android или эмулятор<sup>2</sup>;
- 3) Само ПО в виде работающего приложения;
- 4) Стабильная работа API EGEEnglish.ru;
- 5) Много свободного времени (проект частично реализован, требуются доработки);

### **План выполнения проекта:**

- 1) Выбор и установка языка программирования, настройка рабочей среды;
- 2) Создание макета приложения;
- 3) Разработка графической части приложения;
- 4) Выбор способа хранения текста заданий;
- 5) Написание кода для получения пользовательской речи, её обработки и постредактирования (для улучшения распознавания);
- 6) Создание кода для взаимодействия приложения и API EGEEnglish.ru (Получение оценок и дополнительной информации, предоставление её пользователю).
- 7) Занесение информации о пользователе в базу данных для сбора статистики;
- 8) Введение обучения при первом использовании приложения;

### **Реализация плана:**

- 1) Языком программирования был выбран java, так как он используется как нативный язык для андроид разработки. В качестве рабочей среды была использована Android Studio.
- 2) Создан прототип приложения с помощью программы Adobe XD;
- 3) Разработан графический интерфейс: экран авторизации, выбора вариантов ЕГЭ, заданий (изображения можно посмотреть в приложении);
- 4) Задания хранятся локально на телефоне, в формате JSON<sup>3</sup>;

- 5) Для записи речи использованы модули из стандартной библиотеки андроида, для её перевода в необходимый формат используется внешняя библиотека;
- 6) Написан код, обрабатывающий события, которые будут получены от API EGEEnglish.ru по HTTP<sup>4</sup> соединению;
- 7) Планируется сделать в ближайшем времени (с помощью платформы Firebase)
- 8) Планируется сделать в ближайшем времени (в виде анимации 9 patch image)

### **Оценка эффективности и результативности:**

Андроид приложение практически полностью реализовано, необходимы небольшие доработки, после которых приложение можно будет выкладывать в Play Market.

Ссылка на бета-версию приложения:



### **В перспективах:**

- 1) Техническая доработка и поддержка приложения (выявление и устранение багов);
- 2) Улучшение дизайна;
- 3) Расширение функциональности приложения;

### **Литература:**

- 1) Head First java. Эксмо. 2017. 720с
- 2) Философия java. Питер. 2016. 1168с
- 3) Head First. Программирование для Android. Питер. 2016. 704с
- 4) Материалы по android разработке от IT школы samsung

## Приложение 1

### Список терминов:

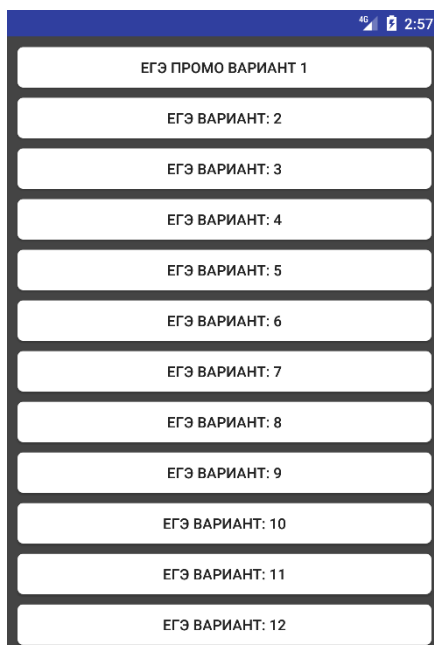
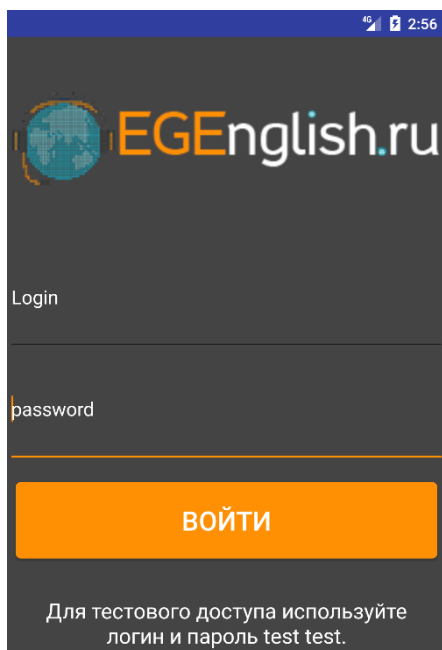
- 1) API (Application Programming Interface) — это интерфейс программирования, интерфейс взаимодействия приложений.
- 2) Эмулятор для андроид – программа, которая позволяет на персональном компьютере пользоваться приложениями смартфона, не предназначенными для Windows, Linux и OS X.
- 3) JSON - текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми.
- 4) HTTP – протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных). Основой HTTP является технология «клиент-сервер».



