Concordia University
COEN/ELEC 390
Winter 2019
Technical Assignment 2

Deadline:	Sunday February 3rd, 2019
Late Submission:	-20% for every 1 day late after due date

Objective: Design and implement an android mobile application with description given below. Using Shared Preferences to save data and read data in an MVC structure. Preventing/detecting any invalid data to be input by the user.

Application Description

- Two Activities: mainActivity and profileActivity.
  - profileActivity is a child Activity to mainActivity.
- mainActivity has:
  - A button with a label of the name of the profile saved that takes you to the profileActivity.
  - If there is no name saved already the user will be redirected to the profileAcitivity.
- When profileActivity is created:
  - The user will be able to see the profile information displayed.
  - The user can switch to "edit mode" by pressing an action from the action bar.
  - The user can edit the fields of the profile information when in "edit mode"
  - The user can save the information that were input by the user.
- The Profile has the following information: **Name, Age, Student ID.**
- The profileActivity will have two modes:
  - **Display mode:** The profileActivity is by default in display mode. The profile information is displayed on the activity layout as they were saved in the SharedPreference file. If the information does not exist, the text of every field of information will be empty. The EditTexts displaying the information will not be editable. Meaning the user cannot write any text, but can only view the text.
  - **Edit mode:** when the action edit, from the action bar, is pressed the activity switches to edit mode. The Edit Texts of the fields will switch to being editable where the user can write the information of his/her profile. A Button will show up at the bottom of the activity labeled "Save". When this button is pressed the data (if no wrong values are entered) will be saved to the file and the Save button will disappear and the profileActivity will switch to display mode. If any value entered is wrong the user will get a message Toast and nothing will be saved, and the activity will stay in its current mode.
- Valid data for profile information:
  - Name: only alphabetical characters.
  - Age: 18-99
  - Student ID: numerical values maximum 6 digits.

Things to help you with the assignment

The first part of the assignment was shown as a demo in the tutorial and the android project for that demo is available on moodle.

However, **the project does not follow the MVC structure**. To do so, you will need a “Controller” for the SharedPreference. Which we create as a java class and call SharedPreferenceHelper.

```
public class SharedPreferenceHelper {

    private SharedPreferences sharedPreferences;
    public SharedPreferenceHelper(Context context)
    {
        sharedPreferences = context.getSharedPreferences("ProfilePreference",
Context.MODE_PRIVATE );
    }

    public void saveProfileName(String name)
    {
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("profileName",name );
        editor.commit();
    }

    public String getProfileName()
    {
        return sharedPreferences.getString("profileName", null);
    }
}
```

and in the MainActivity for example, instead of implementing the usage of SharedPreference we use SharedPreferenceHelper as follow:

```
public class MainActivity extends AppCompatActivity {


    protected SharedPreferenceHelper sharedPreferenceHelper;

    protected Button button = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        sharedPreferenceHelper = new SharedPreferenceHelper(MainActivity.this);

        button = (Button) findViewById(R.id.profileButton);


        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {


                 goToProfileActivity();

            }
        });
    }

    protected void onStart()
    {
        super.onStart();

        String name = sharedPreferenceHelper.getProfileName();
        if(name == null)
            goToProfileActivity();
        else
            button.setText(name);

    }

    void goToProfileActivity()
    {
        Intent intent = new Intent(MainActivity.this, Profile.class);
        startActivity(intent);
    }


    }
```

This example only works for saving and getting a String representing the profile name. However, in this assignment you will need to get a Profile and save a Profile.

Tip: create a Profile java class that holds all the information of the profile with getters and setters. And your SharedPreferenceHelper methods will take Profile object as parameter (or return a Profile object) instead of a String.

Detecting invalid inputs:

In order to detect an invalid input, you can do that by reading the input from the edit text when the save button is pressed and implement a method to check if the input is valid or not.

However, few of the inputs that you have can be managed in a way to prevent the user to input any valid information.

For example, for the name you can add the following line to EditText in the xml file to make sure only alphabetical characters are allowed:

android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"

source:

http://stackoverflow.com/questions/2361497/how-to-create-edittext-accepts-alphabets-only-in-android

For the Student ID, you can use a numerical edit text and specify android:maxLength in the xml file to specify the max length of 6. Or you can do it in the activity java class.

Example of the edit text of the name in the demo project:

```
nameEditText = (EditText) findViewById(R.id.editText);
int maxLength = 6;
InputFilter[] FilterArray = new InputFilter[1];
FilterArray[0] = new InputFilter.LengthFilter(maxLength);
nameEditText.setFilters(FilterArray);
```

Preventing the user from writing invalid inputs will reduce the work when reading the input and saving it. For example, for the Name you don't need to check every character if it's an alphabet or not anymore, you just need to check if the field is empty or not. Etc...

