**Software Development and Testing IT7320**

**LAB WORK**

Team Members: Submitted to:

Anjali Khandelwal

Kwino

Ralph Saplan

Thanh Huynh

Contents

1. Understanding Version Control (Anjali)

2. Surveying the GitHub Platform (Kwinno)

3. Creating a Hosted Repository (Thanh)

4. Touring Content Versioning on GitHub (Anjali)

5. Acquiring Repositories Locally (Ralph)

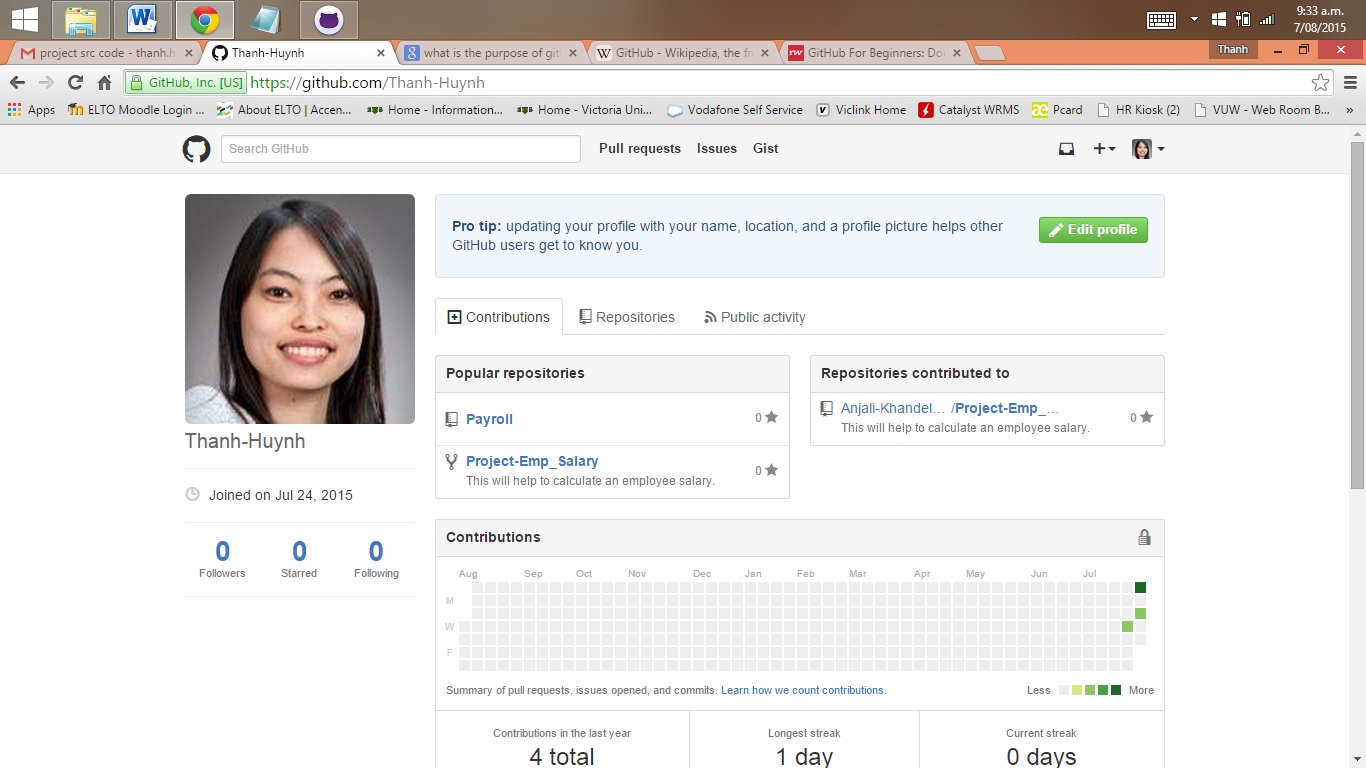
# Understanding Version Control (Anjali)

