CCS3313-Advanced Software Design

Report on

Version Controlling a Class Diagram Using Git

Group 07

22UG1-0285 - K.G.G.R Bandara

22UG1-0306 - L.B.I.R.Lakshan

22UG1-0238 - R.K.N.R.Ranasinghe

22UG1-0258 - W.M.N.A.K Weerasinghe

22UG2-0568 – D.N. Wickramaarachchi

22ug2-0118 - K.S.N.Dilthusha

BSc Hons in Software Engineering

SLTC Research University

08/12/2024

Introduction

Version control systems are essential in modern software development for managing changes and enabling collaboration on projects. This lab focused on utilizing Git, a distributed version control system, to manage and track changes to a class diagram, which serves as a crucial blueprint for representing the relationships and interactions between system entities. The lab emphasized practical tasks such as setting up a repository, collaborating on the diagram, resolving merge conflicts, and producing a final deliverable. Through these activities, the lab aimed to develop skills in applying Git commands and workflows, managing version-controlled files, and ensuring seamless collaboration in a team environment.

Tasks Completed

1 .Repository Setup

- Created a local folder for the project named `InventoryManagementSystem`.
- Initialized a Git repository in the folder

```
C:\Users\Gayanga\Desktop\InventoryManagementSystem>git init
Initialized empty Git repository in C:/Users/Gayanga/Desktop/InventoryManagementSystem/.git/
```

- Created a subfolder named `Diagrams` and saved the class diagram inside.
- Added a `README.md` file to describe the system's purpose and key entities.

```
C:\Users\Gayanga\Desktop\InventoryManagementSystem>echo "InventoryManagementSystem"> README.md
```

- Added files to the Git repository and committed with an initial message.

2. Remote Repository on GitHub

- Logged into GitHub and created a remote repository.
- Linked the local repository to the remote using `git remote add origin`.
- Pushed the initial code to the remote repository.

```
C:\Users\Gayanga\Desktop\InventoryManagementSystem>git remote add origin https://github.com/GayangaBandara/InventoryManagementSystem.git

C:\Users\Gayanga\Desktop\InventoryManagementSystem>git branch -M main

C:\Users\Gayanga\Desktop\InventoryManagementSystem>git push -u origin main
```

3. Collaborative Work with Git

- Cloned the repository locally.

```
C:\Users\user\OneDrive\Desktop\GIT>git clone https://github.com/GayangaBandara/InventoryManagementSystem.git
Cloning into 'InventoryManagementSystem'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 21 (delta 3), reused 6 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (21/21), 83.36 KiB | 49.00 KiB/s, done.
Resolving deltas: 100% (3/3), done.
```

- Updated the `README.md` file with contributors' details.

```
C:\Users\user\OneDrive\Desktop\GIT\InventoryManagementSystem>git checkout -b update-README
Switched to a new branch 'update-README'
```

- Modified the class diagram by adding relationships and renaming a class.

```
C:\Users\user\OneDrive\Desktop\GIT\InventoryManagementSystem>git commit -m "Updated README,Supplier"
[update-README 9bdde6f] Updated README,Supplier
 1 file changed, 12 insertions(+)
```

- Created a new branch named `update-class-diagram` and made the changes.

