Assignment #4

Due Date: 4/11/21 by 11:59pm

Deliverable:

Post your homework as a SINGLE PDF file on Blackboard with the name "HW4_YourLastName"

Important Notes:

- Do NOT communicate or share your assignment with others
- Do NOT share your personal laptop with your classmates

High-Level Requirements:

1. The assignment will be completed in 3 phases.
2. All your deliverables will be based on work you have done on GitHub.
3. Phase-I is due on 4/5/2021 by 9:00am
4. Phase-II is due on 4/7/2021 by 9:00am
5. Phase-III is due on 4/11/2021 by 11:59pm

Phase-I Requirements:

1. You have to create your GitHub account id as instructed in the tutorial.

- For a student with the name John Doe with A number A12345678.
- His id will be JSP21SCM78D.
- In this ID J is the first character of his first name.
- SP21SCM will be common for each student.
- 78 is the last two numbers of his A number.
- D is the first character of his last name.

Once the TA receives all of your GitHub accounts you will be added as Collaborators to the master repository. For this, the TA will be sending out a mail to get all your created GitHub accounts. Use your hawk id while registering for your GitHub accounts. Do not use newly created email ids just for the assignment as GitHub may consider these id's as Bots and remove them. You will be proceeding to the next step only after the TA sends out a mail confirming that you have been added as a collaborator.

2. Next you will be forking the repository SCM587SP21 from SPM587SP21 GitHub organization to the GitHub account that you have created. This repository can be opened with the following URL.
   https://github.com/SPM587SP21/SCM587SP21.git

3. Now using the URL given above you will login to the repository listed above. You will go to the issues tab and create a new issue.
4. The **title** of the issue will be as listed below.
   "<Your GitHub id> adding first name"
   Example – "JSP21SCM78D adding first name"
5. This issue will be **labeled** using the labeling scheme listed below.

6. **Labeling Scheme**: Your Labeling Scheme follows the **key:value** format "**LabelName:*LabelValue***" for every label

7. **Types of Labels**: There are different types of labels, the following is the list of labels used for issues listed that you will use based on the **first digit of your A-Number**:
    i. **OriginationPhase**:
        1. **If the first digit is odd number:** use one of the values {Coding, Testing, Documentation, Field}
        2. **If the first digit is even number:** use one of the values {Requirements, Design }
    ii. **DetectionPhase**:
        1. **If the first digit is odd number:** use one of the values { Coding }
        2. **If the first digit is even number:** use one of the values {Design}
    iii. **Priority**:
        1. **If the first digit is odd number:** use one of the values { Major , Low}
        2. **If the first digit is even number:** use one of the values {Medium, Critical}
    iv. **Status**:
        1. **If the first digit is odd number:** use one of the values { Rejected, Completed }
        2. **If the first digit is even number:** use one of the values { pendingReview , Approved, inProgress, }
    v. **Category**:
        1. **If the firs digit is odd number:** use one of the values { Enhancement, Inquiry }
        2. **If the first digit is even number:** use one of the values {Bug }

8. You will now clone the repository that you have forked into your account on to your local machine. This repository will contain files a.java, b.java, c.java, …, z.java. You will open the file that has the first character of your first name. Once you open the file you will enter your GitHub account id in a new line and save it.

   Example – For John Doe with GitHub account JSP21SCM78D, he will open j.java and add his GitHub account username in a new line and save it. So j.java will contain JSP21SCM78D.

9. Now you will do a commit and then push this repository into your GitHub account. While making the commit you have to add your Issue number in the commit message in the following format.

Example – git commit –m “Fixes #<IssueNumber>. <GitHub username> first name added.”

Say for John Doe the issue he has created is Issue #24. His commit message will be as follows.

```
git commit –m “Fixes #24. JSP21SCM78D first name added”
```

10. Once that is done you will create a pull request to the master repository so that your changes can be merged onto the master repository. You will just create the pull request. It will be the TA who will accept your changes. Your pull request title should also be the exact same as your commit message.