# Surveying the GitHub Platform (Kwinno)

# Creating a Hosted Repository (Thanh)

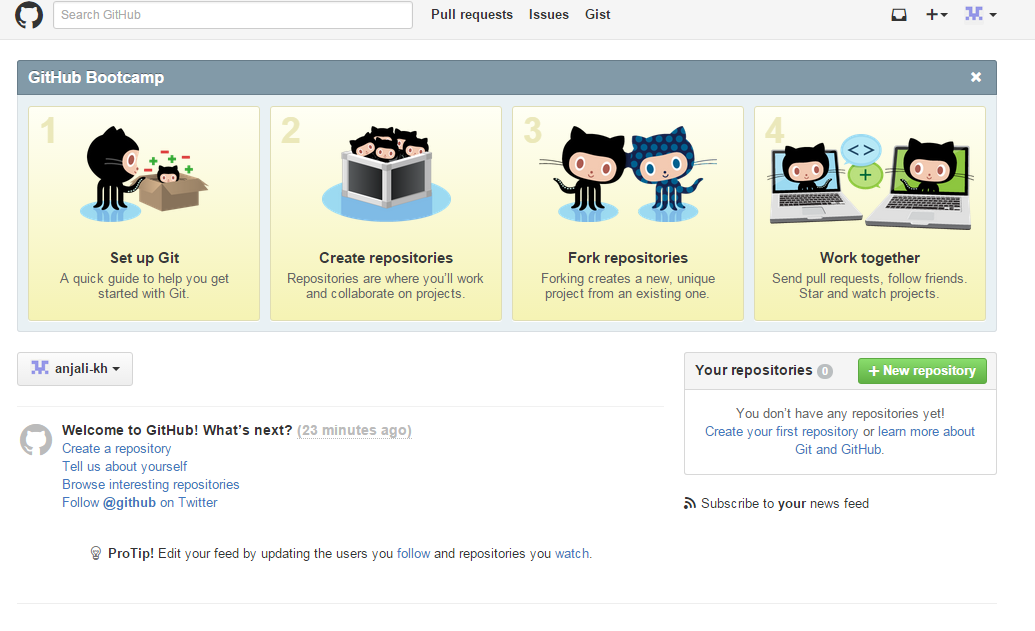
## Visit GitHub.com

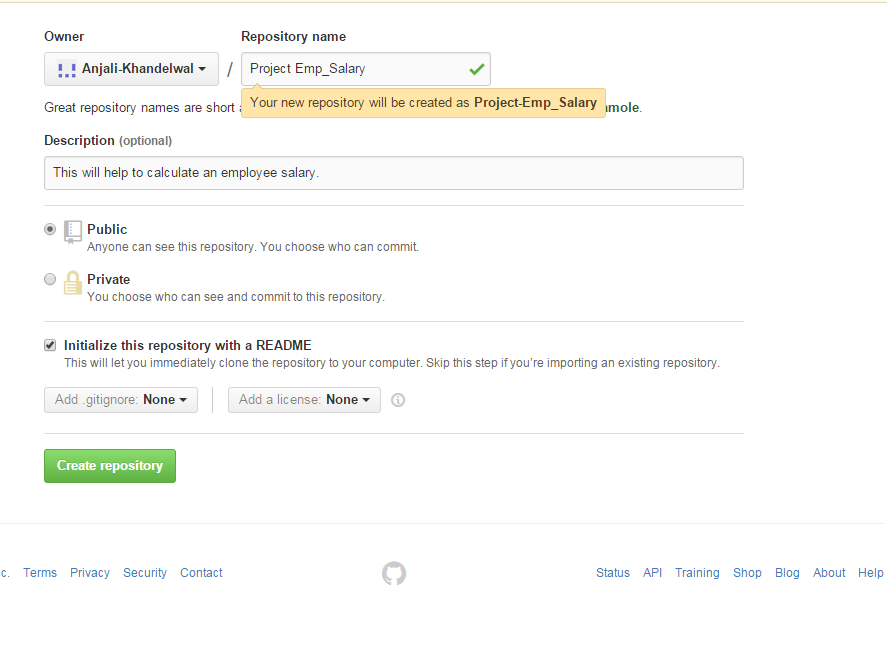




* + Go to the website GitHub.com website
  + Sign up an free account with username (Name to be recognized by team member), email address and password
  + The reason for having a GitHub account:
    - This is free for an unlimited quantity of public repositories.
    - This is a place where we build a profile, upload a project to share connect with the team members by “following” their accounts
    - GitHub can store any type of files like programs, code, text documents and so on in the project folders.
    - It is a public place where everyone can access and see your work. However GitHub does make sure the ownership is well retained. It states in their Term of Service that *“We claim no intellectual property rights over the material you provide to the Service. Your profile and materials uploaded remain yours.”*

