Step 1: Install eclipse IDE.

Install eclipse from

<https://www.eclipse.org/downloads/>

The latest version of eclipse is eclipse Neon (version 4.6).

If you already have eclipse but have a different version, you do not have to reinstall. Just go to step 2.

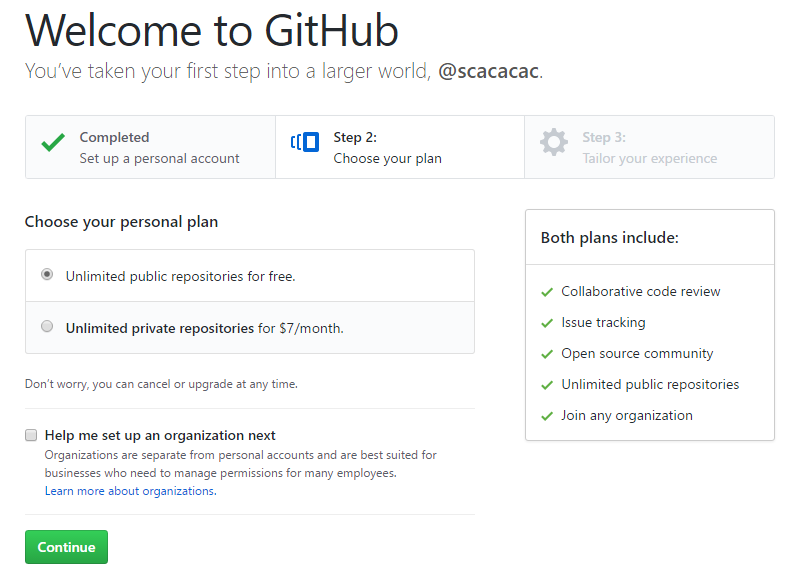
Step 2: Create a GitHub account.

Go to <https://github.com/> and create an account.

I recommend you pick a username that contains your real name.

Double check that your email address is correct. You will be asked to verify your account.

Choose the free plan.



Step 3: Do the GitHub Tutorial.

<https://guides.github.com/activities/hello-world/>

This is a direct link to the tutorial. It should take about 10 minutes. You do not need to use eclipse for this tutorial.

Step 4: Find my profile.

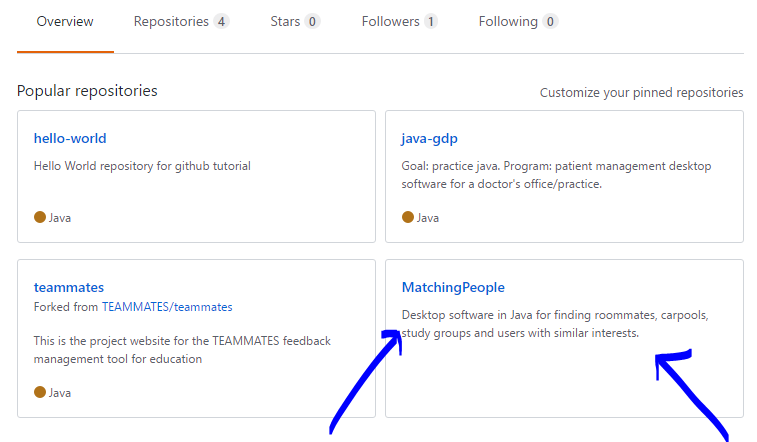
If you are not already signed in, sign into GitHub and click on the icon  to go to the home page.

Type “LakhouaMehdi” in the search bar and you should find my profile under “users”. If you do not find me you are using the wrong search bar and you are searching inside a repository only.

Here’s a direct link just in case.