    This will complete your Phase 1. You will also need to submit two screenshots as the commit report for each phase. The last page in this document instructs you on how to capture those screen shots. Both the screenshots that you have captured need to be put in a pdf called Report_Phase1_<GitHubUsername>.pdf for your phase I. Once your changes are all accepted in the master repository you will receive an email to commence Phase II. Then you will be following the Phase II instructions. Do not start Phase II till you have received the email to proceed with Phase II from the TA.

Phase-II Requirements:

1. You will open the master repository from your GitHub account and create a new issue.
2. This issues **title** will be as listed below.
   "<Your GitHub id> adding last name"
   Example – "JSP21SCM78D adding last name"
3. This issue will be **labeled** using the labeling scheme listed below.
4. **Labeling Scheme**: Your Labeling Scheme follows the **key:value** format "**LabelName:*LabelValue***" for every label
5. **Types of Labels**: There are different types of labels, the following is the list of labels used for issues listed that you will use based on the **last digit of your A-Number**:
   i. **OriginationPhase**:
      1. **If the last digit is odd number:** use one of the values {Requirements, Design }
      2. **If the last digit is even number:** use one of the values {Coding, Testing, Documentation, Field}

   ii. **DetectionPhase**:
      1. **If the last digit is odd number:** use one of the values { Testing }
      2. **If the last digit is even number:** use one of the values {Field}

   iii. **Priority**:
      1. **If the last digit is odd number:** use one of the values {Critical, Major }
      2. **If the last digit is even number:** use one of the values { High, Low, Medium}

   iv. **Status**:
      1. **If the last digit is odd number:** use one of the values { Approved, Rejected, Completed }
      2. **If the last digit is even number:** use one of the values { inProgress, pendingReview }

   v. **Category**:
      1. **If the last digit is odd number:** use one of the values { Bug, Enhancement }
      2. **If the last digit is even number:** use one of the values {Inquiry }

6. Now you go to the forked repository in your account and update the repository. This is done using the following command.

   ```
   git remote add upstream <Master Repository Git URL>
   git fetch upstream
   git rebase upstream/master
   ```

7. Update your local clone to the latest in your repository.

8. Now you will open the file with the first character of your last name. Then you will be adding your GitHub account username in a new line

   Example – John Doe will be opening the file d.java and add his GitHub account username which is JSP21SCM78D in a new line.

9. Now you will do a commit and then push this repository into your GitHub account. While making the commit you are to add your Issue number in the commit message in the following format.

   Example – git commit –m “Fixes #<IssueNumber>. <GitHub username> last name added.”

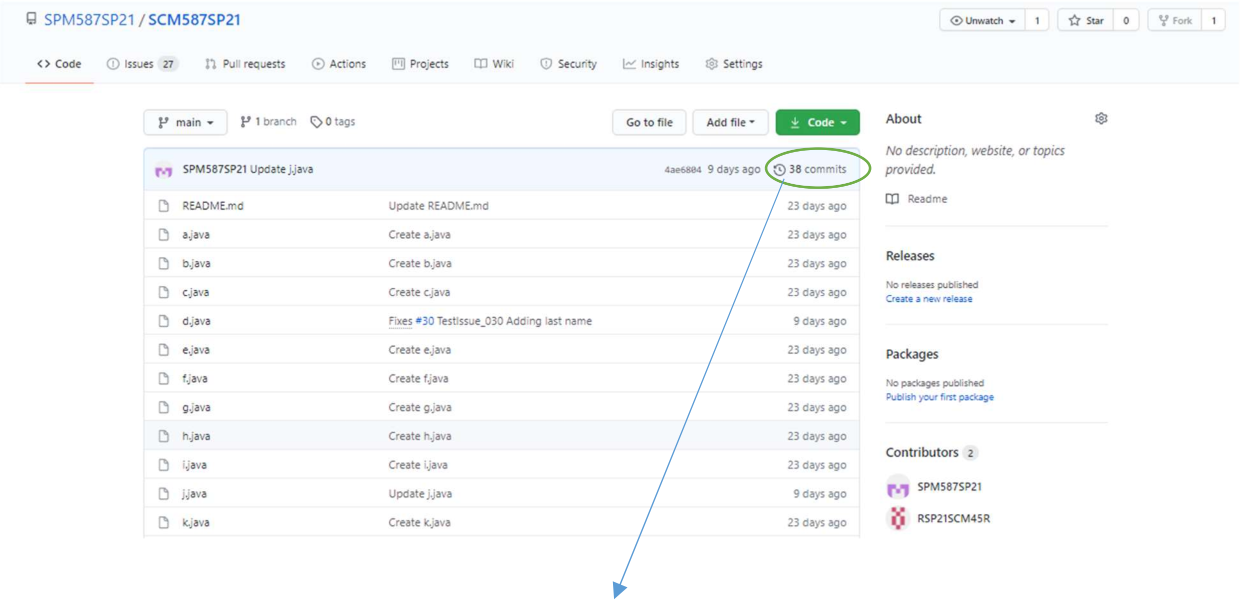
   Say for John Doe the issue he has created is Issue #65. His commit message will be as follows.
   git commit –m “Fixes #65. JSP21SCM78D last name added”