## Create a repository and ways to interaction

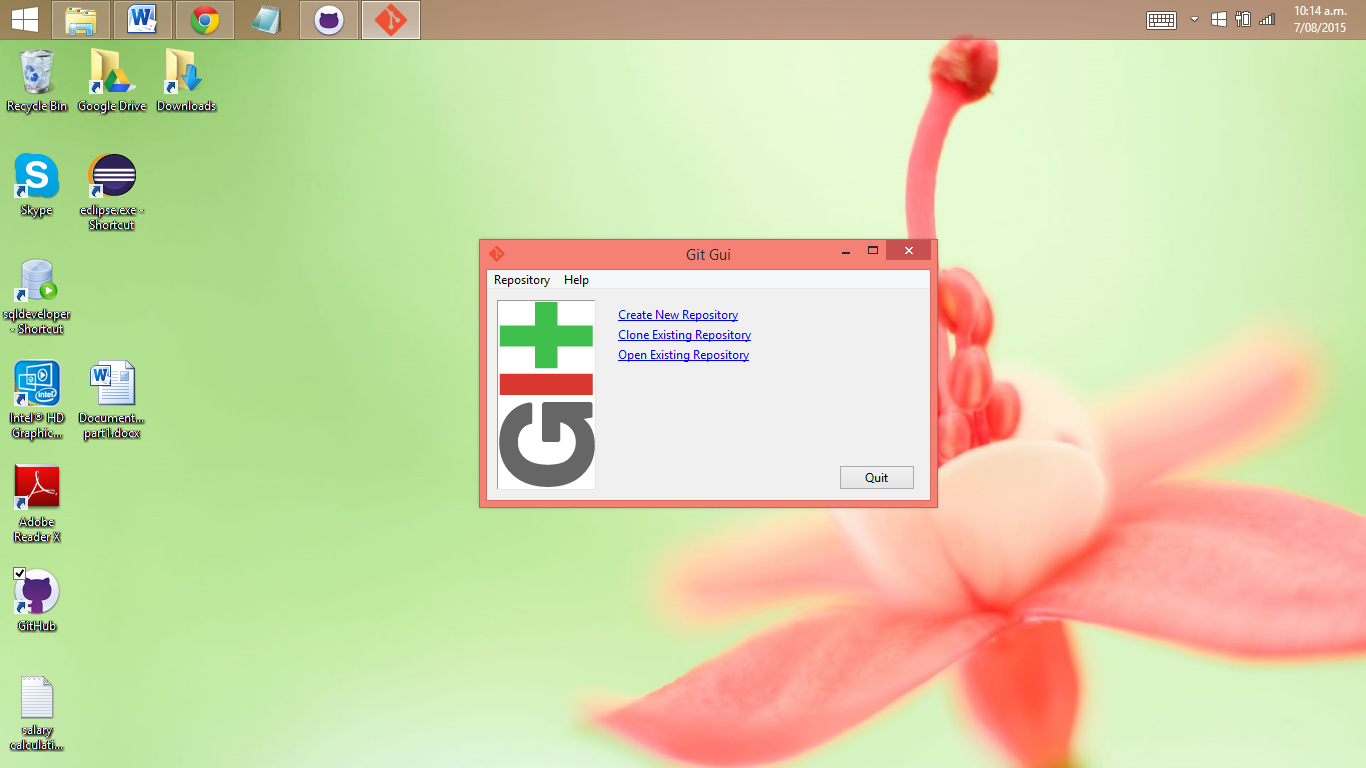




* + Create a short and meaningful name for your project repository. This is to help yourself and your team members to understand and remember. Our repository is **Project-Emp\_Salary**
  + Add a description of the project repository: **This will help to calculate an employee salary**
  + Choose public repository because it is free and can be accessed by everyone.
  + Select Initialize this repository with a README
  + Click Create repository. Now we are ready for the first commit

The above is one of the ways to create a repository using GitHub website user interface. There are other ways to create a repository such as:

**Using GitHub GUI**



**Using Git init at the command line**

There are two approaches:

1. Takes an existing project or directory and imports it into Git.

We will create a new Git repository skeleton \*.git by go to the project’s directory and type”$ git init”.

At this point, nothing in your project is tracked yet. To begin tracking those files and do an initial commit by

$ git add \*.c

$ git add README

$ git commit -m 'initial project version'

At this point, you have a Git repository with tracked files and an initial commit.

1. Clones an existing Git repository from another server.

If there is a project you’d like to contribute to — the command you need is “git clone [url]”. This will help Git to copy of all data that the server has.

**Using Eclipse**

Download EGit at <http://www.eclipse.org/egit/download/>

An Eclipse plug-in to use the distributed version control system Git.

The toolbar entries allow you to add an existing local Git repository to the view, clone a Git repository and to create a new Git repository.

