BSc (Hons) in Information Technology
Year 2 – Semester 2

Lab Exercise 03

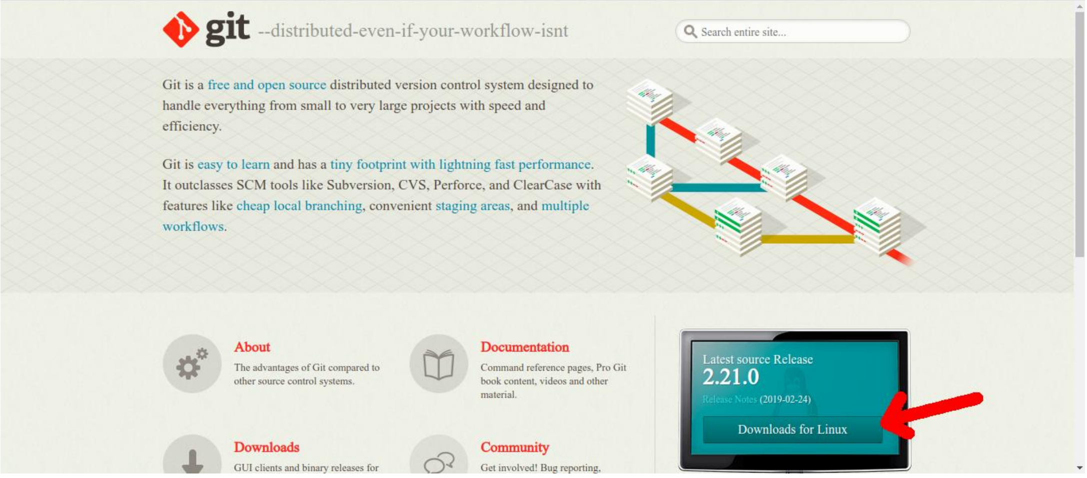
IT2010 – Mobile Application Development

Using Git version control system

- Git is a version control system (VCS) for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development.
- GitHub is a web-based Git or version control repository and Internet hosting service. It is mostly used for code. It offers all the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It lets you and others work together on projects from anywhere.

1. Download Git using the following URL.

URL: https://git-scm.com/



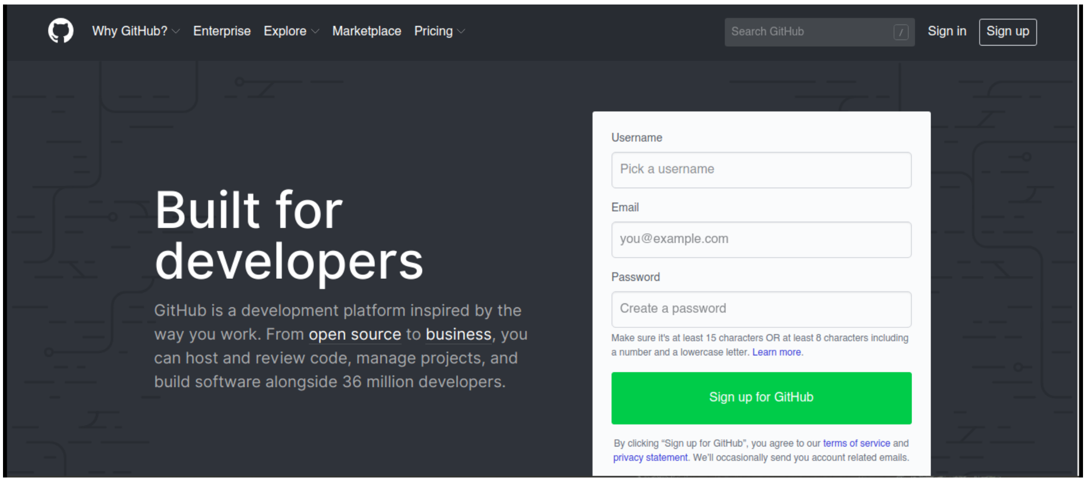
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

About
The advantages of Git compared to other source control systems.
Documentation
Command reference pages, Pro Git book content, videos and other material.
Downloads
GUI clients and binary releases for
Community
Get involved! Bug reporting,
Latest source Release
2.21.0
Release Notes (2019-02-24)
Downloads for Linux

2. Sign up for GitHub using the following URL.

URL: https://github.com/



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 36 million developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Sign up for GitHub

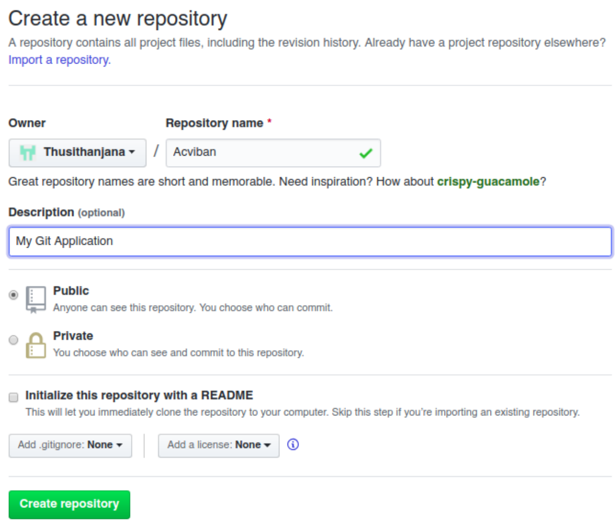
By clicking "Sign up for GitHub", you agree to our terms of service and privacy statement. We'll occasionally send you account related emails.