<https://github.com/LakhouaMehdi>

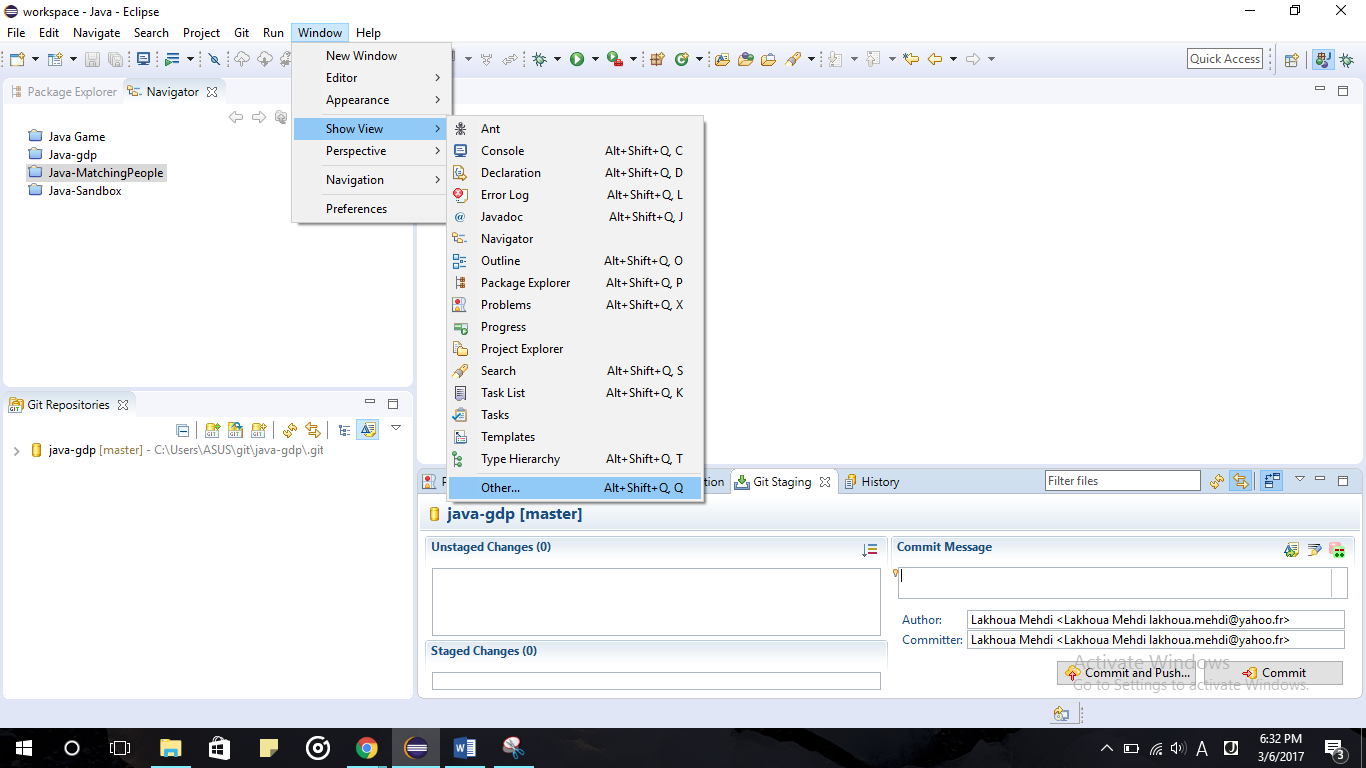
You should see one repository named “MatchingPeople”. Don’t open it yet.



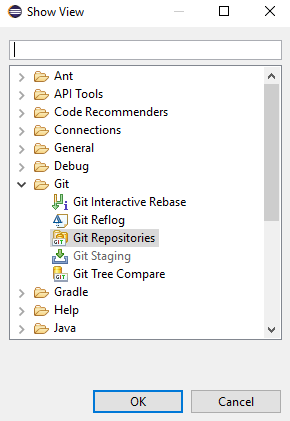
Step 5: Go back to Eclipse.

Open eclipse. You want to bring up some useful views before importing the project.

Click on “Window” > “Show View” > “Other”



This window will open.



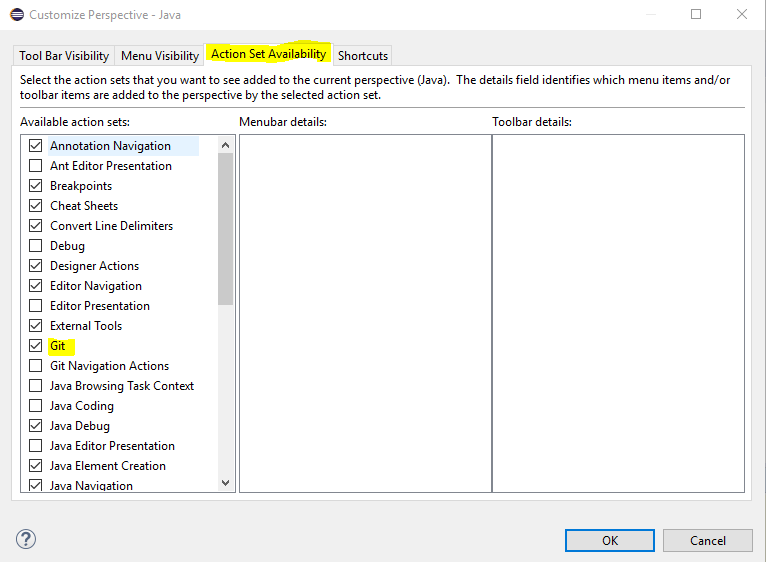
Add the “Git Repositories”, “Git Staging”, “History”, and “Navigator” views.

If you cannot find them, use the search bar.

Note: You can drag them around on your screen to change their location.

