**Министерство образования и науки Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление – Информационные системы и технологии

**Лабораторная работа №3**

по дисциплине: Программирование мобильных устройств и встраиваемых систем

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил: студент гр. 8И6А | \_\_\_\_\_\_\_\_\_\_\_ | Стецов И.С. | |
|  |  | |  |
| Проверил: ассистент ОИТ | \_\_\_\_\_\_\_\_\_\_\_ | | А. В. Погребной |

Томск 2020 г.

# Задание

**Вариант 1**

На экран со студентами добавить SearchView. При поиске искать по ФИО студентов в списке на наличие введённой фразы в SearchView и выводить отфильтрованный список. Для каждого результата подсвечивать цветом кусок текста, который нашёлся (через SpannableString).

# Ход работы

Поле SpannableString это поле ФИО, к которому будут применяться выделения цветом и он же и будет выводиться в списке.

В задании требуется выделять текст с помощью SpannableString, поэтому добавляем в класс студента это поле.

public Spannable shownName;

Далее нам требовалось добавить элемент поиска:

MenuItem search = menu.findItem(R.id.search);

SearchView searchView = (SearchView);

MenuItemCompat.getActionView(search);

search(searchView);

Сама функция search обрабатывает изменения текста в поле поиска и запрашивает применение фильтра через адаптер.

Выше сказанная функция поиска проверяет произошло ли нажатие кнопки «подвердить»:

public boolean onQueryTextSubmit(String query) {

studentsAdapter.getFilter().filter(query);

return false;

}

И проверяет любое изменение текста ниже:

public boolean onQueryTextChange(String newText) {

studentsAdapter.getFilter().filter(newText);

return true;

}

Создаётся второй список студентов, которые имеет отфильтрованные списки, которые впоследствии выводятся на экран

private List<Student> filteredStudents = new ArrayList<>();

Если запрос отсутствует: пользователь его стер или ещё не начал вводить, то требуется убрать все выделения и вернуть стандартные список студентов.В противном случае происходит обход всех студентов, среди которых обнаруживаются подходящие запросу с помощью функции contains(). У всех найденных таким образом студентов сначала сбрасывается предыдущее выделение, а затем выполняется поиск, на каких именно отрезках строки было найдено совпадение. По найденным отрезкам накладываются новые выделения, после чего полученный отфильтрованный список передается на вывод.

ArrayList<Student> filteredList = new ArrayList<>();

for (Student student : students) {

if (student.getName().toLowerCase().contains(charString)) {

clearColor(student);

List<Integer> positions = findWord(charString,

student.shownName.toString().toLowerCase());

for (int i = 0; i < positions.size()-1; i = i + 2)

{

if (positions.get(i) != -1)

student.shownName.setSpan(new ForegroundColorSpan(Color.RED), positions.get(i), positions.get(i+1)+1, Spanned.SPAN\_EXCLUSIVE\_EXCLUSIVE);

}

filteredList.add(student);

}

}

filteredStudents = filteredList;

После активируем функцию publishResults, которая обновляет список и уведомляет об его изменении:

protected void publishResults(CharSequence charSequence,

FilterResults filterResults)

{

filteredStudents = (ArrayList<Student>) filterResults.values;

notifyDataSetChanged();

}

# Работа приложения

На главном экране приложения выводится список студентов. В правом нижнем углу есть кнопка, для добавления студентов(рисунок 1):