Note: You must remember the username, email address and password used here.

3. Create a repository.

A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets – anything your project needs.

1. In the upper right corner, click and then select New repository.
2. Name your repository <<project-name>>
3. Write a short description
4. Click Create repository. Note that, here we haven't initialized the readme file.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: Thusithanjana / Repository name *: Acviban ✓

Great repository names are short and memorable. Need inspiration? How about crispy-guacamole?

Description (optional): My Git Application

● Public
Anyone can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

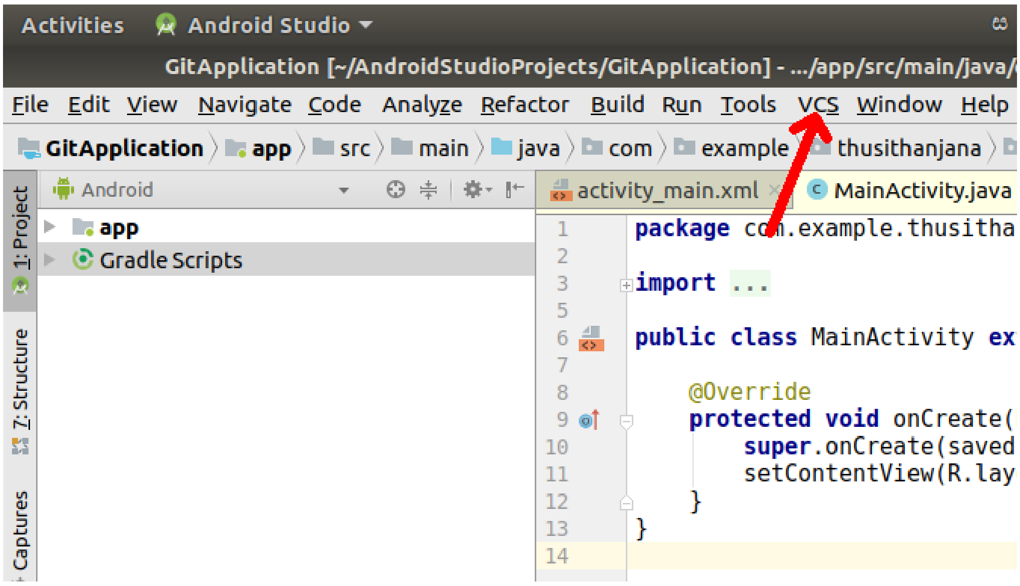
Add .gitignore: None ▾ | Add a license: None ▾ ⓘ

Create repository

4. Push Android Studio project to GitHub

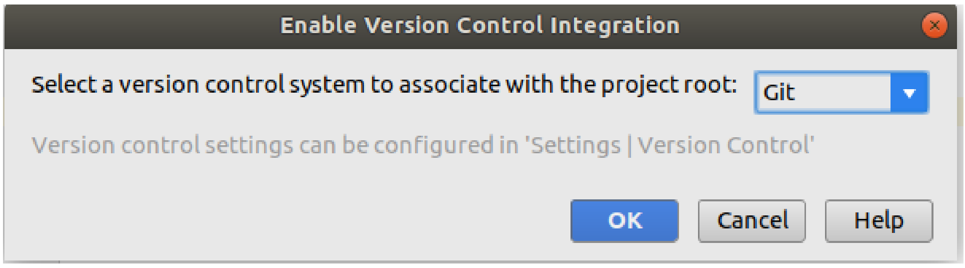
First you need to open and load your android project in Android Studio. Next follow these steps.

1. Click VCS from the Menu bar and select ‘Select Version Control Integration...’.


Activities
Android Studio
GitApplication [~/AndroidStudioProjects/GitApplication] - .../app/src/main/java/
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
GitApplication app src main java com example thusithanjana
1: Project
7: Structure
Captures
Android
app
Gradle Scripts
activity_main.xml
MainActivity.java

```
package com.example.thusitha

import ...

public class MainActivity ex

    @Override
    protected void onCreate(
        super.onCreate(saved
        setContentView(R.lay
    }
}
```

