CHIANG MAI UNIVERSITY

Bachelor of Science (Software Engineering)

College of Arts, Media and Technology

2nd Semester / Academic Year 2020

SE 234 Basic Development and Operations

Lab III – Issue Tracking

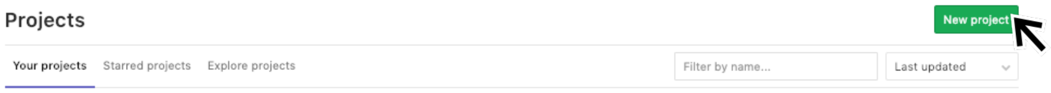
Objective In this session, the students will learn how it is difficult for developers to live without a usable issue tracking system.

Instruction This exercise has the similar rules as the previous one such that students are not allowed to talk face to face. On the other hand, you are allowed to use any kind of other communication channels such as email, sms, phone, instant messenger, etc.

0. Setting - Revisit
   - 0.1. Download today's lab resource
     - A web template – restaurant.zip in the course website under this week's lab.
   - 0.2. Let say the group of 3 students are Alice, Bob, and Charlie – One may pretend to be both Alice and Bob if it is difficult for you to find the 3rd member.
   - 0.3. In this week, Alice will be the project manager, Bob will be the developer, and Charlie will be the customer.

1. GitLab

1.1. **Alice** creates a new (remote) repository named “234-Lab3”.



1.2. **Alice** adds Bob to the list of project collaborators. This time Bob will be a part-time developer, so he will have the role of **Developer**. Alice has to restrict his privilege and allows him to fully access only **a sub branch related to bug fix.** This will be instructed in 1.9.

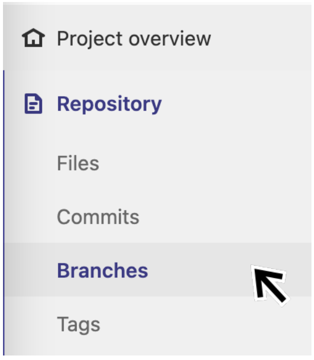
1.3. **Alice** adds Charlie to the list of project collaborators. This time Charlie will be a customer who should not see what are discussed by the devs, so he will have the role of **Guest**. Alice has to restrict his privilege and allows him to fully access only a sub branch related to bug fix. This will be instructed in 1.9.

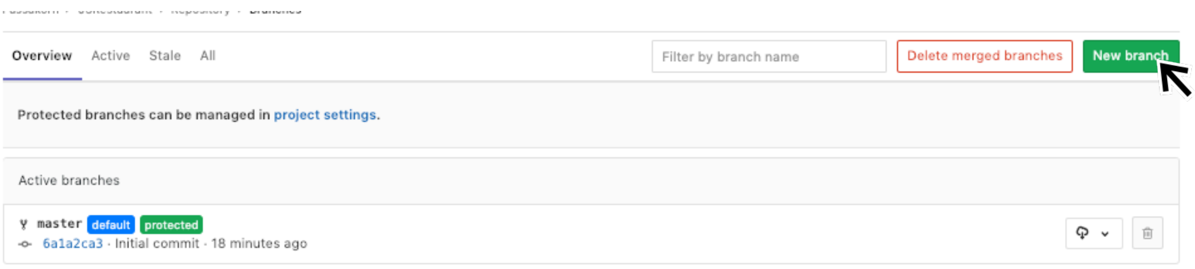
1.4. **Alice** and **Bob** clone the repository using the same method done in the previous lab.

1.5. **Alice** uncompresses the restaurant.zip file, adds them to the 234-Lab3 folder, stages all the files, and makes the first commit

1.6. **Alice** pushes her commit to the Gitlab remote repository.

