Project Guidelines

This document provides all information regarding the project rules, organization and deadlines. Hence, it is ***very important*** to read it carefully in order to know the rules and also to ensure a good progress of your project.

Contents

1 Project Overview

The complete description of the project can be found in the *Project overview* document available on our website (see: `http://doplab.unil.ch/eda-project/`).

2 Software/Hardware Requirements

For each phase, we will use the project tools presented at the beginning of the semester that you can find at this link: `http://doplab.unil.ch/wp-content/uploads/2018/02/IntroductionLabProjectTools_2018.pdf`. Additionally, we use *Git* as version control system. As Git clients, students can use *Netbeans built-in Git client* for Java code and *Xcode built-in Git client* for Swift code. Instead of using Netbeans and Xcode built-in Git clients, they can also decide to only use *SourceTree*. For more information about Git, please read Section 7 (Appendix).

3 Prerequisite

The prerequisite of this course is the course *Introduction to Distributed Systems (IDS)* (see: `http://doplab.unil.ch/ids/`). So, in order to perform the project, you need a good knowledge of Java distributed programming (for the backend implementation with Glassfish and for Android application if you decide to choose Android), average knowledge of Swift or a strong will to learn Swift and iPhone programming in a short time duration (about 2 or 3 weeks) if you decide to create an iOS application.

4 Organization

In order to do the project, students must form groups. Groups are formed at the beginning of the semester. Each group will create one Git repository on `https://gitlab.unil.ch/` where code and the documents of their project should be found at each phase. Please read carefully Section 7 (Appendix) to know more about the Git repositories and how you can use them for the development of you project.

For organizational reasons, such as ***creation of the Git repository***, some steps should be done. These steps are described below.

4.1 Step 1 - Course registration

All students, regardless of their university, must follow this link `http://doplab.unil.ch/eda-registration/` and edit the required fields of the Google document indicated. The deadline for the course registration is

Wednesday 28th February, 2018 at 23:59. EPFL students in particular, need to perform some additional steps for the registration to this course. They should go to this link (`http://ic.epfl.ch/cycle_master_en`), and then follow the instructions under the item *"In case you take a course outside the study plan at UNIL-HEC, ..."*.

4.2 Step 2 - Creation of a Gitlab account

During the project, you will use your Gitlab username/password to connect to your project repository on `https://gitlab.unil.ch/` and to be able to create the Git repository needed. Depending on you university, there are two types of Gitlab accounts:

- **Students who have a UNIL email address:** if you have an email address at UNIL, you already have a Gitlab account. Your Gitlab username/password are the same as the one's used to connect to your UNIL mailbox at `https://owa.unil.ch/`.
- **Other students (including EPFL students):** if you do not have an email address at UNIL, you will create your Gitlab on this website: `https://gitlab.unil.ch/users/sign_in`. A confirmation email regarding the creation of the Gitlab account should be sent to your academic mailbox. The confirmation email contains your Gitlab username/password.

4.3 Step 3 - Group formation

Each group should be composed of 3 or 4 members. Then, each group must do the following:

1. Each member of the group should activate her/his Gitlab account (you can skip this step if you have already activated your account on Gitlab). In order to do so, each member should open: `https://gitlab.unil.ch/` in a web browser, in the window "Sign in" ("LDAP" for UNIL students and "Standard" option for other students - see Figure 1), login with Gitlab username and password (see Section 4.2 if you do not know what are your Gitlab username/password). If you do not have a Gitlab account yet, please follow the instructions in Section 4.2. In case you have any problem to login with your Gitlab account's username/password, please contact `Michel.Schuepbach@unil.ch`.
   Once logged in, you can change your password and then close the browser window. Note that ***it is very important that each group member does this process***. In fact, it enables you to create your group Git repository on Gitlab. You will use your group Git repository for the project.