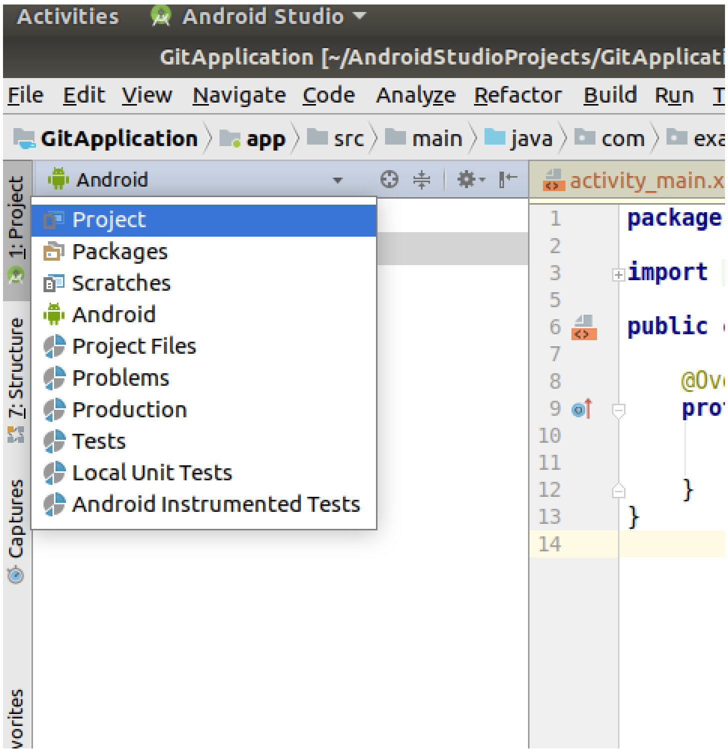
2. Then Select GIT from the message box selection dropdown and press ok.



Select a version control system to associate with the project root: Git

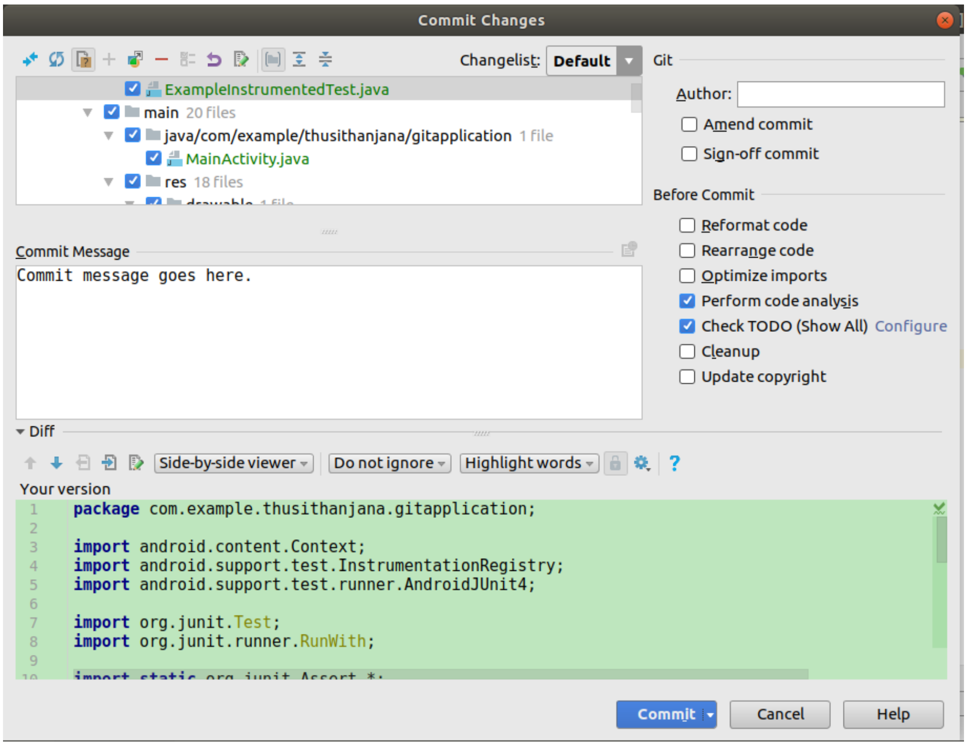
Version control settings can be configured in 'Settings | Version Control'

3. Select the folder structure to ‘project’. By default, its ‘Android’.



4. After that right click the project folder, go to Git and select ‘add’.
5. Next, right click the project folder again. Go to ‘Git’ and select ‘Commit Directory…’

6. Then add a relevant commit message and press commit. This step will commit the all the files and folders to the repository. You must put the GitHub account email to continue.