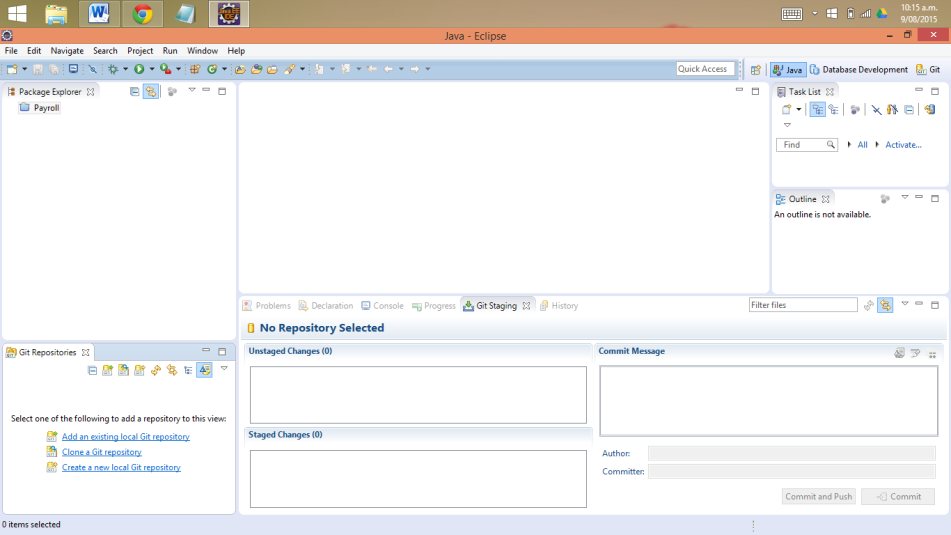
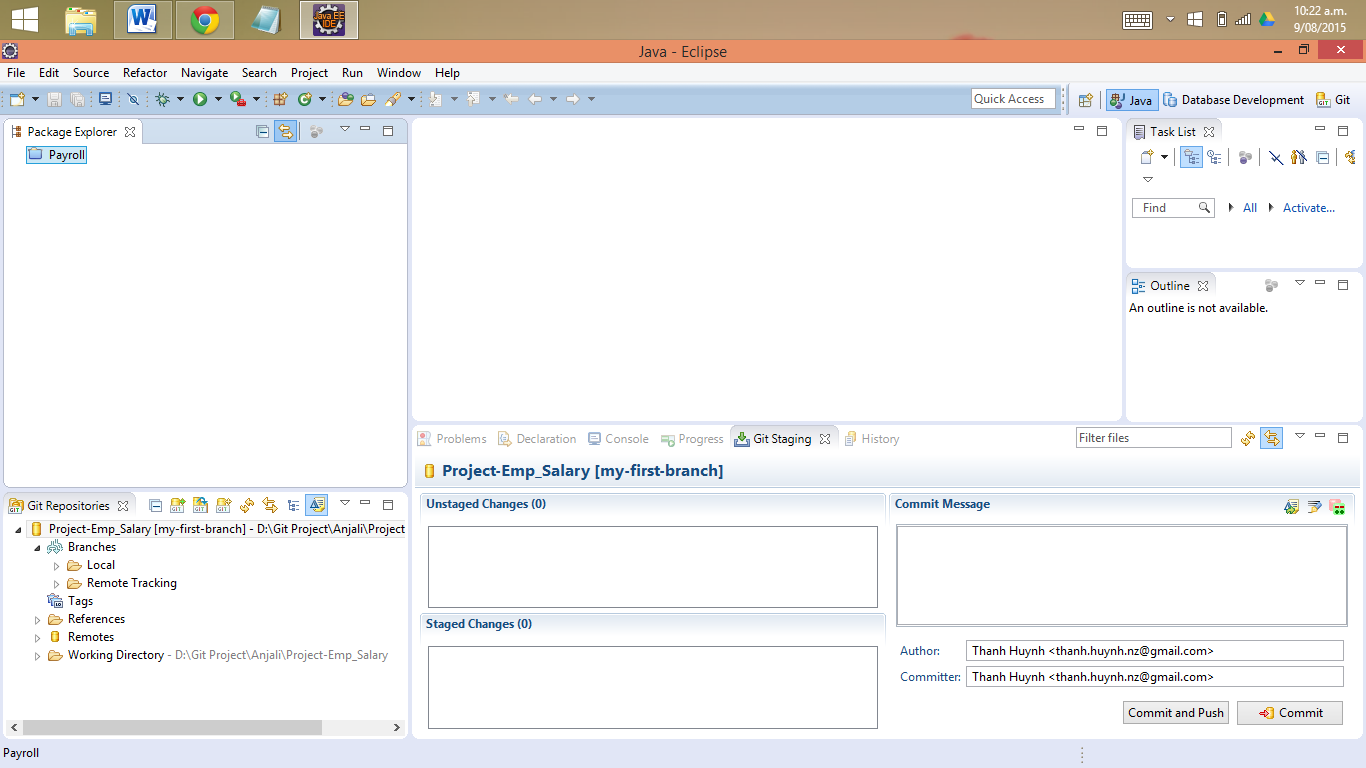
There is a recommendation to place your Git repositories outside the Eclipse workspace. This separates your Git repository from any additional meta-data which Eclipse might create.