## Приложение 2

### Графический интерфейс приложения:

- 1) Экран входа
- 2) Экран выбора вариантов
- 3) Диалог выбора задания
- 4) Первое задание
- 5) Второе задание
- 6) Третье задание
- 7) Четвёртое задание
- 8) Внесение поправок в распознанную речь
- 9) Получение оценки





## Приложение 3

### Фрагменты кода:

#### 1) Авторизация пользователя

```
public void login(View view) {  
    TextView login = findViewById(R.id.login);  
    TextView password = findViewById(R.id.password);  
  
    Retrofit retrofit = new Retrofit.Builder()  
        .baseUrl("https://egenglish.ru")  
        .addConverterFactory(GsonConverterFactory.create())  
        .build();  
    AuthApi api = retrofit.create(AuthApi.class);  
  
    api.checkUser(login.getText().toString(), password.getText().toString()).enqueue(new Callback<User>() {  
        @Override  
        public void onResponse(@NonNull Call<User> call, @NonNull Response<User> response) {  
            User user = response.body();  
            if (user.IsRegistered()) {  
                Intent intent = new Intent(getApplicationContext(), tests.class);
```

```

        intent.putExtra(userLogin, login.getText().toString());
        intent.putExtra("hasAccess", user.hasAccess());
        startActivity(intent);
    } else {
        Toast.makeText(getApplicationContext(),
            "Неверный логин и/или пароль.",
            Toast.LENGTH_LONG).show();
    }
}

```

**@Override**

```

public void onFailure(@NonNull Call<User> call, @NonNull Throwable
t) {
    Log.e("ERROR", t.getMessage());
    Toast.makeText(getApplicationContext(),
        "Произошла ошибка при подключении к сайту Egenglish.ru",
        Toast.LENGTH_SHORT).show();
    }
});
}

```

## 2) Считывание задания из JSON формата

```

public static String[] readFromJsonFile(Context context, int variantNumber, int
taskNumber) {

```

```
String ret = "";
```

```
try {
```

```
    InputStream inputStream = context.openFileInput("ege" + Integer.toString(variantNumber) + ".json");
```

```
    if (inputStream != null) {
```

```
        InputStreamReader inputStreamReader = new InputStreamReader(inputStream);
```

```
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
```

```
        String receiveString;
```

```
        StringBuilder stringBuilder = new StringBuilder();
```

```
        while ((receiveString = bufferedReader.readLine()) != null) {  
            stringBuilder.append(receiveString);  
        }
```

```
        inputStream.close();
```

```
        ret = stringBuilder.toString();
```

```
    }
```

```
} catch (FileNotFoundException e) {
```

```
    Log.e("tests activity", "File not found: " + e.toString());
```

```
} catch (IOException e) {
```

```
    Log.e("tests activity", "Can not read file: " + e.toString());
```

```
}
```

```
String fr1Text = "NaN";
```

```
String fr2Text = "NaN";
```

```

JSONObject obj;
try {
    obj = new JSONObject(ret);
    JSONObject variant =
obj.getJSONObject(Integer.toString(variantNumber));
    JSONArray tasks = variant.getJSONArray("task" + Integer.toString(taskNumber));

    fr1Text = tasks.get(0).toString();
    fr2Text = tasks.get(1).toString();

} catch (JSONException e) {
    e.printStackTrace();
}
return new String[]{fr1Text, fr2Text};
}

```

### 3) Создание таймера

```

timer = Rx2Timer.builder()
    .take((taskNumber == 1 || taskNumber == 2) ? 90 : 120) //90 seconds for
1 and 2 tasks else 120
    .unit(TimeUnit.SECONDS)
    .onEmit(count -> {
        if (count < 10) {
            timerText.setText("0" + count + " секунд");

```

```

        } else {
            timerText.setText(count + " секунд");
        }
    })
    .onError(e -> timerText.setText(R.string.timer_text))
    .onComplete(() -> {
        stopRecording("Вы не уложились по времени! Попробуйте
снова.");
        recordButton.setImageResource(R.drawable.icon_start_recording);
        isRecording = false;
    })
    .build();

```

P.S. полный код можно посмотреть в репозитории [github](#)