One Last Hint:

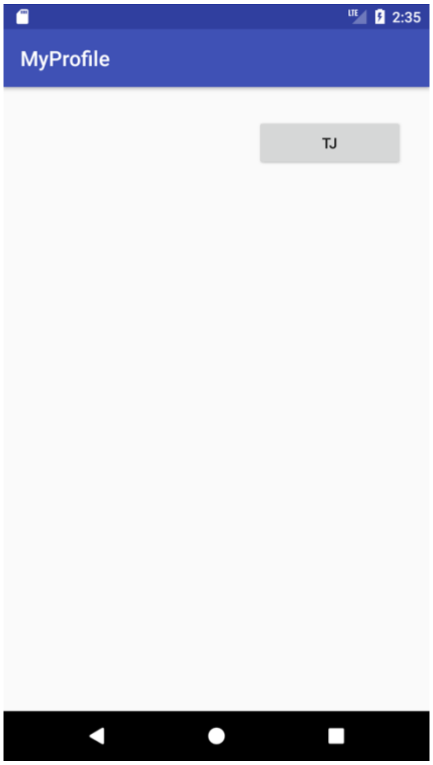
To make an edit text editable:

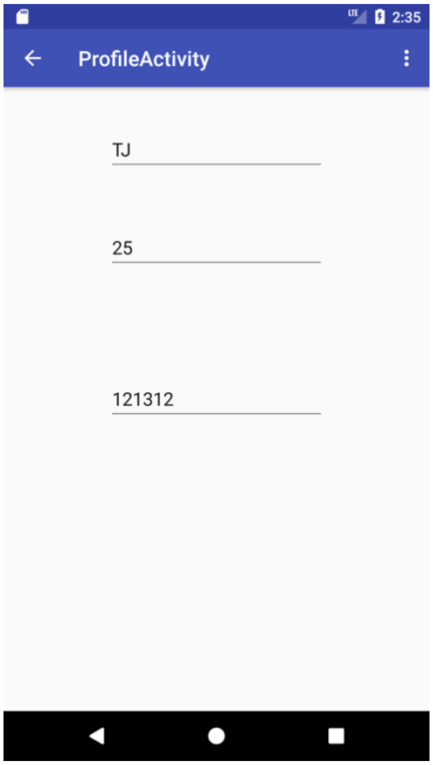
```
mNameEditText.setFocusableInTouchMode(true);
```

To make an edit text un-editable

```
mNameEditText.setFocusable(false);
```

Sketches to help you understand the functionality of the App

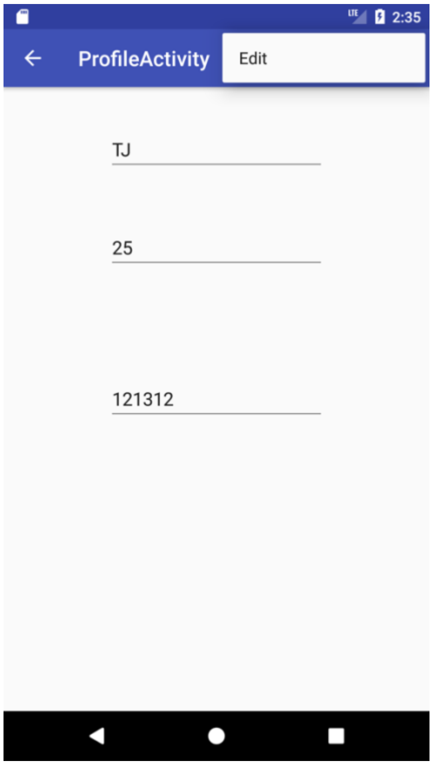




TJ

25

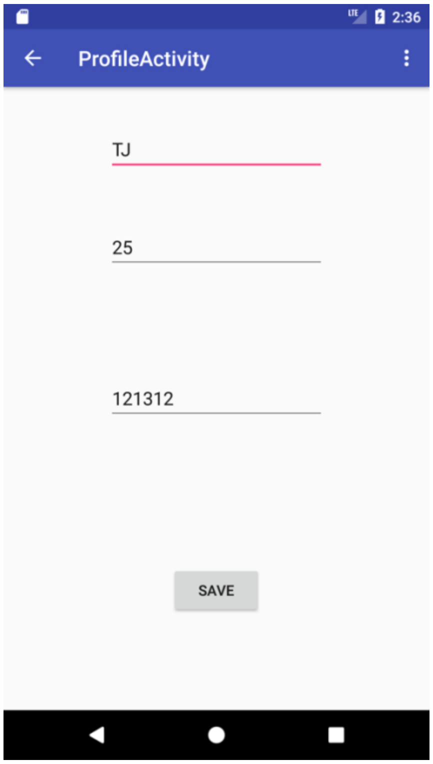
121312



TJ

25

121312



TJ

25

121312

Link to see this app in action: https://youtu.be/zw495ryLCtY

Assignment submission and procedure

You have to submit your assignment before midnight on the due date using moodle Assignment Submission. The file submitted must be a **.zip** file named **StudentID_Ass2** containing your android project. **Before submitting your code make sure you clean the project.**

```
Android Studio --> Build --> Clean
```

Evaluation criteria and grading scheme

Meeting the requirements and use cases	70%
Using MVC design	25%
Clean code: well commented, proper naming, easy to read and understand.	5%

If the project submitted does not compile and run the student will receive a grade of 0! So, make sure even if the assignment is not completely done that you submit an application that can be built and run. We will not grade none compiling code.