Once you have added an existing local Git repository (Project-Emp\_Salary). It will show under the Git repositories view (as below). A right-click any element in the Git repositories view allows you to perform related Git operations. For example if you right-click on a branch you can “checkout” the branch or delete it.

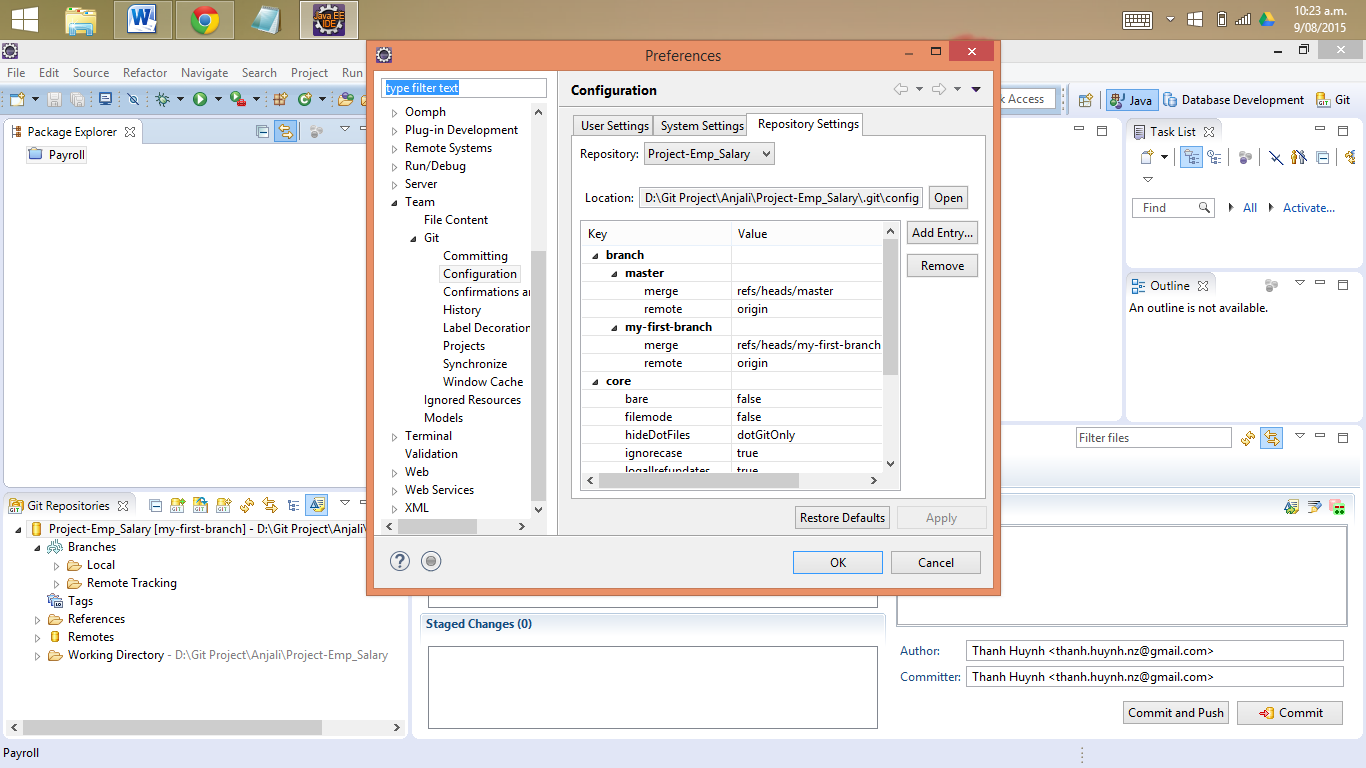
The Git Staging view is non-modal, you can switch between different repositories or even restart Eclipse without losing a commit message and it allows incremental staging for changes.

Open the Git Staging view via the Window → Show View → Other → Git → Git Staging menu.

Also you can choose any file which has changed and move them into the “Staged Changes” area. To commit the staged changes you write a comment message and select the “Commit”



The Git configuration settings can be adjusted via the Eclipse preference setting. Select Window → Preferences → Team → Git → Configuration to see the current configuration and to change it.

You can add entries to your Git configuration by pressing the Add Entries button on the Git Configuration preference page.