- ☑ ExampleInstrumentedTest.java
- ☑ main 20 files
  - ☑ java/com/example/thusithanjana/gitapplication 1 file
    - ☑ MainActivity.java
  - ☑ res 18 files
    - ☑ drawable 1 file

Author:

☐ Amend commit
☐ Sign-off commit

Before Commit

☐ Reformat code
☐ Rearrange code
☐ Optimize imports
☑ Perform code analysis
☑ Check TODO (Show All) Configure
☐ Cleanup
☐ Update copyright

Commit Message

```
Commit message goes here.
```

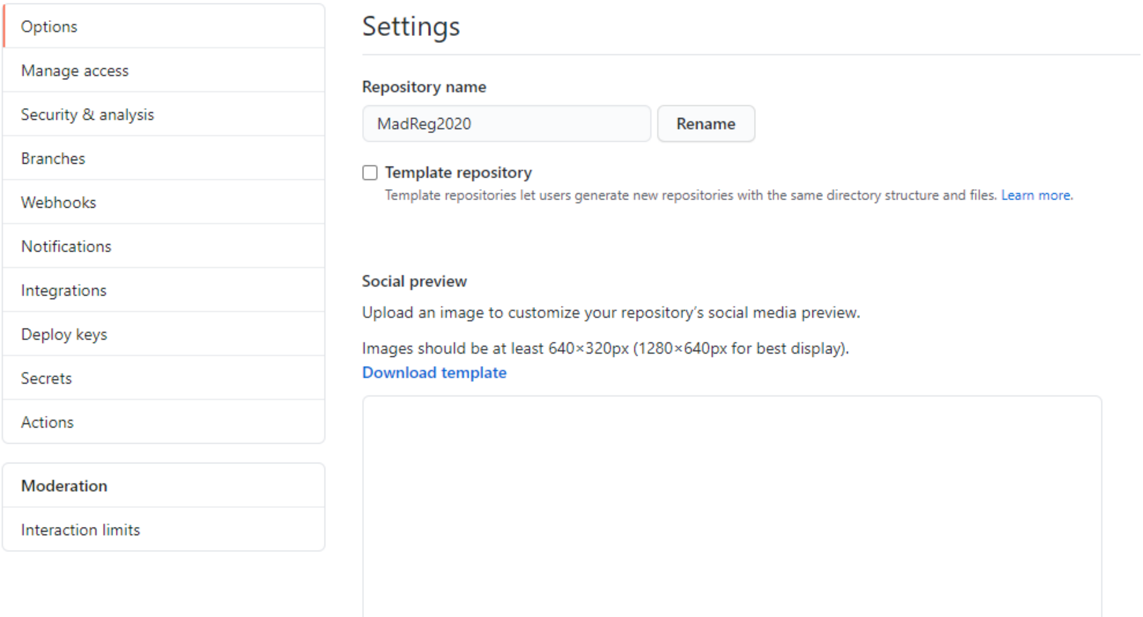
Your version

```
package com.example.thusithanjana.gitapplication;

import android.content.Context;
import android.support.test.InstrumentationRegistry;
import android.support.test.runner.AndroidJUnit4;

import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;
```

7. It will show some warnings you can press ‘commit’ to move forward of press ‘Review’ to analyze the warnings.
8. Then you need to push the code into GitHub repository. To do that, right click project folder, go to ‘Git’, go to ‘Repository’ and select ‘Push’.
9. That step will open a dialog box. Click Define remote URL. Then you will be asked to enter the URL of the repository. Enter the URL of the repository created in the github.com and press ok.
10. Then press push. It will ask you to login to the GitHub. Provide the credentials and press ok.

5. Collaborations
   1. Go to your repository in GitHub
   2. Go to settings

3. Click manage access



Options
Manage access
Security & analysis
Branches
Webhooks
Notifications
Integrations
Deploy keys
Secrets
Actions

Moderation
Interaction limits

Settings

Repository name

MadReg2020 | Rename

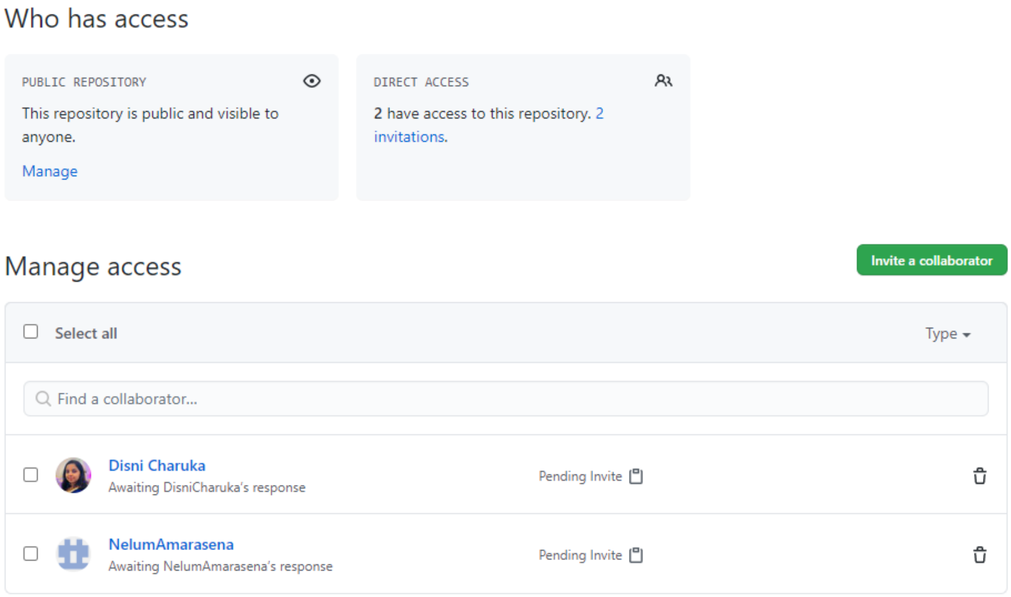
□ **Template repository**
Template repositories let users generate new repositories with the same directory structure and files. Learn more.

Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).
Download template

4. Click "Invite a Collaborator"



Who has access

PUBLIC REPOSITORY
This repository is public and visible to anyone.
Manage

DIRECT ACCESS
2 have access to this repository. 2 invitations.

Manage access

Invite a collaborator

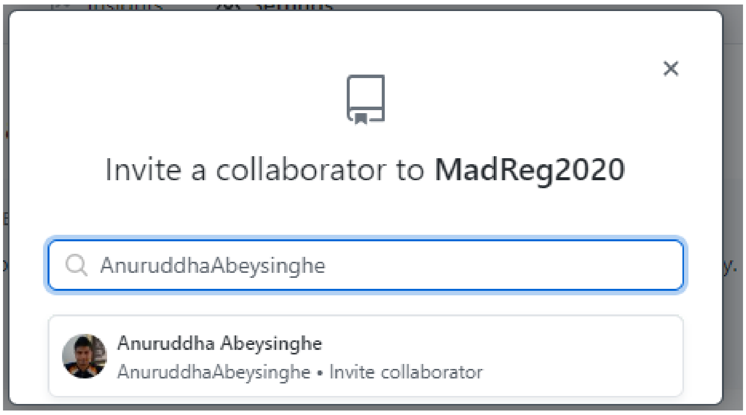
□ Select all — Type ▾

Find a collaborator...

□ Disni Charuka — Awaiting DisniCharuka's response — Pending Invite

□ NelumAmarasena — Awaiting NelumAmarasena's response — Pending Invite

5. Add the GitHub username or email address of the user

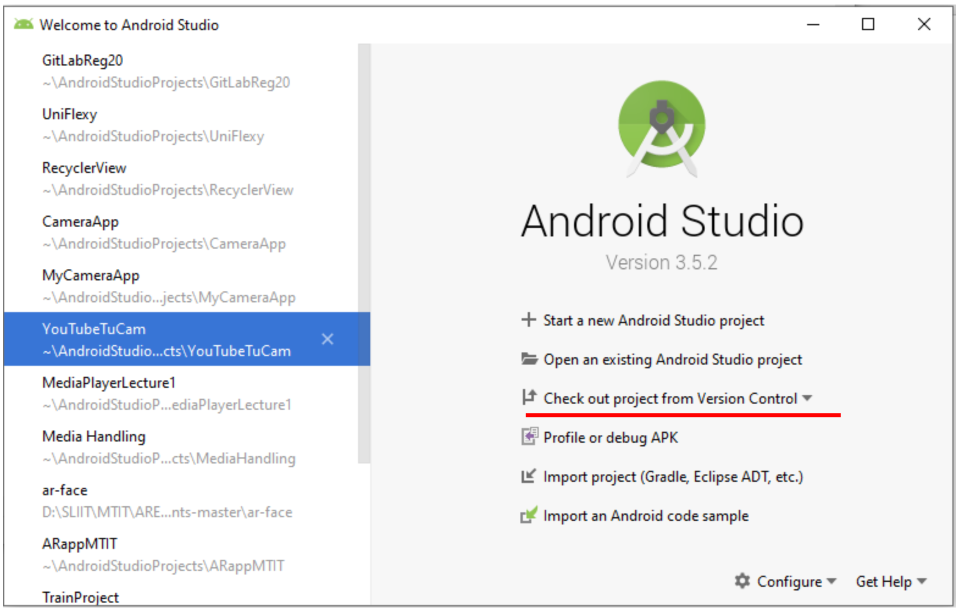


Invite a collaborator to **MadReg2020**

AnuruddhaAbeysinghe

Anuruddha Abeysinghe
AnuruddhaAbeysinghe • Invite collaborator

6. Then that user must accept it
7. Then the user can clone the repository to their machine. Click the underlined option in the welcome screen. Select the git as the option and paste the GitHub link to the repository.

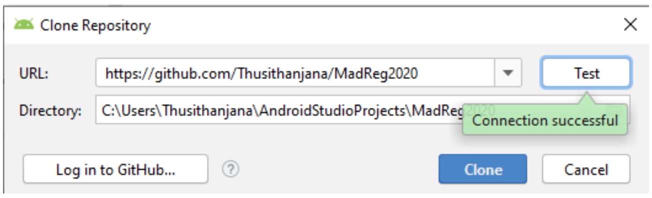


Welcome to Android Studio

GitLabReg20
~\AndroidStudioProjects\GitLabReg20
UniFlexy
~\AndroidStudioProjects\UniFlexy
RecyclerView
~\AndroidStudioProjects\RecyclerView
CameraApp
~\AndroidStudioProjects\CameraApp
MyCameraApp
~\AndroidStudio...jects\MyCameraApp
YouTubeTuCam
~\AndroidStudio...cts\YouTubeTuCam
MediaPlayerLecture1
~\AndroidStudioP...ediaPlayerLecture1
Media Handling
~\AndroidStudioP...cts\MediaHandling
ar-face
D:\SLIIT\MTIT\ARE...nts-master\ar-face
ARappMTIT
~\AndroidStudioProjects\ARappMTIT
TrainProject