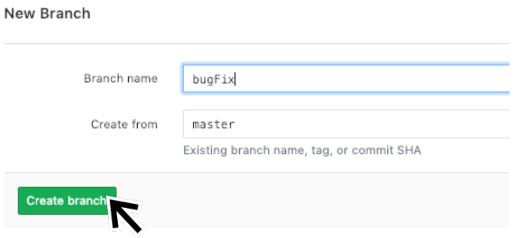
1.7. **Alice** creates a new branch named **bugFix** in the GitLab system.





Protected branches can be managed in project settings.

master
6a1a2ca3 · Initial commit · 18 minutes ago

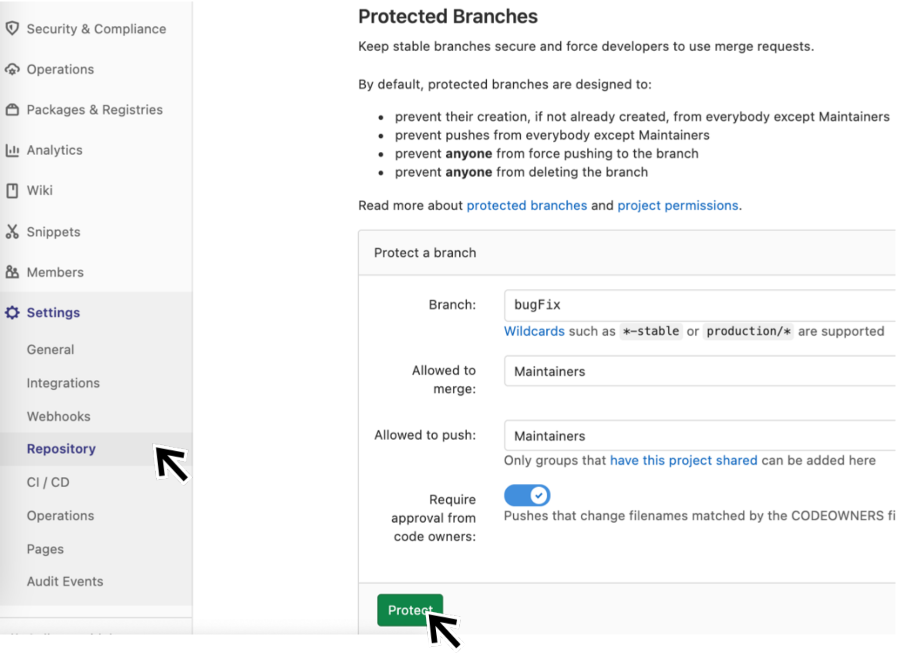


Branch name: bugFix

Create from: master

1.8. Bob pulls the repository as to sync with Alice.

1.9. Alice enables branch restrictions to manage and control the administrative privileges of the repository.



Protected Branches

Keep stable branches secure and force developers to use merge requests.

By default, protected branches are designed to:

- prevent their creation, if not already created, from everybody except Maintainers
- prevent pushes from everybody except Maintainers
- prevent **anyone** from force pushing to the branch
- prevent **anyone** from deleting the branch

Read more about protected branches and project permissions.

Protect a branch

Branch: bugFix
Wildcards such as `*-stable` or `production/*` are supported

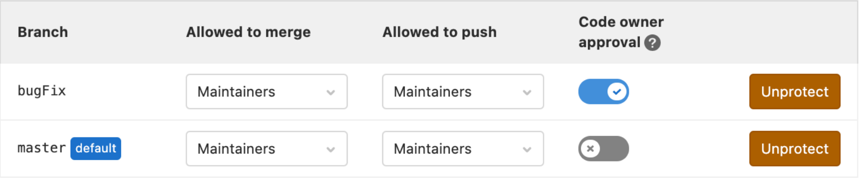
Allowed to merge: Maintainers

Allowed to push: Maintainers
Only groups that have this project shared can be added here

Require approval from code owners: ✓
Pushes that change filenames matched by the CODEOWNERS fi

- The bugFix branch is now protected.