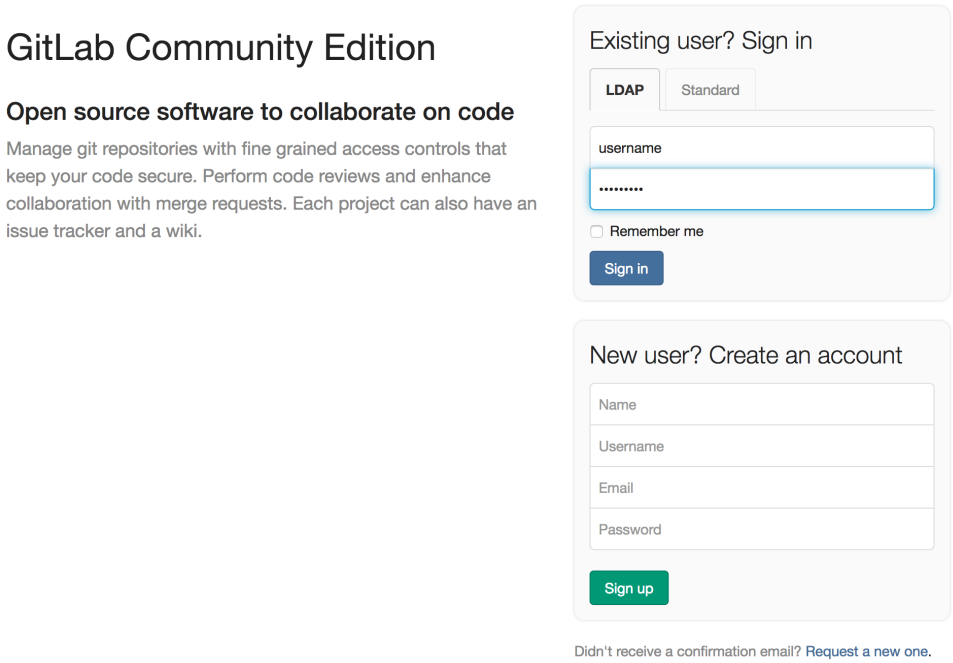

Figure 1: Account activation on Gitlab

2. Once the above (activation of an account on Gitlab) is done for all group members; each group should send to `vaibhav.kulkarni@unil.ch` and `arielle.moro@unil.ch` an email until **Monday, 5th March, 2018 at 23:59**, and announces its group formation. Of course, one email per group is sufficient. Your email should respect the following format:

   - **Email object:** eda18groupformation
   - **Email content:** the list of the group members as well as a name for your group. For each member you should indicate: (first name, last name, academic email address). If you are less than 3 students in your group or if you are alone and you are not able to find any group, please indicate shortly this problem in your email.

Note that your emails only will be taken into consideration if they ***respect the above mentioned format*** and if they are sent ***before the deadline***.

4.4 Step 4 - Git repositories creation

Once Step 1 and Step 2 are done, each group will be able to create its own Git repository (see Section 7.2 below).

5 Submission deadlines

At the end of each phase of the project, your project code (Java and/or Swift) and the PDF format of your presentations slides should be found on the Git repository of your group before the indicated deadline. These deadlines are:

- ***Phase 1:*** Wednesday, 14th March, 2018 at 23:59 (slides only).
- ***Phase 2:*** Wednesday, 11th April, 2018 at 23:59 (slides + code).
- ***Phase 3:*** Wednesday, 30th May, 2018 at 23:59 (slides + code).

Note that we only check your Git repository on `https://gitlab.unil.ch/` and on no other personal repositories. Furthermore, ***you do not need to provide a report for the project*** but we advise you to comment your code.

6 Presentations

At the end of each phase of the project, you have to do a short presentation explaining the work you have done. To know the presentation dates, please check the calendar on our web site [1].