Android Studio
Version 3.5.2

+ Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control ▾
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure ▾
Get Help ▾



Clone Repository

URL: https://github.com/Thusithanjana/MadReg2020

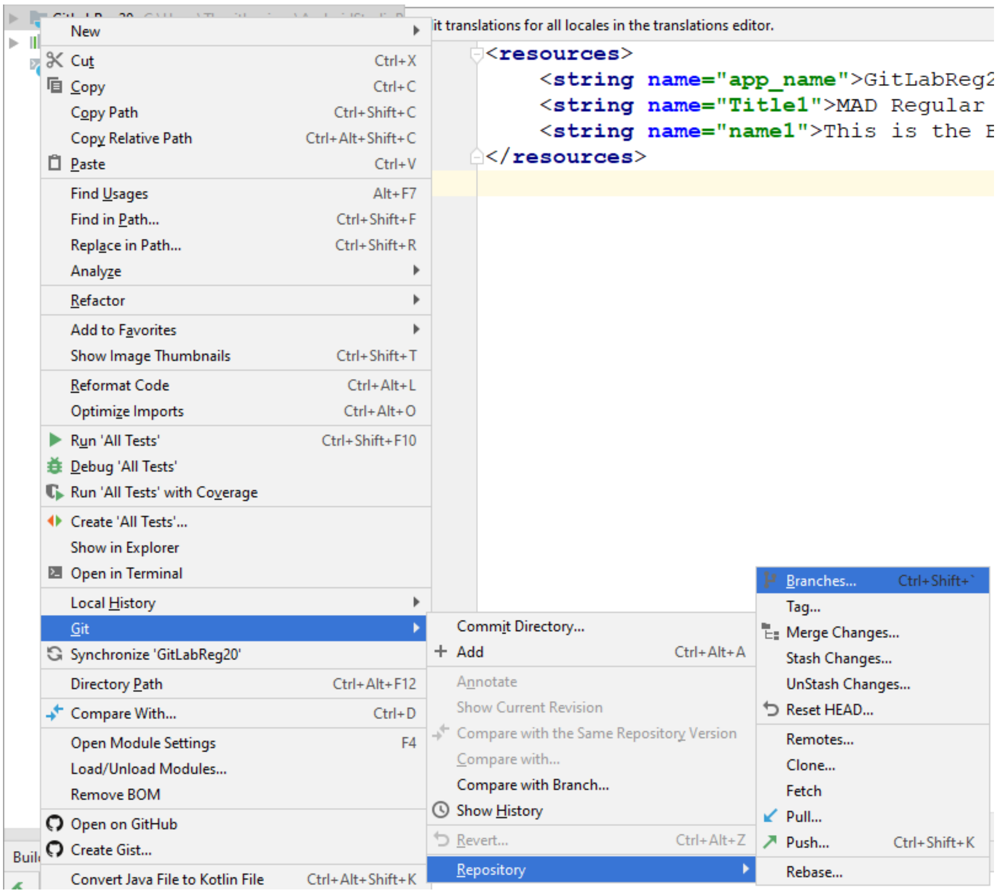
Directory: C:\Users\Thusithanjana\AndroidStudioProjects\MadReg2020

Connection successful

Test
Log in to GitHub...
Clone
Cancel

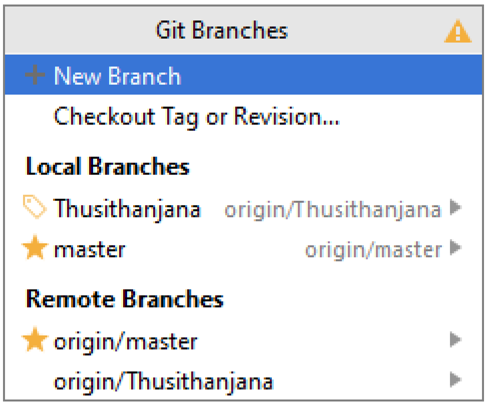
8. That user can do changes, add new content to the project, commit changes and push to the GitHub

Creating new branches in the repository: To avoid the interference with the main code. Right Click the project folder and follow the steps as follows

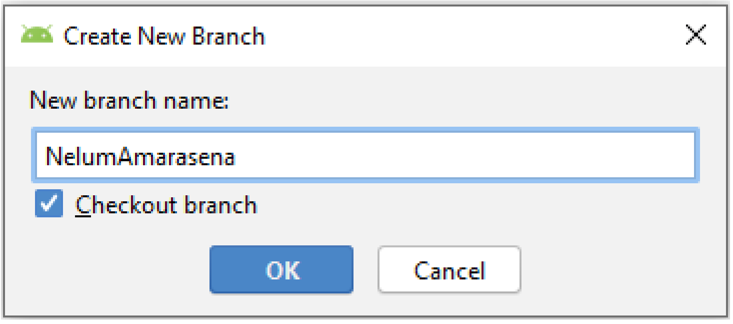


it translations for all locales in the translations editor.

```
<resources>
    <string name="app_name">GitLabReg2
    <string name="Title1">MAD Regular
    <string name="name1">This is the E
</resources>
```



9. Then give a name to the branch. Make sure you tick the checkout branch option. Then your changes will occur only in that branch.