1.10. Show the staff that your experimental setup is completed.



Branch	Allowed to merge	Allowed to push	Code owner approval ?	
bugFix	Maintainers	Maintainers	✓	Unprotect
master default	Maintainers	Maintainers	✕	Unprotect

Staff's Signature ____________________________ (1)

2. The real task begins – Instruction

2.1. Charlie is the customer who paid for this project. Suppose that the earlier stages of the development were not done neatly such that there has been a lot of deviations between the product and the Charlie's expectation. Let the terms bug/defect/issue be used interchangeably, and they all denote anything deviated from Charlie's expectation.

2.2. After the exercise has begun, let Charlie submit his request for changes as issues in the Alice's GitLab page of this repository. These changes are listed under 4.

2.3. Let Alice manage to fix all the Charlie's requests by hiring Bob as her assistant in part time basis. So, this is the reason why Bob should not have the privilege to directly modify the important branches in the remote repository.

2.4. **Rule**: Only two issues can be active at the same time, e.g., Only 4 defects can have the **assigned** state at the same time. When the pipeline is full, one issue must be resolved before the new one is allowed to put to the queue. **You need to find your own way to manage these restrictions by yourself.** For example, let Bob keep informing Alice on what he is doing or how many tasks are he carrying out. This is important because, in the real world, it is totally mental unhealthy for anyone who feels overwhelmed by their to-do list.

- Bob can start doing a task only if it is assigned by Alice.
- Alice can assign herself any task (or she can simply throw all the tasks to Bob).

2.5. Let issues described in 4 states --- New, Assigned, Won't Fix, Closed.

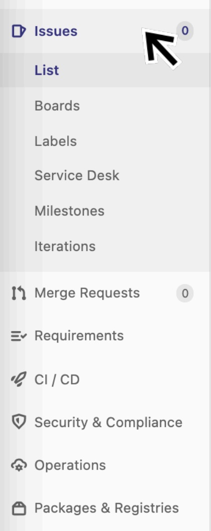
2.6. Let Priority described in 3 states – Low, Medium, and High. Whenever a defect described with high priority arrives, Alice has to pause any task being carried out and start resolving it immediately, e.g., she can either do by herself or force Bob to do it immediately.

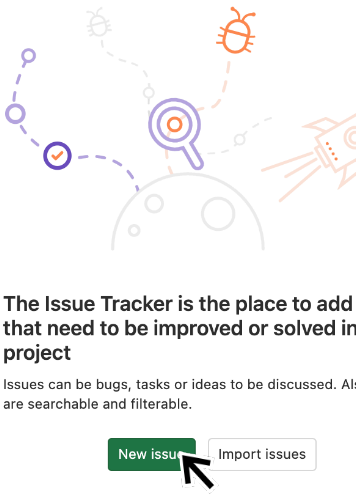
2.7. Every time that Charlie creates an issue, he should provide a possible solution to the issue in the comment section of the issue submission page. And every time he receives a reply stating the completeness, he should acknowledge or write a short thank you note to the developer team.

2.8. **Note**: It will be messy to manage all of these stuffs to go smoothly. In the next week, we will discuss how to use Kanban to better manage the flow of the process. Please do not worry, I will ask you to swap the roles between Alice, Bob, and Charlie on the next week.