The exact time and place for each group presentation will be announced by email a few days before each presentation. The presentations will be closed and private (like an oral examination) and will take place every 15 minutes (the last 5 minutes are for our questions). The presentations can be in English or French. All group members should be present during the presentation of their group. It is preferable that each group member presents a part of the presentation (1 or 2 slides or at least a part of a slide). In the case where a member is absent, she/he should send us an email explaining the reason of the absence. If you take this course, we assume that you keep free your Thursday mornings for this course. Please do not send us an email and request to change the schedule. We can not change the schedule of the presentations according to the personal availability of the students. However, if you have a valid reason to be absent and if your group mates do not have a problem with your absence, you can inform us by sending an email and can be absent during the presentation. In this case, you will be graded as the others.

The first presentation will include the three following parts:

- **First part(about 10 min.):** presenting the slides. The slides should contain:
  - Title and context of your project: you mainly describe your motivation to define a new application;
  - Goal and a short description of your application;
  - Architecture of your application: including Android/iOS application + MatchMore service + backend (with Glassfish);
  - Description of the main building blocks of your application: e.g., Android/iPhone client, Java Server, MatchMore service, etc...
  - Communication between building blocks;
  - List of functionalities (or how you will use it) provided by each building block;
  - Work distribution among group members: you should precise which part of the application will be developed by which member.
- **Second part (about 5 min.):** Questions & Answers.

The last two presentation will include the three following parts:

- **First part (about 5 min.):** a demo of your application. The demo can be done on your laptops or one of the Mac (Internef 143) computers. However, if you are an EPFL student we advise you to use your laptops for demos because your access to room Internef 143 is only guaranteed during the exercise sessions. Note that for the demos, all elements of your application (e.g., Android/iPhone application, backend, etc...) can run on the same machine (localhost).
- **Second part (about 5 min.):** presenting the slides. All the following slides should present:
  - Architecture of your application: main building blocks of your application (e.g., web tier, business tier, iPhone client, etc...);
  - Communication between building blocks (e.g., HTTP, JMS, etc...);
  - Work did by each member of the group.
- **Third part (about 5 minutes):** Questions & Answers.

Some advice for your presentations: (1) Do not repeat the demo in the slides by using screen shots and (2) get prepared (so we do not lose time).

7 Appendix

7.1 Git

Throughout this project, you should use Git as your version control system. A version control system manages all changes occurring to a software over time, that is, as the project is worked on and developed. Thus, a version control system allows more than one person to work on a single project. It also provides a complete history of all files, which allows the project to be backtracked and comparisons made with previous versions.

Git can be used in the following modes (for more information see `https://www.atlassian.com/git/tutorials/comparing-workflows`):

1. Centralized workflow,
2. Feature branch workflow,
3. Gitflow workflow,
4. Forking workflow.

One of the advantages of using Git over other version control systems such as subversion(SVN), is that Git enables a user to save locally all the work progress and a history of it. In addition, as already described, we can use Git in different modes depending on our way of working.

In this project, you can use command line to use Git or a Git client called SourceTree that will be presented later.

7.2 Gitlab web interface and Git repository creation

For this project, one member of the group can now create the two Git repositories on `https://gitlab.unil.ch/`. Once logged in with your with Gitlab username and password (see Section 4.2 if you do not know what are your Gitlab username/password), you can see your dashboard and create the two repositories by clicking on **+New Project** as indicated in Figure 2.

Thus, for this course you will have to create **only one private repository** per group, called "Project" on the website, having the following name format "eda18-*groupname*". This project must contain 3 folders: 1 folder for each phase of the project containing all the code you will write in Java and/or in Swift as well as another folder containing all your presentations (slides).