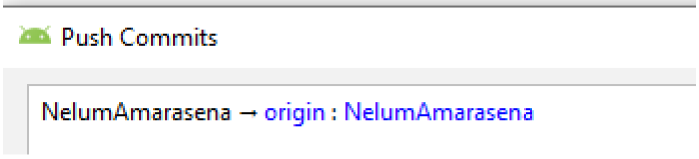


Create New Branch

New branch name: NelumAmarasena

☑ Checkout branch

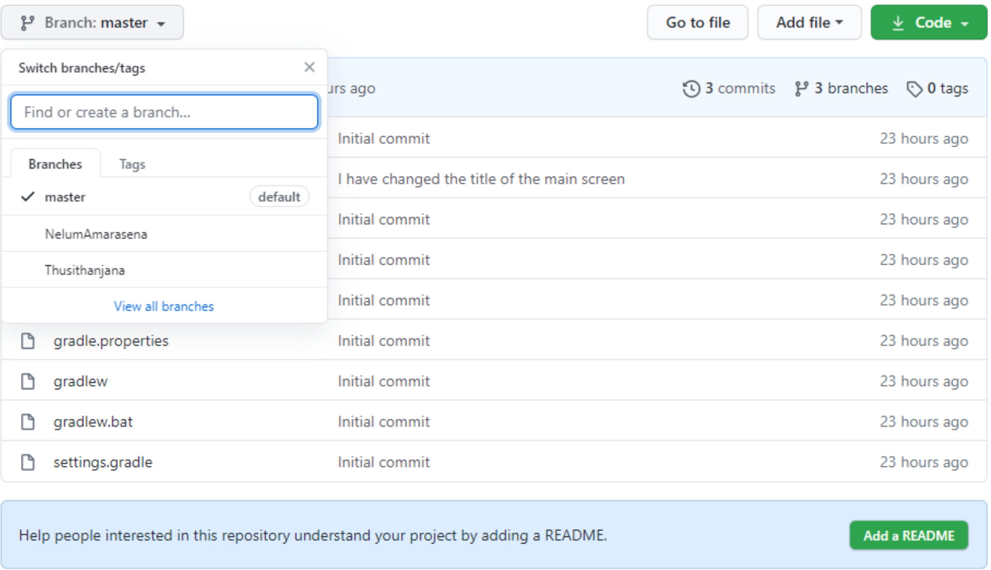
10. When you push the code to GitHub you should see the following



Push Commits

NelumAmarasena → origin : NelumAmarasena

11. After you push the new branch to the GitHub you should be able to see the branch like below.



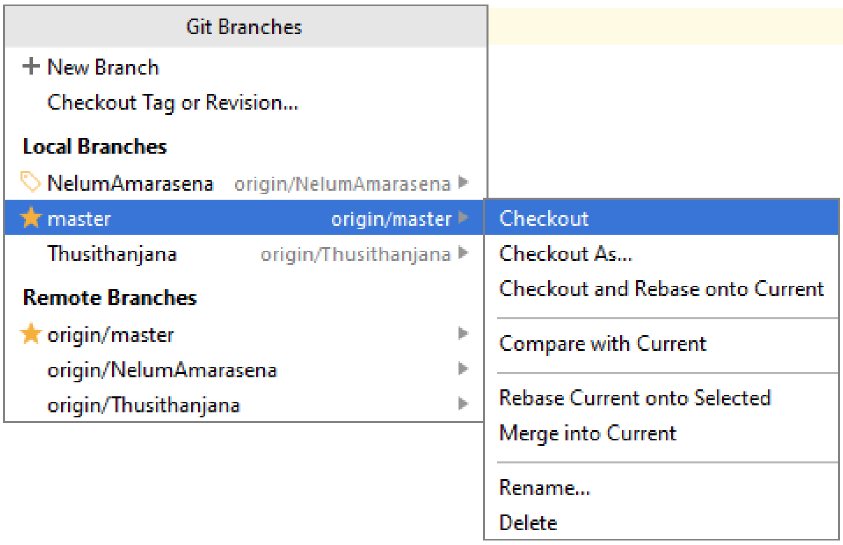
✓ master default
NelumAmarasena
Thusithanjana

urs ago 3 commits 3 branches 0 tags

	Initial commit	23 hours ago
	I have changed the title of the main screen	23 hours ago
	Initial commit	23 hours ago
	Initial commit	23 hours ago
	Initial commit	23 hours ago
gradle.properties	Initial commit	23 hours ago
gradlew	Initial commit	23 hours ago
gradlew.bat	Initial commit	23 hours ago
settings.gradle	Initial commit	23 hours ago

