

```
git clone https://www.github.com/eim2017/Laborator01.git
cd Laborator01
git remote add Laborator01_perfectstudent https://github.com/perfectstudent/Laborator01
git push Laborator01_perfectstudent master
git add *
git commit -m "implemented x"
git push Laborator01_perfectstudent master
```

## Laborator 01

- Utilizare git
- efect de fade (animatie)

## Laborator 02

- logs si filtre (LogCar)
- metode ce monitorizeaza ciclul de viata al unei aplicatii: `graphicuserinterface`
- dezactivarea optiunii de salvare a starii: in `file.xml` se adauga la obiectele selectate proprietatea `android:saveEnabled="false"`
- salvarea informatiilor din UI: `onSaveInstanceState()` si `onRestoreInstanceState()` ce poate fi cuprinsi si in `onCreate()`
- `onCreate()`:  
`EditText usernameEditText = (EditText) findViewById(R.id.username_edit_text);`

## Laborator 03

- Git repository cu fisiere
- Aplicatie Android noua; design vizual
- Clase ascultator pentru tratarea evenimentelor:
  - Implementare clasa listener: `private class CallImageButtonClickListener implements View.OnClickListener`
  - instantiere obiect listener
  - legare obiect listener de buton:  
`callImageButton.setOnClickListener(callImageButtonClickListener);`
- Intentia care realizeaza legatura telefonica
- Gestionare rotire ecran: nou fisier de layout pentru landscape

## Laborator 04

- `ScrollView`
- Vizibilitate sectiuni: `setVisibility()`
- lansarea in executie a altei aplicatii: `startActivity(intent);` app sa poata fi lansata + set launch to nothing + sa fie lansata de alta aplicatie

## Laborator 05

- Starter service: onStartCommand se porneste threadul de difuzare intentii;  
context.sendBroadcast(intent)
- Starter service activity: BroadcastReceiver
  - Pornire serviciu
  - Pornire broadcast receiver
  - Filtru intentii
  - Connect them onPause
- Boudner service: `public class BoundedServiceBinder extends Binder`

## Test constants

### *PracticalTest01;*

```
ro.pub.cs.systems.eim.practicaltest01;  
PracticalTest01MainActivity;  
activity_practical_test01_main.xml.
```

## Anexe

```
@Override  
public void onSaveInstanceState(Bundle savedInstanceState) {  
    super.onSaveInstanceState(savedInstanceState);  
    EditText usernameEditText = (EditText) findViewById(R.id.username_edit_text);  
    EditText passwordEditText = (EditText) findViewById(R.id.password_edit_text);  
    CheckBox rememberMeCheckBox = (CheckBox) findViewById(R.id.remember_me_checkbox);  
    if (rememberMeCheckBox.isChecked()) {  
        savedInstanceState.putString(Constants.USERNAME_EDIT_TEXT,  
usernameEditText.getText().toString());  
        savedInstanceState.putString(Constants.PASSWORD_EDIT_TEXT,  
passwordEditText.getText().toString());  
        savedInstanceState.putBoolean(Constants.REMEMBER_ME_CHECKBOX,  
rememberMeCheckBox.isChecked());  
    }  
}  
@Override  
public void onRestoreInstanceState(Bundle savedInstanceState) {  
    super.onRestoreInstanceState(savedInstanceState);  
    if (savedInstanceState.containsKey(Constants.USERNAME_EDIT_TEXT)) {  
        EditText usernameEditText = (EditText) findViewById(R.id.username_edit_text);  
  
usernameEditText.setText(savedInstanceState.getString(Constants.USERNAME_EDIT_TEXT));  
    }  
    if (savedInstanceState.containsKey(Constants.PASSWORD_EDIT_TEXT)) {  
        EditText passwordEditText = (EditText) findViewById(R.id.password_edit_text);  
  
passwordEditText.setText(savedInstanceState.getString(Constants.PASSWORD_EDIT_TEXT));  
    }  
    if (savedInstanceState.containsKey(Constants.REMEMBER_ME_CHECKBOX)) {  
        CheckBox rememberMeCheckBox = (CheckBox) findViewById(R.id.remember_me_checkbox);  
  
rememberMeCheckBox.setChecked(savedInstanceState.getBoolean(Constants.REMEMBER_ME_CHECKBOX));  
    }  
}  
  
Integer.parseInt(first.getText().toString());  
first.getInputType() != TYPE_CLASS_NUMBER;
```