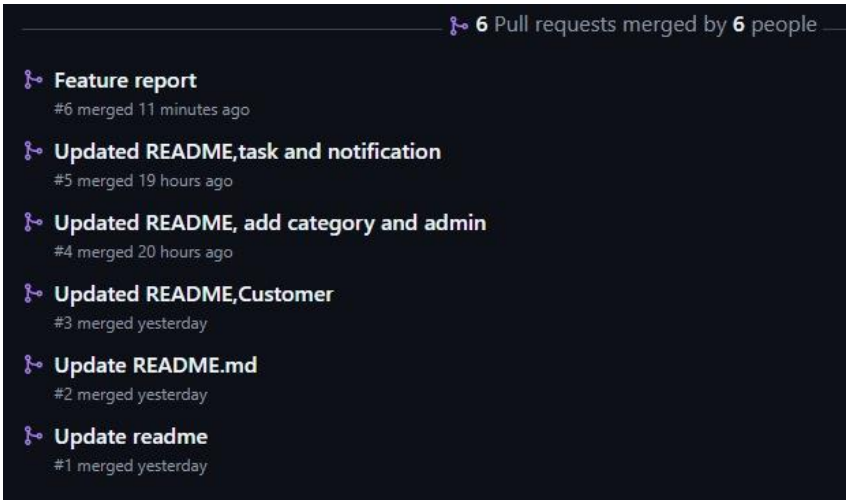
```
4. Supplier
    - What it Represents: Companies or individuals providing products.
    - Attributes:
        - Supplier ID
        - Name
        - Contact Information
    - Functions:
        - Maintain supplier records for procurement.
```

- Committed and pushed the changes to the branch.

```
C:\Users\user\InventoryManagementSystem>git push origin update-README
info: please complete authentication in your browser...
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/GayangaBandara/InventoryManagementSystem.git
   023b896..2206f31  update-README -> update-README
```

4. Review and Merge

- Opened a pull request to merge the branch `update-class-diagram` into `main`.
- Assigned a group member for code review.
- Merged the pull request after review and pulled the changes to the local repository.



5. Conflict Resolution

- Intentionally edited the same part of the `README.md` file on two separate branches.
- Merged the branches into the `main` branch and resolved conflicts locally.

Challenges Faced and Solutions

1. Permission Denied Error While Pushing

- Challenge: Encountered a `403 Permission Denied` error when trying to push changes to the remote repository.

```
C:\Users\user\OneDrive\Desktop\GIT\InventoryManagementSystem>git push origin update-README
info: please complete authentication in your browser...
remote: Permission to GayangaBandara/InventoryManagementSystem.git denied to Dewninim.
fatal: unable to access 'https://github.com/GayangaBandara/InventoryManagementSystem.git/': The requested URL returned error: 403
```

- Cause: The collaboration request from a group member had not been accepted yet.

@GayangaBandara has invited you to collaborate on the **GayangaBandara/InventoryManagementSystem** repository

You can accept or decline this invitation. You can also head over to https://github.com/GayangaBandara/InventoryManagementSystem to check out the repository or visit @GayangaBandara to learn a bit more about them.

This invitation will expire in 7 days.

View invitation

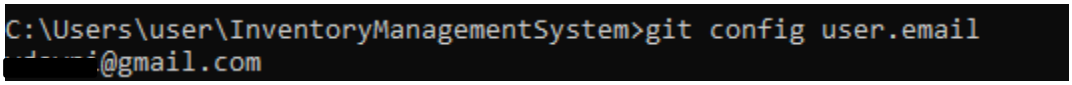
- Solution: Accepted the collaboration request from the repository owner on GitHub and retried the push successfully.

2. Conflict While Merging

- Challenge: Conflicts arose while merging branches where the same file was edited.
- Cause: Multiple contributors worked on the same section of the `README.md` file.
- Solution: Resolved the conflicts manually by editing the file, completing the merge, and committing the changes.

3. Initial Git Setup Issues

- Challenge: Group members faced issues setting up Git configurations (e.g., email and username).
- Solution: Guided members on setting up Git configurations using `git config` commands and verified their setup.



Key Learnings

- Successfully practiced using Git for version control in collaborative environments.
- Learned to manage branches, resolve merge conflicts, and review code through pull requests.
- Understood the importance of clear communication and permissions for smooth collaboration.

Conclusion

Version controlling the class diagram using Git provided valuable hands-on experience in managing collaborative projects. We successfully set up a Git repository, collaborated with team members, and resolved challenges such as permissions errors and merge conflicts. This project underscored the significance of using version control tools to ensure efficient and error-free development processes. The skills gained will be essential for future software development endeavors.