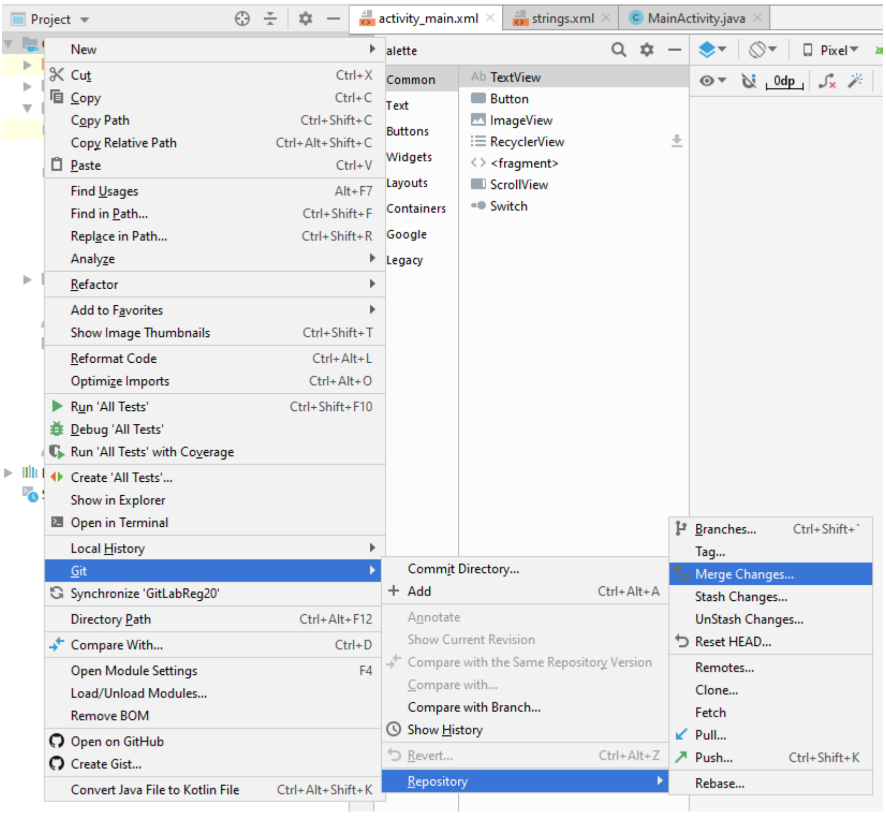
Help people interested in this repository understand your project by adding a README.

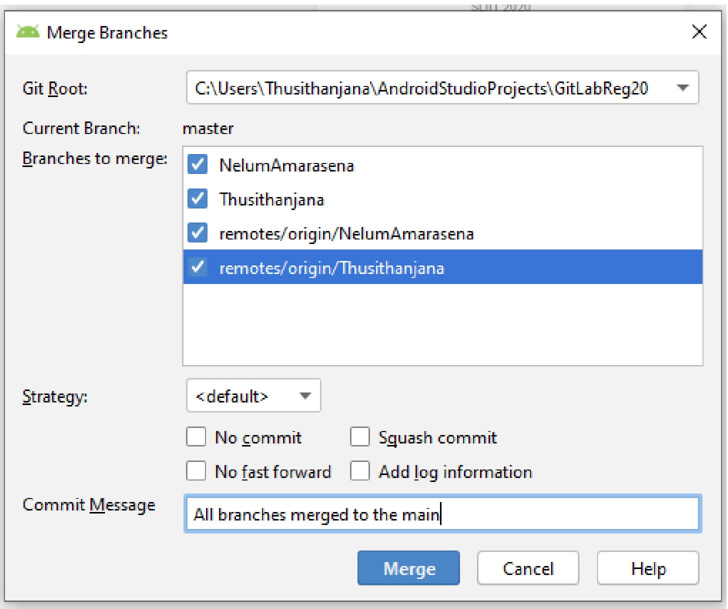
12. If you want to switch back to another branch you can follow the below steps



13. Then when you push the local repository it will go to the checkout branch.

14. To merge all the changes, go to the master branch and follow the below steps





Merge Branches

Git Root: C:\Users\Thusithanjana\AndroidStudioProjects\GitLabReg20

Current Branch: master

Branches to merge:

☑ NelumAmarasena
☑ Thusithanjana
☑ remotes/origin/NelumAmarasena
☑ remotes/origin/Thusithanjana

Strategy: <default>

□ No commit □ Squash commit
□ No fast forward □ Add log information

Commit Message: All branches merged to the main

15. Always remember to commit important changes and push to the GitHub
16. Put meaningful Commit messages