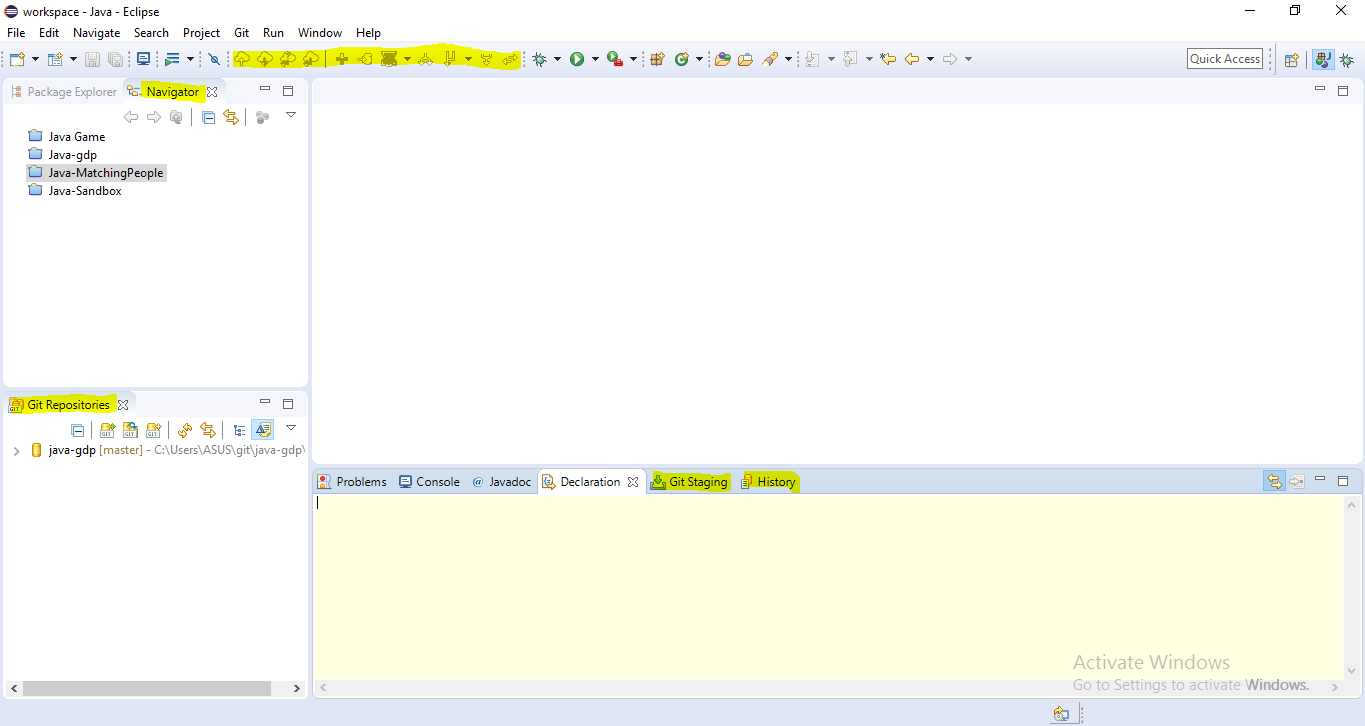
Next, add the Git toolbar.

Click on “Window” > “Perspective” > “Customize perspective”



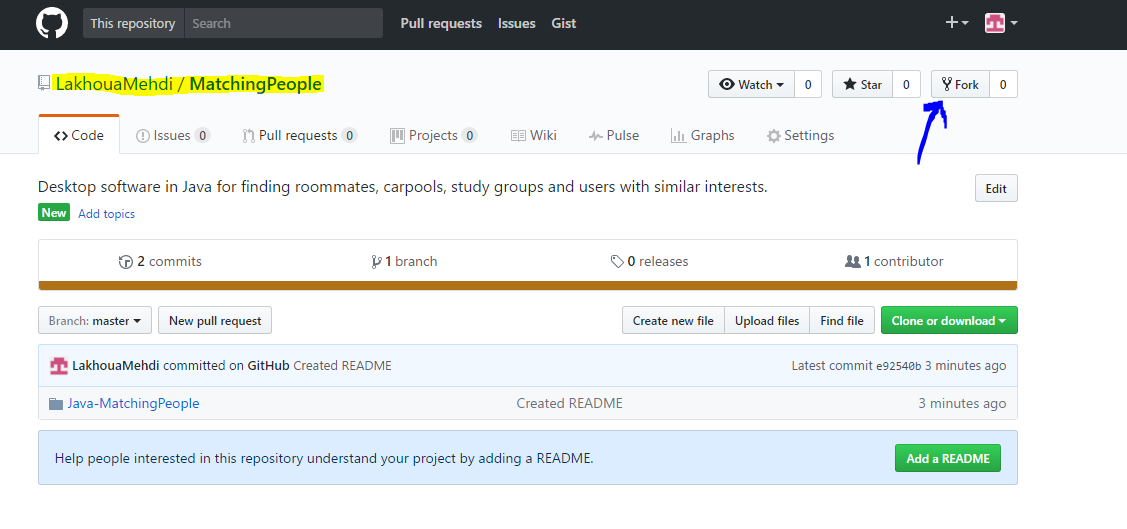
Under “Availability”, put a check mark on Git. This will add the Toolbar to your screen.

It should look like.



Step 6: On GitHub

Open the MatchingPeople repository and click on Fork on top of the screen.