3. Example

3.1. Charlie feels that the map is too large and wants it to be smaller by 30%

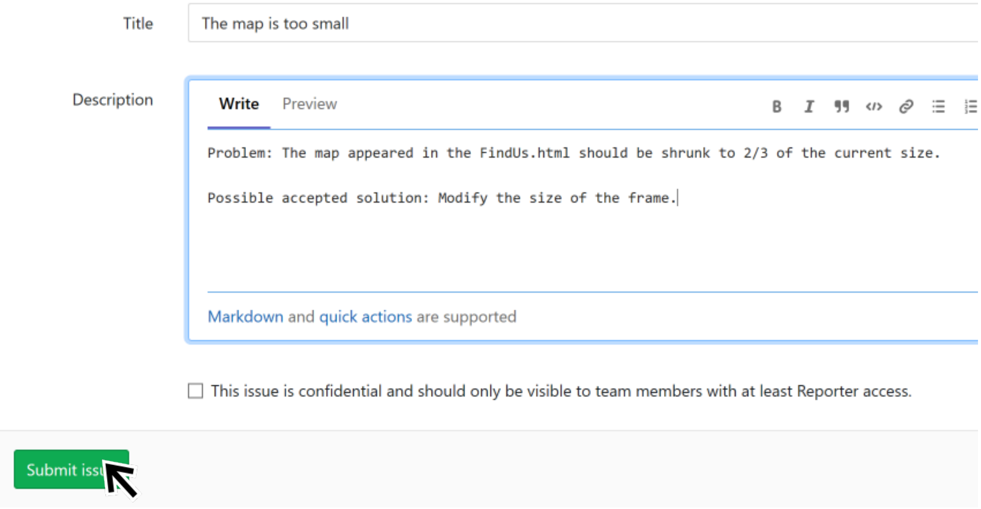




The Issue Tracker is the place to add that need to be improved or solved in project

Issues can be bugs, tasks or ideas to be discussed. Al: are searchable and filterable.

- He asks the developer team for the changes.



Title: The map is too small

Description:

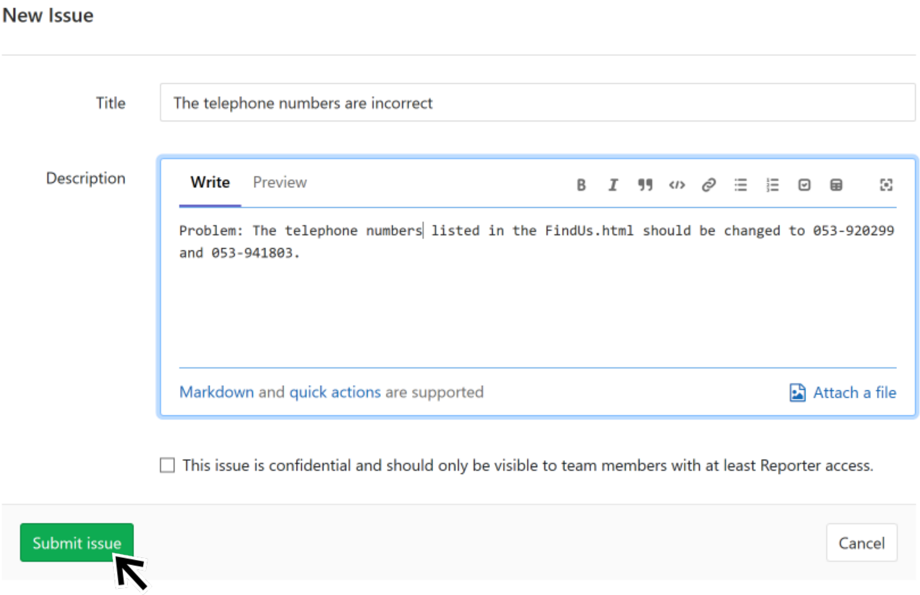
```
Problem: The map appeared in the FindUs.html should be shrunk to 2/3 of the current size.

Possible accepted solution: Modify the size of the frame.
```

Markdown and quick actions are supported

□ This issue is confidential and should only be visible to team members with at least Reporter access.