After the creation of the repository on Gitlab, you will be able to see your project on your Gitlab dashboard and you will also be able to easily add all the remaining group members to the project by clicking on it and on **Members** on the left menu of the projet. If and only if the all the group

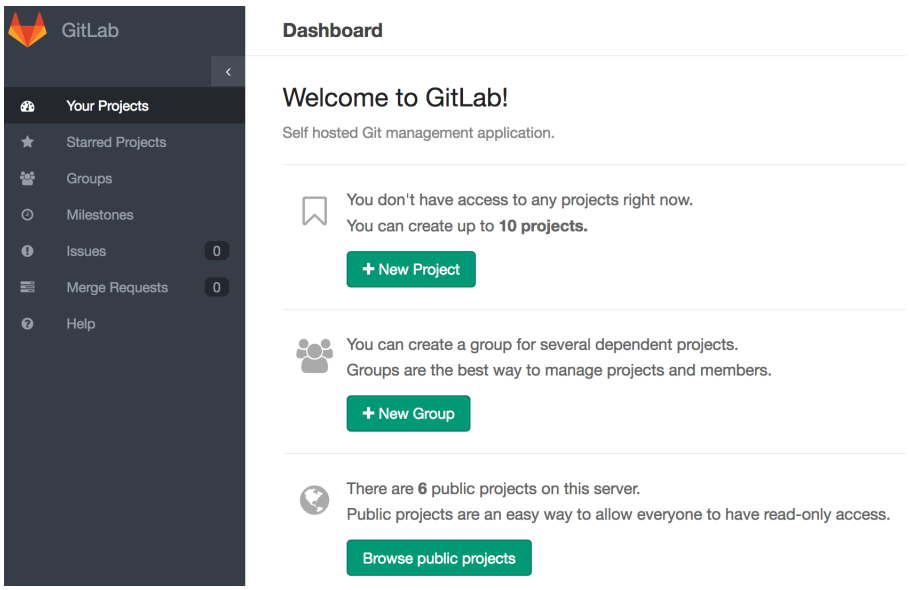

Figure 2: Gitlab dashboard

members already have or have created their Gitlab account as previously indicated, they can be added to the repository (or "Project"). You also need to add the professor and the teaching staff to this Git repository as members of the project because we need an access to check your work at the end of each phase.

7.3 Using Git with Sourcetree

In this section, you will discover how to use Git with Sourcetree, which is a Git client [3]. Firstly, a member of your group creates a folder on her/his computer and upload it on your group repository on Gitlab. Then, the rest of group members download the project that is uploaded on your group repository on their computers. During the project, you save and share your work using Git commands. Below, we describe each of these steps in more details.

7.3.1 Create a project and upload it in your remote Git repository

1. Create a new folder on your computer that will contain all the 3 folders of the course mentioned above,

2. Open *Sourcetree* and click on "Add a local repository via Sourcetree" as described in Figure 3,

3. Select the path of your folder created before, select the type "Git" and click on "Create" as depicted in Figure 4,

4. Right-click on "Remote repositories" and on "Add a remote repository" in the window of your Sourcetree project as described in Figure 5,

5. Put the name of the remote repository ("origin" by convention) and its URL as depicted in Figure 6,

6. Create a little README text file in your local folder and commit it in Sourcetree by clicking on "Commit" as described in Figure 7,

7. Select the first commit you did previously and click on "Push" in Sourcetree as depicted in Figure 8,

8. Select the "master" branch and click on "OK" in Sourcetree as described in Figure 9.

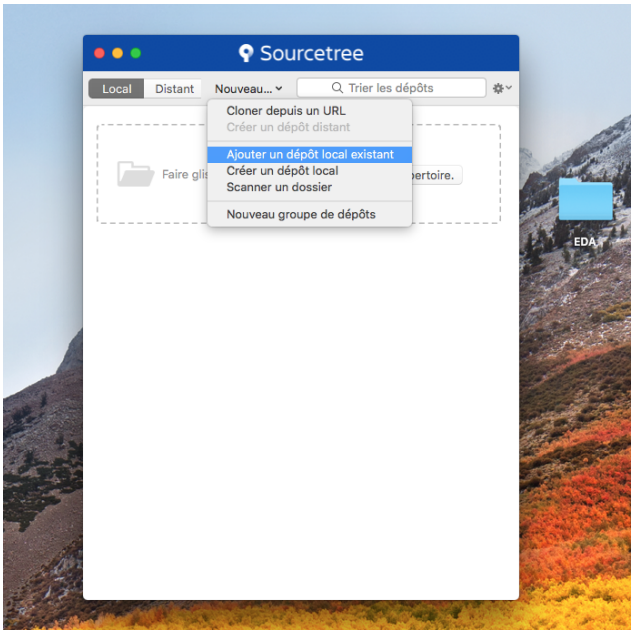

Figure 3: Add a local repository in Sourcetree - Step 1

At the end of this process, you should be able to see your README file on the Gitlab website in the "Files" section. Now, we will see how to clone the project on a computer for the remaining members of the group.