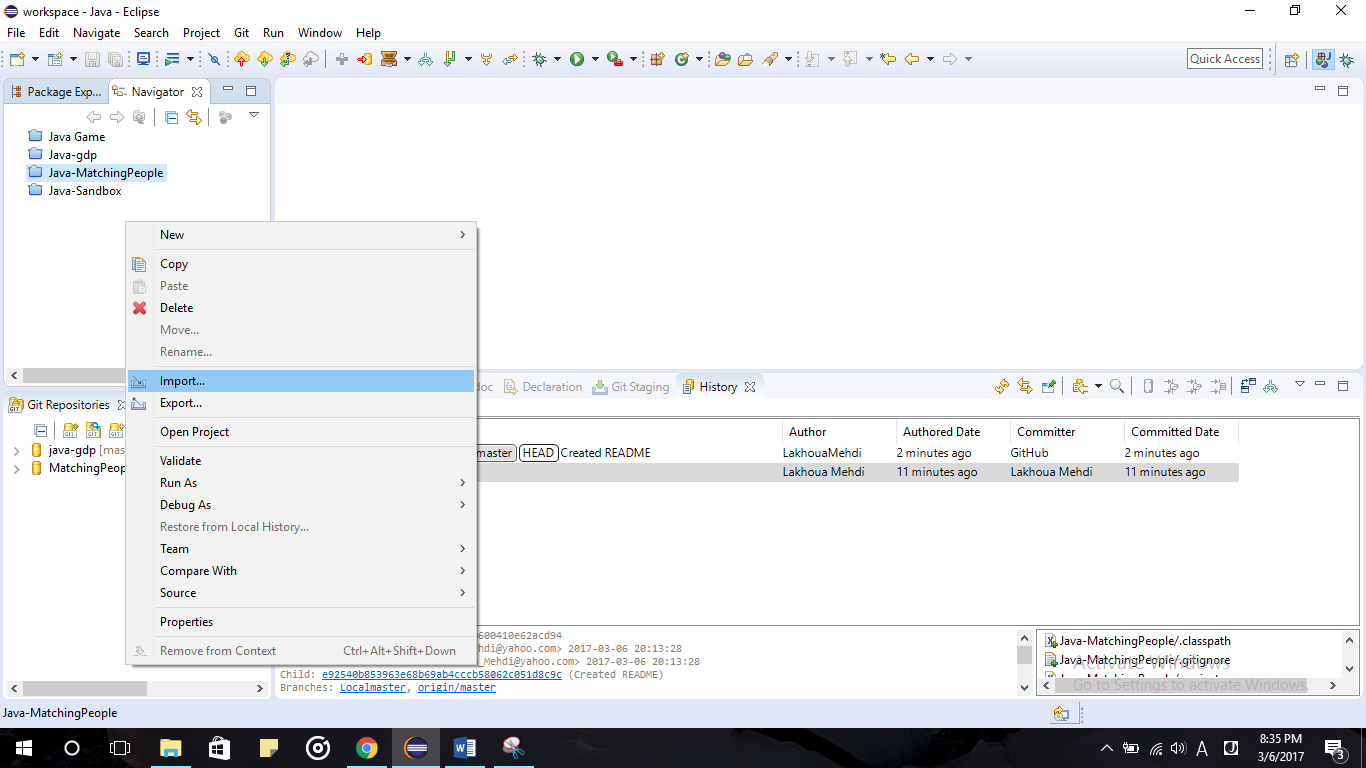
Now, go to your own profile. You should see that the project appears there too with the text “forked from LakhouaMehdi/MatchingPeople”. Open that. It is a duplicate of my project on your own profile.

Copy the URL of the page on top of the screen to your clipboard. It should be https://github.com/yourUsername/MatchingPeople.

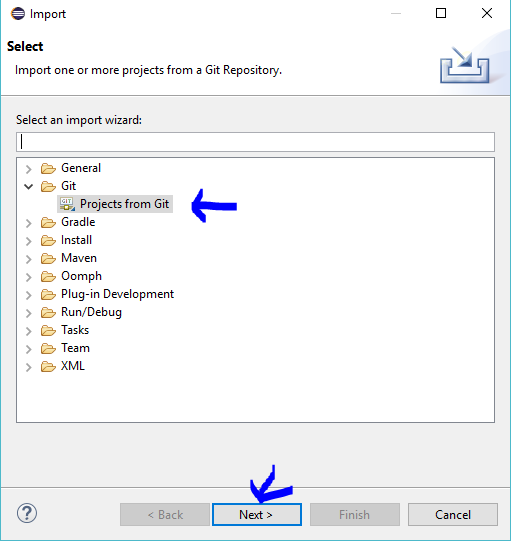
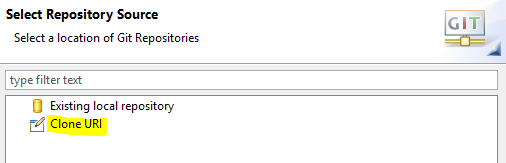
Note: Make sure you copy the URL of the forked repository. Not the one from my profile.

Step 7: On Eclipse

Right click on the navigator window, and click Import.