3.2. As being a customer, Charlie should not be able to see any tags.

3.3. Charlie found another issue and he decided to submit it. Click “New Issue”



New Issue

Title: The telephone numbers are incorrect

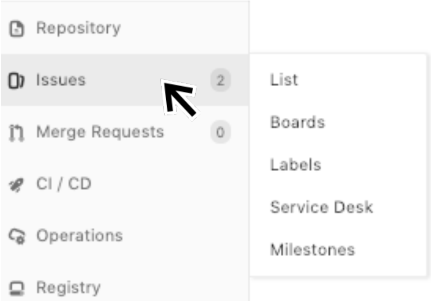
Description:

```
Problem: The telephone numbers listed in the FindUs.html should be changed to 053-920299
and 053-941803.
```

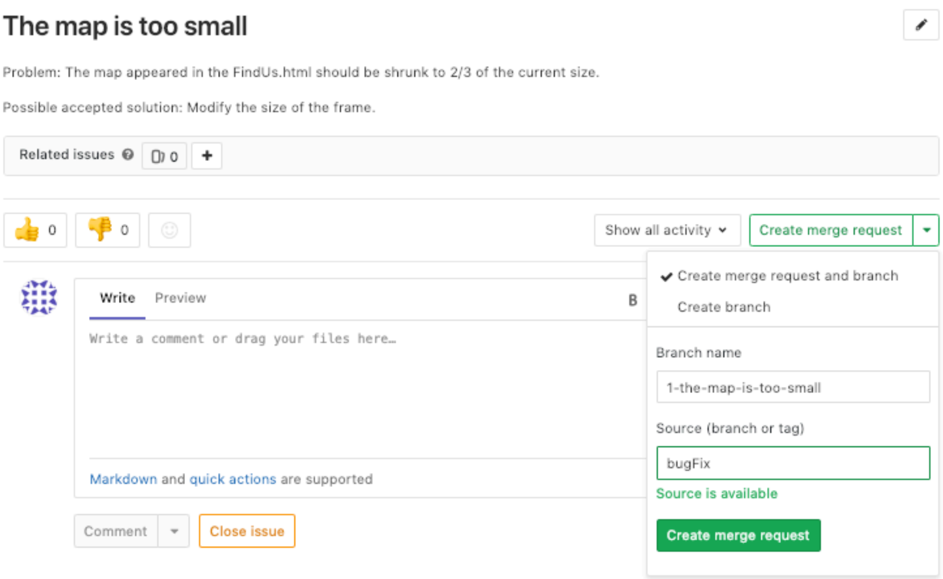
Markdown and quick actions are supported

□ This issue is confidential and should only be visible to team members with at least Reporter access.

3.4. In reality, Alice shall not know when a new issue is arrived, so she may have to periodically or actively check the Gitlab dashboard. Suppose that Alice sees the two incoming issues at the same time. In turn, she divides the workload by assigning the issue #1 to herself and the issue #2 to Bob.



3.5. Alice creates a new merge-request and branch in Gitlab:



The map is too small

Problem: The map appeared in the FindUs.html should be shrunk to 2/3 of the current size.

Possible accepted solution: Modify the size of the frame.

Branch name: 1-the-map-is-too-small

Source (branch or tag): bugFix

Source is available

Write a comment or drag your files here…

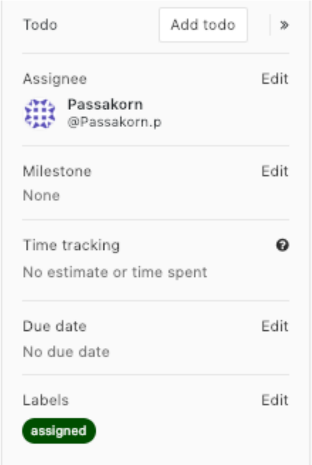
Markdown and quick actions are supported

Note: the source must be bugfix.

The creation of this branch will allow Bob to pull and push on this branch.

3.6. Alice creates required tags named assigned, fixed, won't fix, closed, priority:high, priority:low. We leave the other tags as the default values.