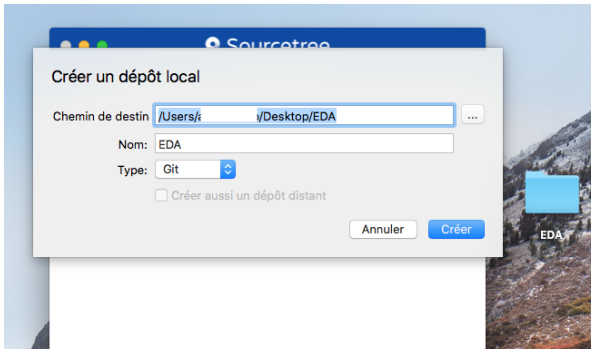

Figure 4: Add a local repository in Sourcetree - Step 2

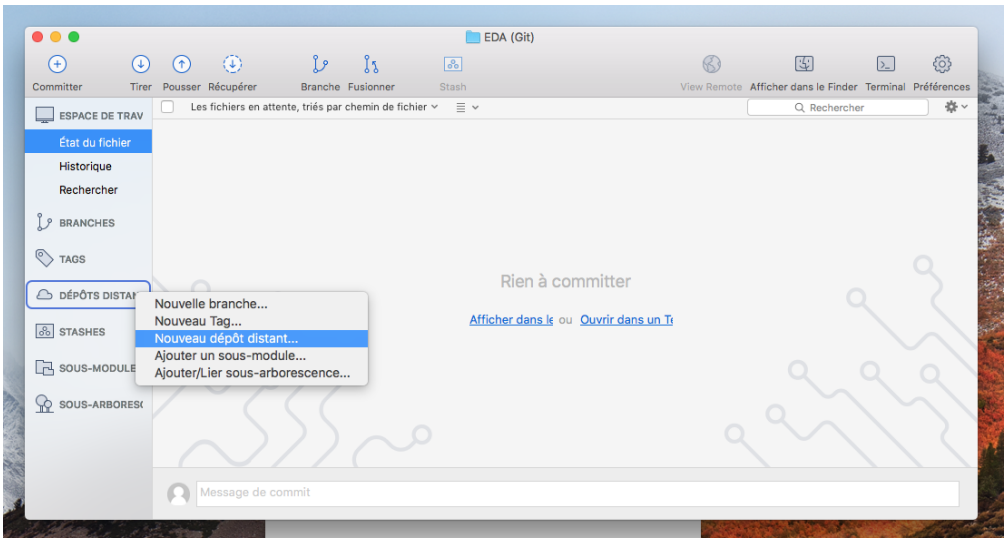

Figure 5: Set the remote repository (1) in Sourcetree - Step 3

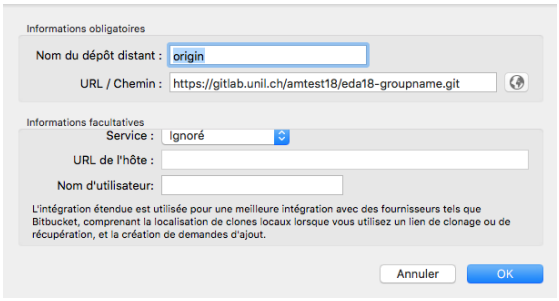

Figure 6: Set the remote repository (2) in Sourcetree - Step 4

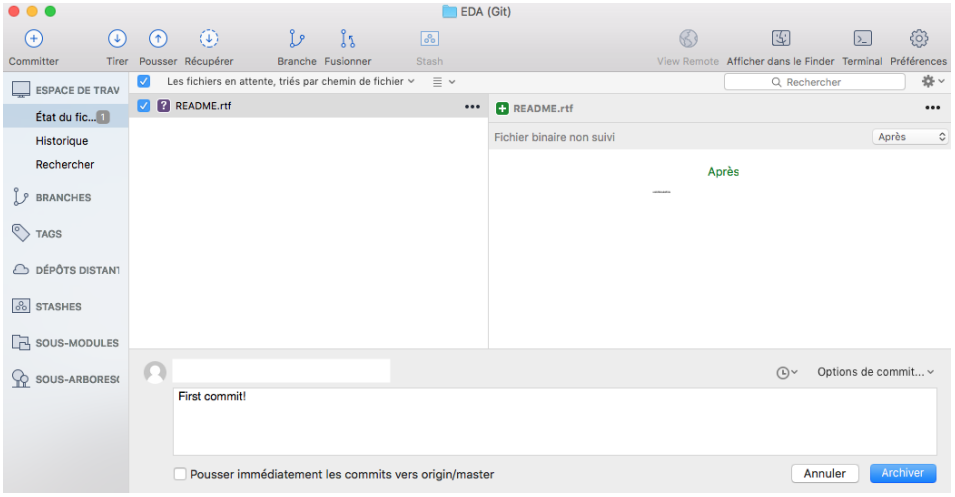

Figure 7: Commit in Sourcetree - Step 5