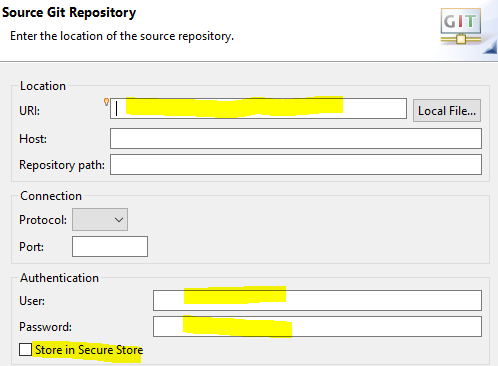
In the window that opens, select Projects from Git and click next. In the next screen select Clone URI



Paste what you copied in the URI field. The rest should fill up automatically.

Enter your GitHub account’s username and password under authentication.

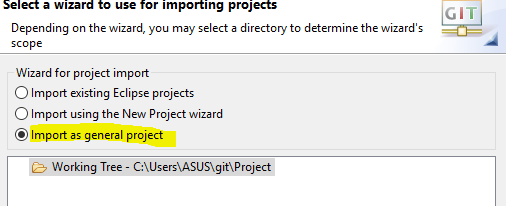
You should store your password and user in Secure Store so eclipse will remember it next time.



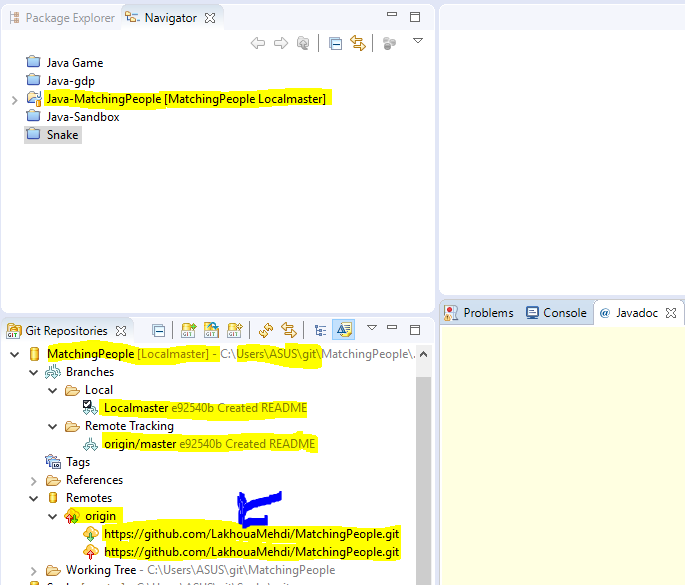
When the branch selection window shows up, select the master branch and click next.

You can keep the default directory and remote name “origin” and click next.

In the next screen, import the project as general project and click next the finish.



Now MatchingPeople should have appeared in your Navigator and Git Repositories views.



Note: The folder of the project will not be created in the folder “workspace” with your other projects. It will be at C:\Users\...\git\MatchingPeople.

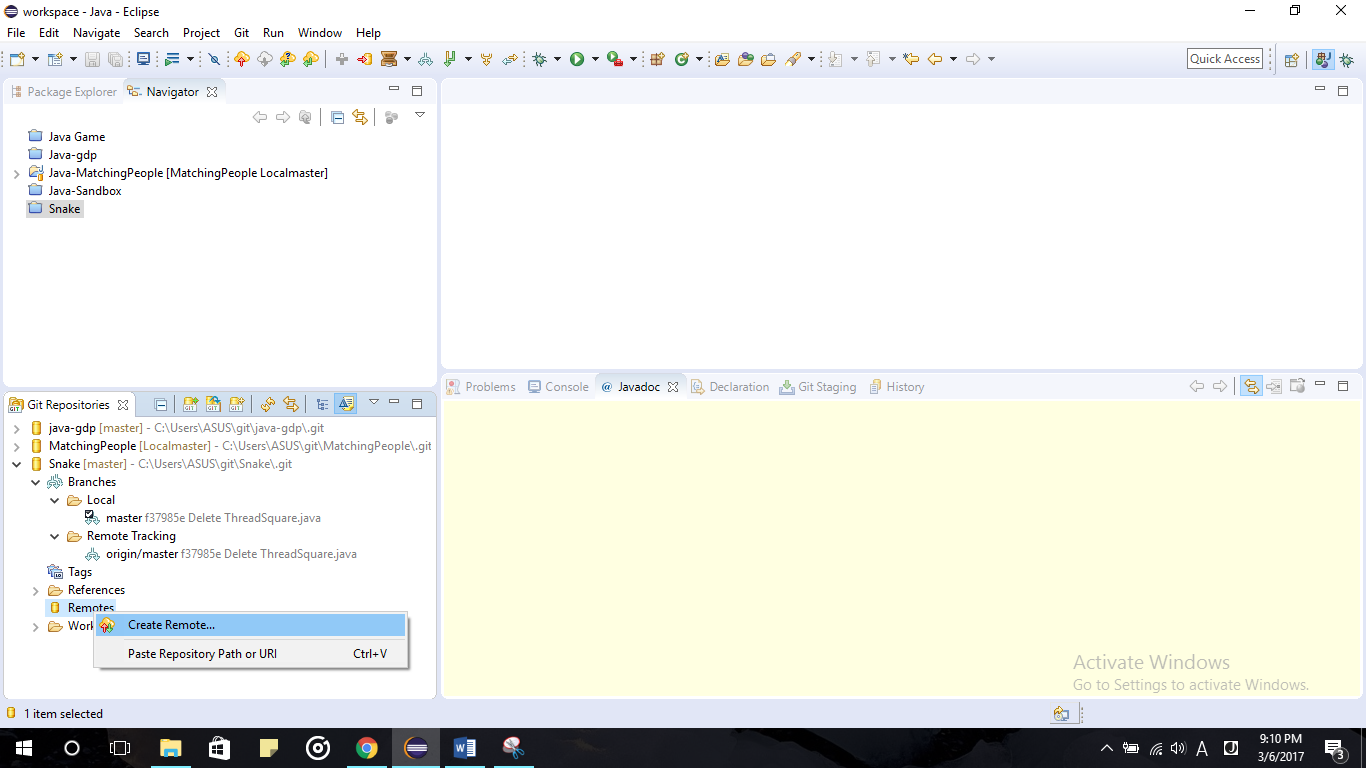
Under Remotes. There should be one remote named origin already added. This Remote accesses the forked repository that is on your GitHub profile.