10. Once that is done you will create a pull request to the master repository so that your changes can be merged onto the master repository. You will just create the pull request. It will be the TA who will accept your changes. Your pull request title should also be the exact same as your commit message.

    This will complete your Phase II. You have to capture the same screenshots mentioned above for phase II and attach it in a document called Report_PhaseII_<GitHubUsername>.pdf and submit it.

Deliverable for Phase-II :

1. Login to your GitHub account and open the repository that you have committed to. Click on the commits Tab.



SPM587SP21 / SCM587SP21

SPM587SP21 Update j.java 4ae6804 9 days ago 38 commits

README.md	Update README.md	23 days ago
a.java	Create a.java	23 days ago
b.java	Create b.java	23 days ago
c.java	Create c.java	23 days ago
d.java	Fixes #30 TestIssue_030 Adding last name	9 days ago
e.java	Create e.java	23 days ago
f.java	Create f.java	23 days ago
g.java	Create g.java	23 days ago
h.java	Create h.java	23 days ago
i.java	Create i.java	23 days ago
j.java	Update j.java	9 days ago
k.java	Create k.java	23 days ago

About

No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

Packages

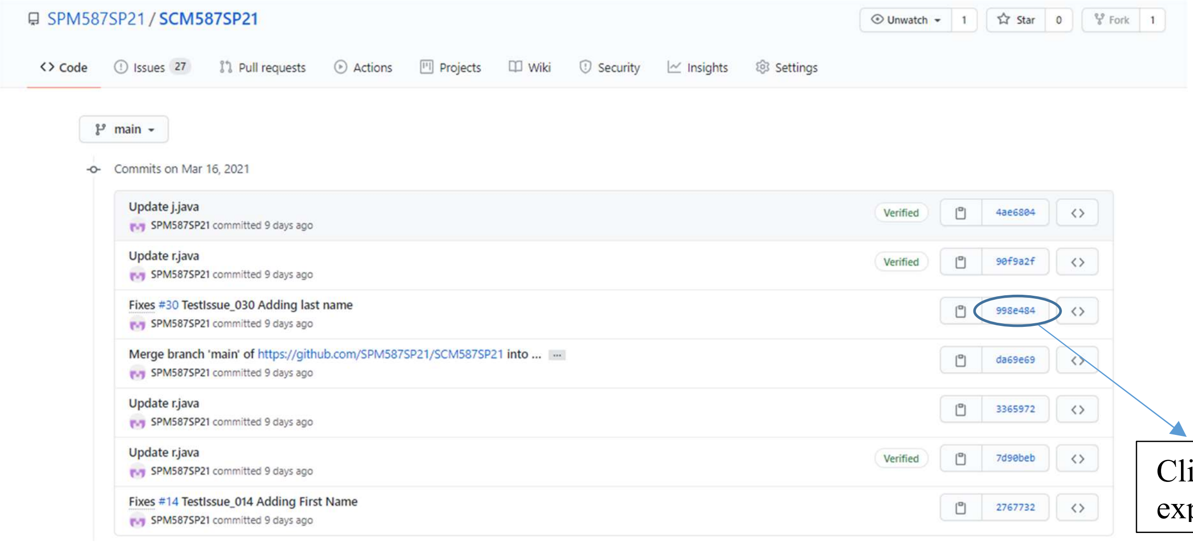
No packages published
Publish your first package

Contributors 2

SPM587SP21
RSP21SCM45R

Click on Commits

2. You will get the following page. You are to take a screenshot of this page. Click on the commit that you have just made to expand it.