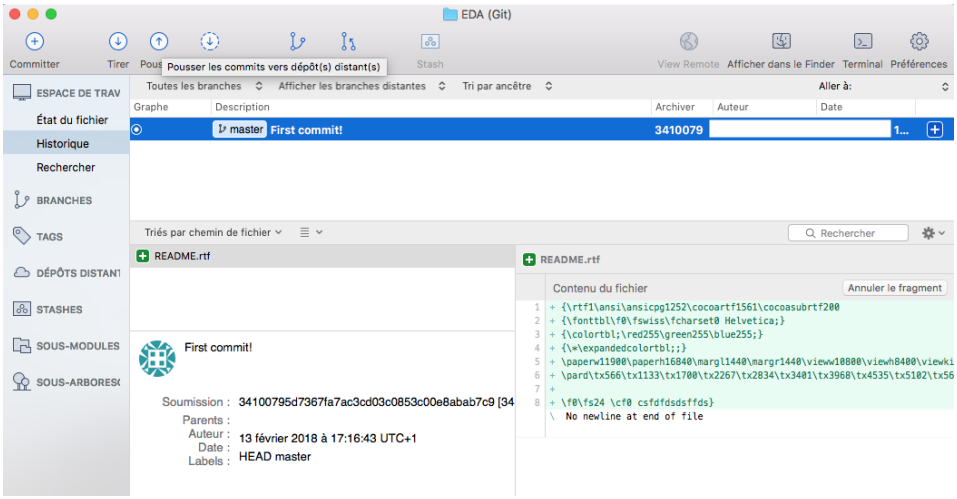

Figure 8: Upload in Sourcetree - Step 6

7.3.2 Download a remote Git project on your computer

As described in Figure 10, you will click on "New" in the main window of Sourcetree and click on "Clone from URL". Then, you will indicate the same URL previously given in step 4, the path of your targeted folder on your computer and click on "Clone".