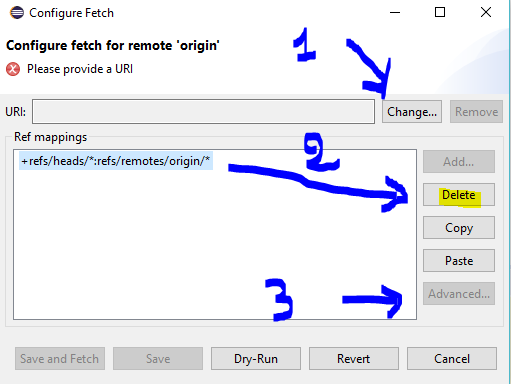
To make sure of that, you need to check that the name that appears in the 2 URL is your username, not mine. If the remote is pointing to my repository, you will not be able to push your changes.

If the remote has my name in it or if it is not there do Step 8. Otherwise go to step 9.

Step 8. ONLY IF REMOTE ORIGIN IS NOT THERE

First, right Click on origin. And click delete remote.

Then right click on Remotes and create Remote.

A window will open. Keep the remote name as default “origin”, Select Configure fetch and click ok.

A new window will open

First click on change

URI and reenter the URL

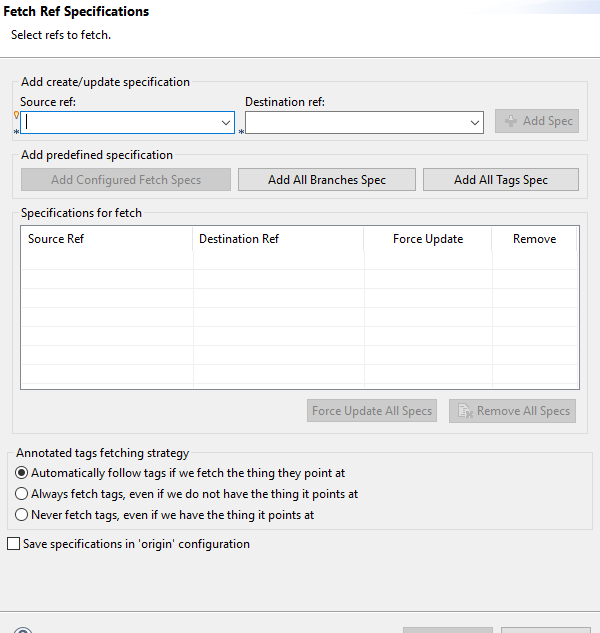
You copied earlier.

It has your username in it

Not mine.

You Will be asked for a

Password again.

Then delete whatever is under Ref mappings

Then click on advanced.

Under source ref,

Select master.

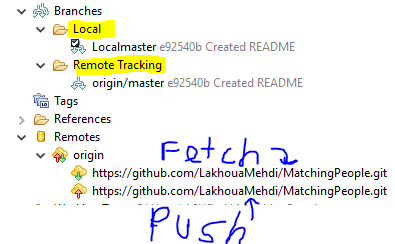
Under destination ref,

Select origin/master.

Then click on add spec.

Then finish and save.