Commits on Mar 16, 2021

Update j.java
SPM587SP21 committed 9 days ago — Verified — 4ae6804

Update r.java
SPM587SP21 committed 9 days ago — Verified — 90f9a2f

Fixes #30 TestIssue_030 Adding last name
SPM587SP21 committed 9 days ago — 998e484

Merge branch 'main' of https://github.com/SPM587SP21/SCM587SP21 into ...
SPM587SP21 committed 9 days ago — da69e69

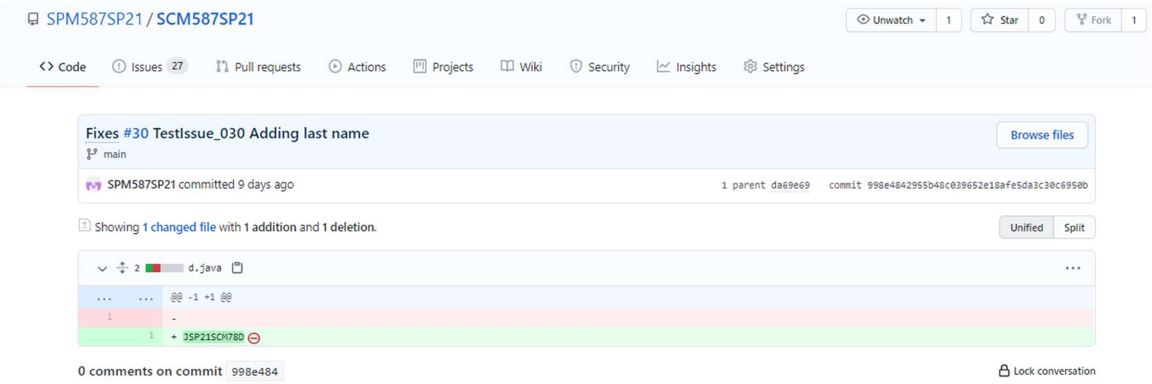
Update r.java
SPM587SP21 committed 9 days ago — 3365972

Update r.java
SPM587SP21 committed 9 days ago — Verified — 7d90beb

Fixes #14 TestIssue_014 Adding First Name
SPM587SP21 committed 9 days ago — 2767732

Click here to expand

3. This will expand and show you the individual commit. You have to take a screenshot of this page as well.



Fixes #30 TestIssue_030 Adding last name
main

SPM587SP21 committed 9 days ago — 1 parent da69e69 — commit 998e4842955b48c039652e18afe5da3c30c6950b

Showing 1 changed file with 1 addition and 1 deletion.

```
2 d.java
@@ -1 +1 @@
1   -
  1 + JSP21SCM78D
```

0 comments on commit 998e484

The report that you have created will be added to the report folder in the repository and submitted. This means that for the phase I report, you will be adding the report document in that Phase1_Report folder and committing it. And for Phase II you will be adding the report document in the folder Phase2_Report folder and committing it. Make sure that the report for each phase has to be committed and pushed to your repository BEFORE you initiate a pull request.

Phase-III Requirements:

Here is the URL to guide you on how to generate your GITHUB_TOKEN https://help.github.com/articles/creating-an-access-token-for-command-line-use/

After you create your GitHub personal access tokens from https://github.com/settings/tokens, update and run the IPYNB script ***Importing_Issues_from_GitHub.ipynb*** to import the issues from the REPO.

Update and run the modified ***Charting_imported_issues.ipynb*** script to plot the following charts:

1. Requirement #1: Plot in Bar Chart the total number of issues closed every day for every Origination Phase
2. Requirement #2: Plot in Bar Chart the total number of issues created for every Phase based on their Status

Deliverable for Phase-III :

Submit your updated ***Charting_imported_issues.ipynb*** script that has the following:

1. Your added code **in every cell** listed in the script
2. Your output **below every cell** listed in the script