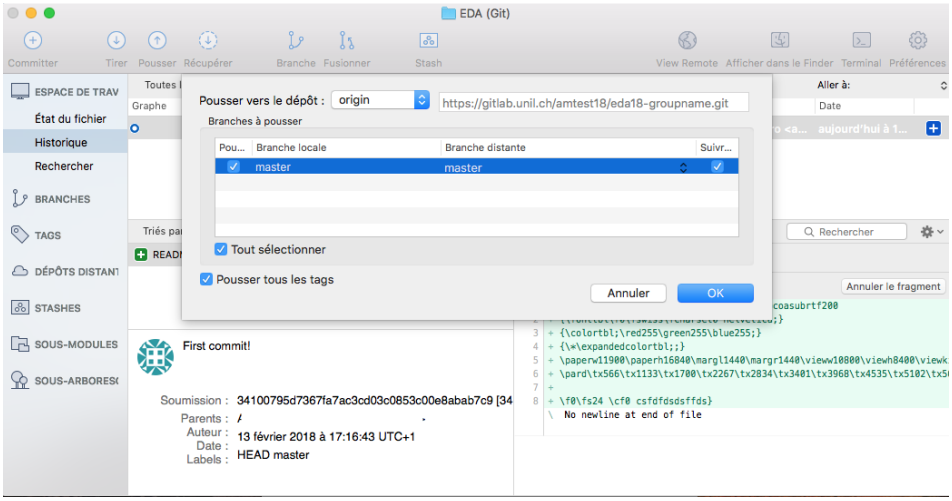

Figure 9: Set the remote repository (2) in Sourcetree - Step 4

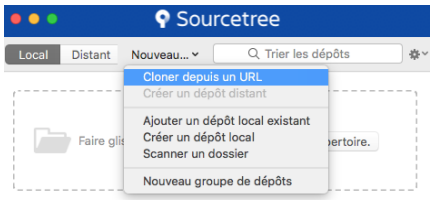

Figure 10: Clone a remote Git repository on your computer via Sourcetree

7.3.3 Save and share your work

In order to save and share you work, you will use three main Git commands: commit, push and pull as depicted in Figure 11. It is important to not edit a same file at the same time by two or more members because this can create Git conflicts you will have to manage. Consequently, you work on your files, you commit and push your modifications on the remote repository (i.e., same steps as previously presented in Section 7.3.1). Every time you do a modification, you can tell the other members to download the new version of your files by clicking on "Pull" in Sourcetree.

References

[1] EDA Calendar.
http://doplab.unil.ch/eda/

[2] Git workflows.
https://www.atlassian.com/git/tutorials/comparing-workflows

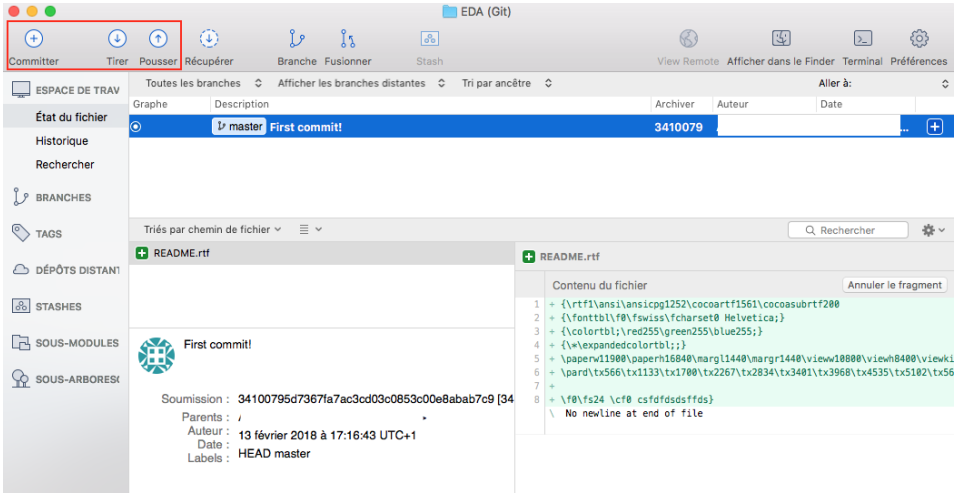

Figure 11: Commit, push and pull via Sourcetree

[3] SourceTree application.
https://www.sourcetreeapp.com