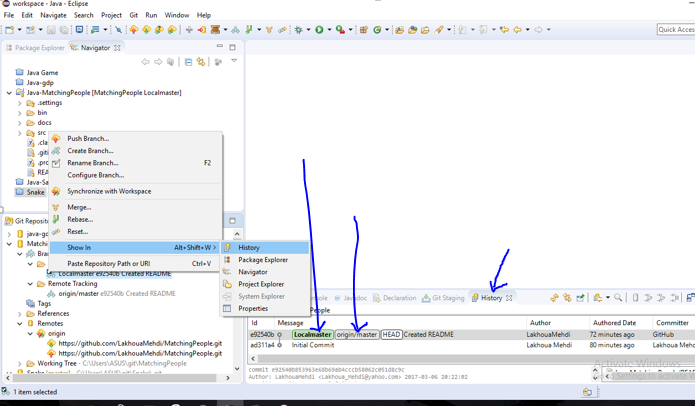
Configure push in the same way, by right clicking on the icon with a red arrow this time.

Step 9: What this is.

Under branches, you will see two branches. The one in local, contains the changes you made to the project and saved on your pc.

Ps: do not rename those if they do not match the names I have that’s fine.

The one under remote contains the changes you made to the project, saved on your pc, and pushed to your forked repository on GitHub.

Right click on the local branch then Show in> History.

This tab shows you the history of the project since its creation.

As time goes by, it will get more complicated. In the picture you can see that for now there are only 2 commits (changes) to the project since it was created. You can also see who the author of those changes is and from where it was committed. (I did the first commit from the IDE, the second from the website).

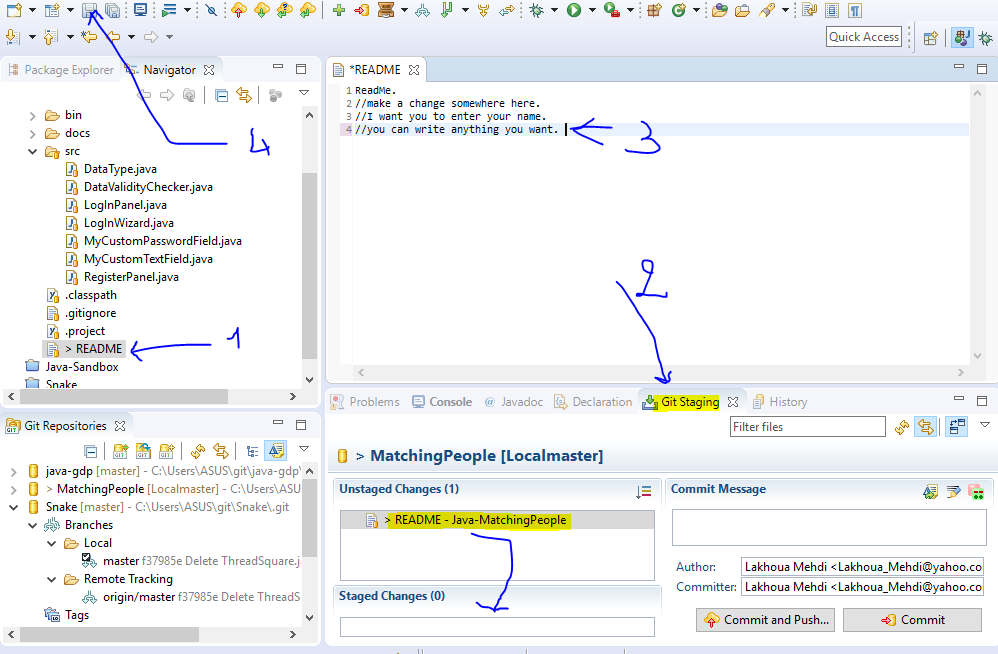
The green tag shows you where your local branch is at. The grey tag shows you where the repository on your profile is at. In the picture you can see that they are both updated with the latest commit. You can also look at the hash code next to every branch to know how up to date it is.

Step 10: Make your first commit

Run the project to make sure it works fine. The project contains some code that I wrote but you don’t have to look at it yet.

You’re going to make a modification to the file named README.

Double click to open it. Then bring up the Git Staging view.



Make a change to the file as instructed. Write your name.

As soon as you save the file, you will see any file that has changed pop up in unstaged changes. Drag the files you want to commit to staged changes.

Next, enter a commit message describing what modifications you have made.

In author, please enter your name and your email address like so.

Lastname Firstname <email>

Do not leave your ip address.

Keep an eye on the hash codes next to your local branch and the one next to the remote branch as you click on commit. (Don’t click commit and push you’re doing it in 2 steps this first time so you can learn. next time you can commit and push if you want). Click commit.

You will see the hash code next to your local branch has changed. You can pull up the history tab again and you will see it is now one commit ahead of the remote.

If you open your forked repository on GitHub, you will see that the change is not there yet. You need to push your local branch to the repository for the change to be there.