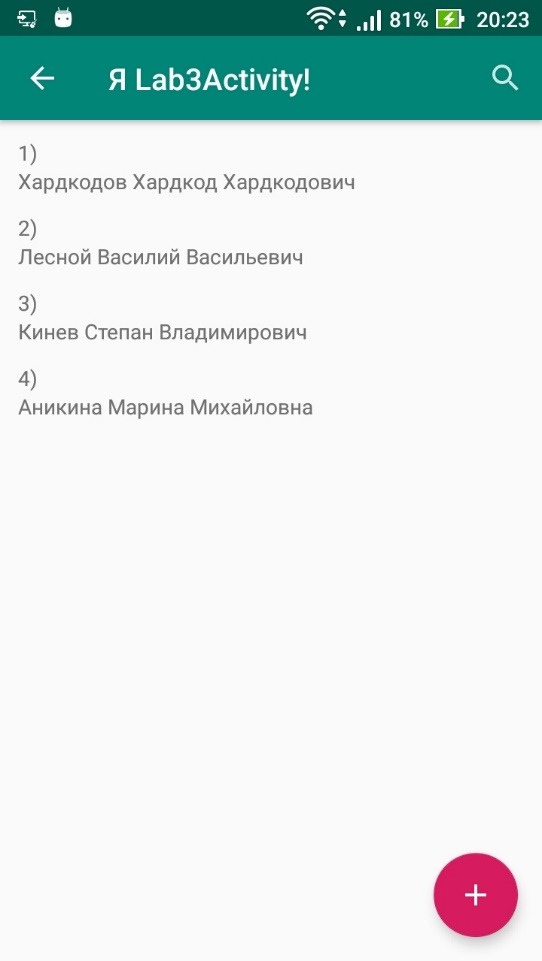


Рисунок 1 – Список студентов и кнопка

При нажатии на кнопку открывается экран с добавлением студента(рисунок2).

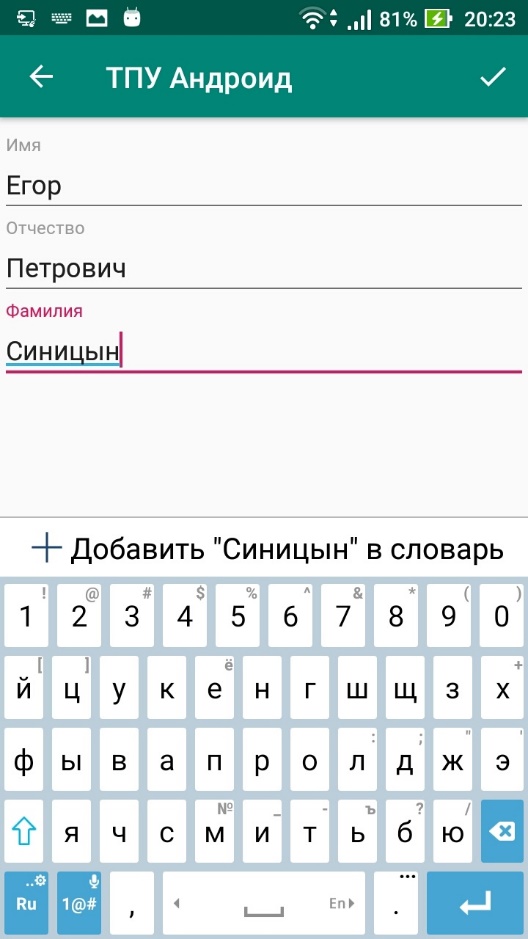


Рисунок 2 – Добавление нового студента

Студент добавился в основной список(рисунок 3).

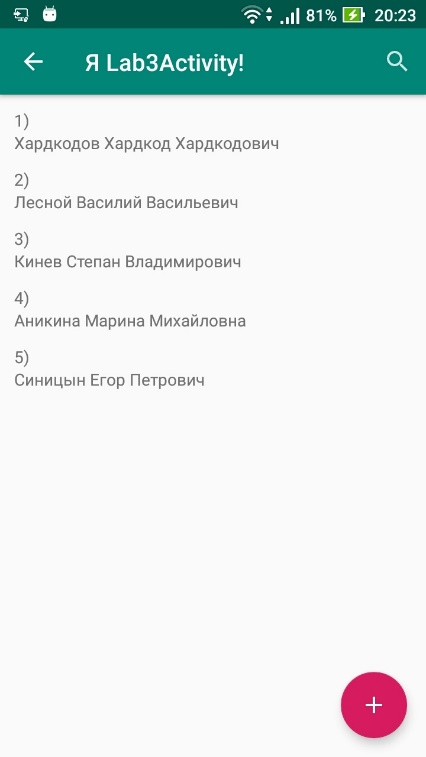


Рисунок 3 – Список с новым студентом

В верхней части строка поиска(рисунок 4):