3.7. Alice ticks both assigned for the two issues, #1 and #2.



Assignee: Passakorn @Passakorn.p

Milestone: None

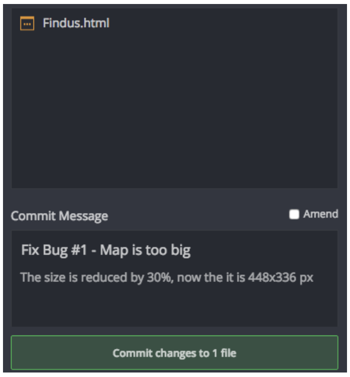
Time tracking: No estimate or time spent

Due date: No due date

Labels: assigned

3.8. From now, after a task is done the assignee must create a merge request and ask Alice to review it.

- Every commit must refer to the corresponding issue id the work was carried out to fix the issue.



Findus.html

Commit Message

Fix Bug #1 - Map is too big

The size is reduced by 30%, now the it is 448x336 px

3.9. When Alice receives a merge request, she has to review it and determine whether the issue has reached the end of its life cycle.

3.10. If no other problem seems to be existed for a particular issue id, Alice can **close** the issue by click the merge button in Gitlab.

3.11. If there is a conflict, check the conflict resolution guideline available on:

- https://about.gitlab.com/2016/09/06/resolving-merge-conflicts-from-the-gitlab-ui/
- https://docs.gitlab.com/ee/user/project/merge_requests/resolve_conflicts.html

3.12. Whenever Alice feels that she should spend more time on the management rather than being a bug fixer herself, she can let Bob do all the bug fixes. But keep in mind that Bob is able to handle only two tasks at the same time.

3.13. After a fix is completed, Alice has to inform Charlie the progress through the Github issue page. If Charlie feels that the fix does not satisfied him, he can request for a reopen again. Otherwise, he should acknowledge or write a short thank you note to the developer team.

Staff's Signature ____________________________ (2)

4. Fix the defects listed below by following the workflow instructed in 1, 2, and 3. The staff will check the results, the Git Tree, and more importantly, the work products showed in the Gitlab issue page. **Note** that the staff will sign this sheet after the issues corresponding to 4.5, 4.10, and 4.15 are closed, respectively.

4.1. Telephone number in other pages are not yet updated.
4.2. Find 3 new photos to replace the slideshow in the front page
4.3. At the main course section, we no longer serve Vegetarian and Poultry dishes.
4.4. Prices shown in the entire sites are not in a correct currency, replace £ with $. (Priority = High)
4.5. The **2FOR12012** code does not have a proper description. Please tell the customer that it is a coupon code and can be redeemed only on site. Also change the year to **2020**.

After 4.5 is completed and merged, let forward the bugfix branch to the master as to be an official updated version. Name the version tag as appropriate.

Staff's Signature ____________________________ (3)

4.6. Delivery service is delegated to Food panda. So, it is no longer a special offer. Remove it from the special offer page and provide the information about Food panda instead.
4.7. Online menu is no longer available, remove any of the corresponding records from the website.
4.8. In the front page and special offers page, the 50% discount is incorrect. It should be 20%. (Priority = High)
4.9. The booking can be now made up to 48 hours in advance.
4.10. Change the address to 239 Huaykaew Rd., Suthep, Muang, Chiang Mai 50200

After 4.10 is completed and merged, let forward the bugfix branch to the master as to be an official updated version. Name the version tag as appropriate.

Staff's Signature ____________________________ (4)

4.11. All the texts are not suited to Charlie's taste, change them all to Arial.

4.12. A psychologist told Charlie that the main color scheme of a restaurant should be in red, so he insists the team to alter the color for the entire site.

4.13. The 12.5% under the booking info is reduced to 7%.

4.14. "Cancellations if possible should be made 24 hours in advance" needed to be changed to "Cancellations must be made at least 24 hours in advance"

4.15. Remove all the email from the website.

After 4.15 is completed and merged, let forward the bugfix branch to the master as to be an official updated version. Name the version tag as appropriate.

Staff's Signature ___________________________ (5)

===== This is the end of today's exercise =====