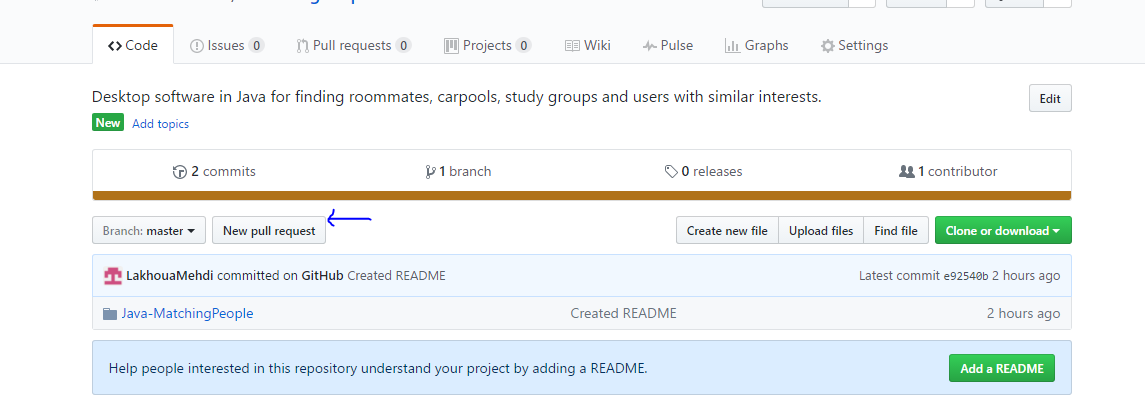
Right click on the local branch and click push, then next. You will see a recap of all the changes that are going to be pushed. Click finish.

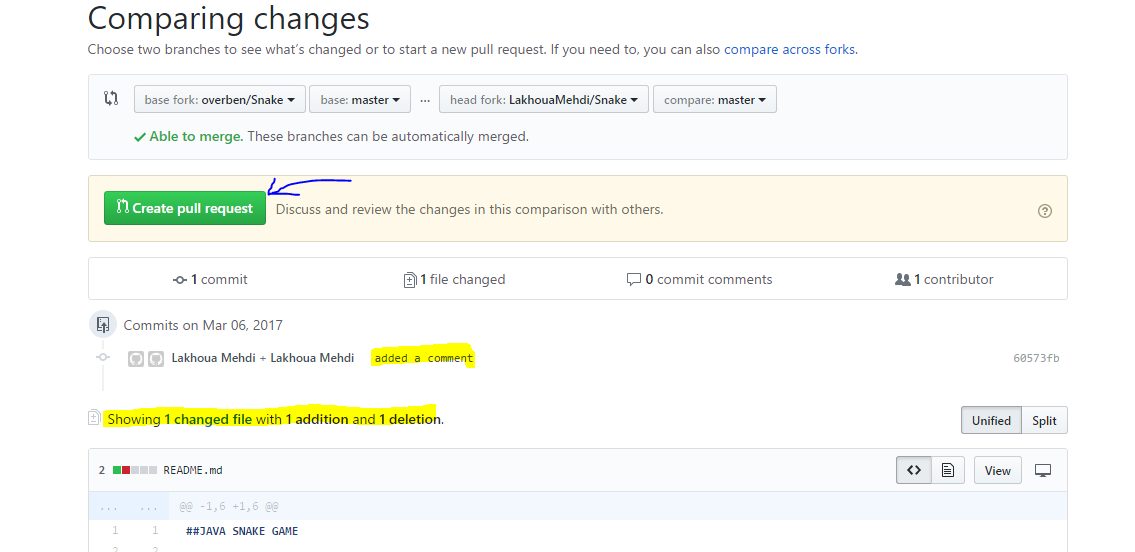
Now you can see the change appear in your forked repository on GitHub.

Step 11: Submit your changes to my repository.

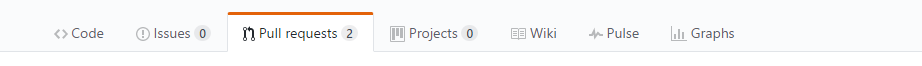
Note, you applied that changed to the forked repository that is under your profile. Not to my repository. If you open my repository you will see that it is completely unchanged.

You need to send me a pull request to ask that I pull your changes into my repository. To do that, go to your project on GitHub and click on new pull request.



Verify your changes then create the pull request. 

Now if you open my project, you will see that your pull request is there.



Everyone else can see the changes you added, and there is a discussion area included to discuss the changes.

If everything looks ok, I can pull the change into my repository and that’s when my repository is updated with your changes.

Step 11: To keep your local branch updated with mine.

Now, someone made a change to the project and I pulled it into my repository. You want to update your main branch so it has the latest modifications.

You need to create a new remote that points to my repository this time. However this time, you only need to configure the fetch, not push.

Refer to step 8 to see how. This time enter my repository’s URL .

When you commit and push your changes. Make sure you push them to your forked repository first, then create a pull request. Do not try to push to my repository it will not let you do that. That’s the problem we ran into this morning.

I think you can also use the same remote. Just configure the fetch to fetch from my repository. And the push to push to yours. But I haven’t tried that.

Step 12: Look at my code and get caught up with the project.

I tried to add as many comments as I could. To get a better understanding please look at the files

DataType.java and DataValidityChecker.java first

Then look at MyCustomTextField.java and MyCustomPasswordField.java

And run the program to see what that does.

Then look at the user interface.

THE END.