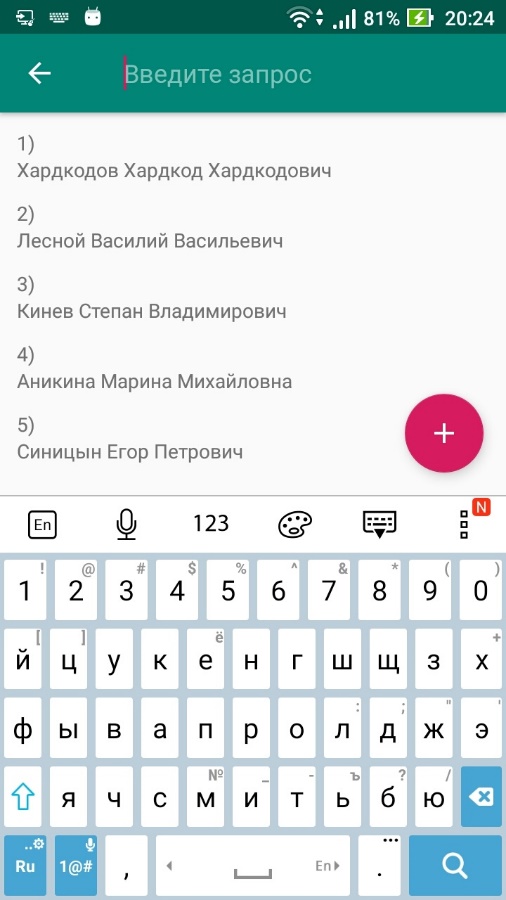


Рисунок 4 – Ввод запроса в строку поиска

На рисунках 5 и 6 показано, что при вводе запроса список автоматически отфильтровывается: на экране остаются только студенты, имеющие совпадения в ФИО с запросом, и эти совпадения окрашиваются в красный.

При вводе в строку поиска букв список автоматически фильтруется и выходят студенты с совпадениями, окрашиваясь цветом(рисунок 5).

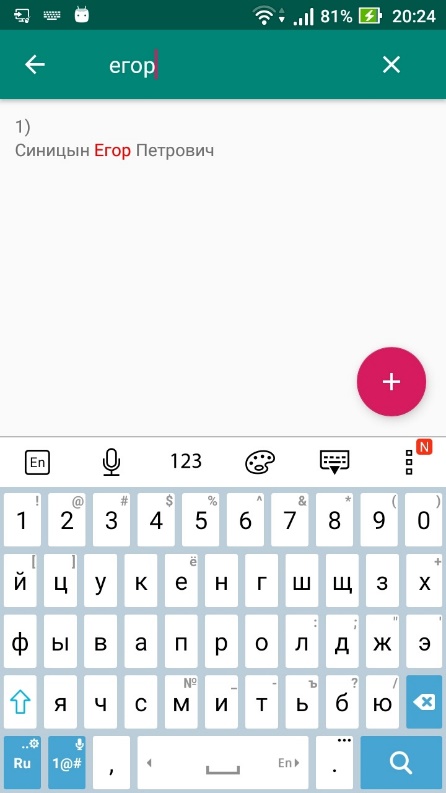


Рисунок 5 – Поиск

Так же есть у нас есть несколько совпадений, то все совпадения окрашиваются и студент так же выводится(рисунок 6).

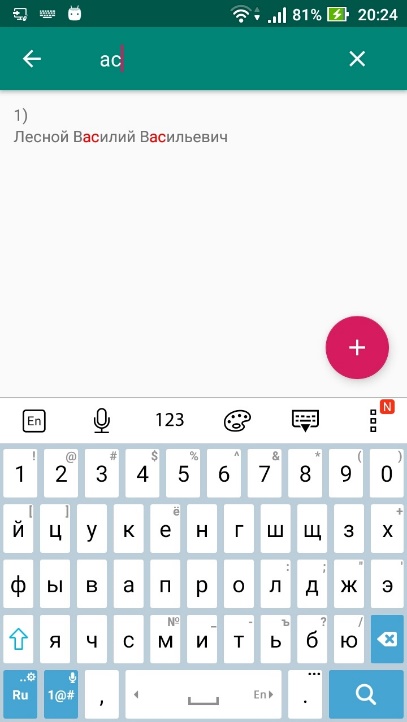


Рисунок 6 – Множество совпадений

# Итоги работы

В ходе выполнения лабораторной работы было создано приложения, которое позволяет хранить список студентов, а так же добавлять их и осуществлять поиск по соответствиям, подсвечивая их.