CCS3313-Advanced Software Design

Report

Version Controlling a Class Diagram Using Git

E-Commerce Platform

Group No: 10

22UG1-0828 R.M.D.A.RATHNAYAKA

22UG1-0010 T.H.R.C.NIROSHA

22UG1-0638 R.T.M BUDDHIKA

22UG1-0392 - K G V T GAMAGE

22UG1-0013 T.H.R.C.NADEESHA

22UG2-0571 F.M.HILHAM

Introduction

Version control systems play a crucial role in modern software development by tracking changes and facilitating collaboration among developers working on the same project. This assignment focused on using Git, a distributed version control system, to manage and monitor modifications to a class diagram, which is a key representation of the system's entities and their relationships. The assignment involved practical exercises such as initializing a repository, collaborating on diagram updates, resolving conflicts, and merging changes into a unified project. Through these tasks, the assignment aimed to enhance our understanding of Git commands, workflows, and conflict resolution, ensuring smooth collaboration and version control in a team-oriented development environment.

Tasks Completed

1. Repository Setup

- Created a local folder for the project named `E-commerce platform` and initialized a Git repository in the folder

- Then we 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.

- 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.

3. Collaborative Work with Git

- Cloned the repository locally.

```
C:\Users\h p>git clone https://github.com/DhanukaRathnayaka/E-CommercePlatform.git
Cloning into 'E-CommercePlatform'...
remote: Enumerating objects: 52, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (41/41), done.
Rremote: Total 52 (delta 12), reused 21 (delta 7), pack-reused 0 (from 0)
Receiving objects: 100% (52/52), 117.51 KiB | 539.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.

C:\Users\h p>
```

- Updated the `README.md` file

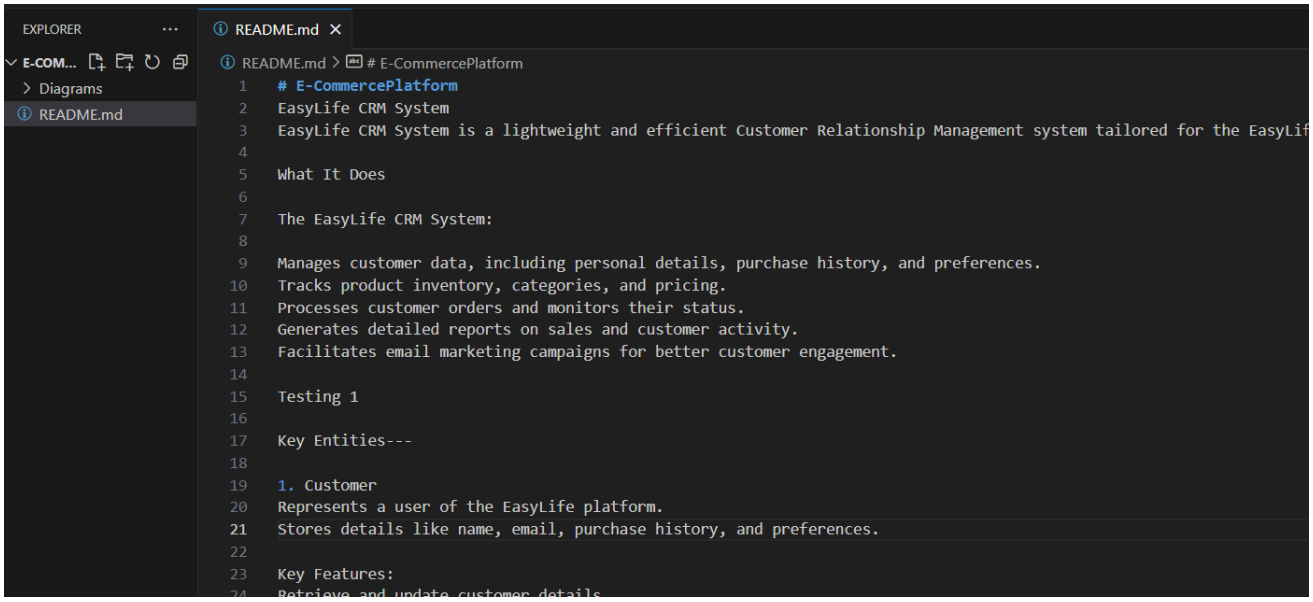
```
C:\Users\h p>git checkout -b update-README
Switched to a new branch 'update-README'

C:\Users\h p>
```

- Modified the customer diagram.

```
C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>git commit -m "Updated README,Customer"
[update-README 1684e3f] Updated README,Customer
 1 file changed, 1 insertion(+), 1 deletion(-)
```

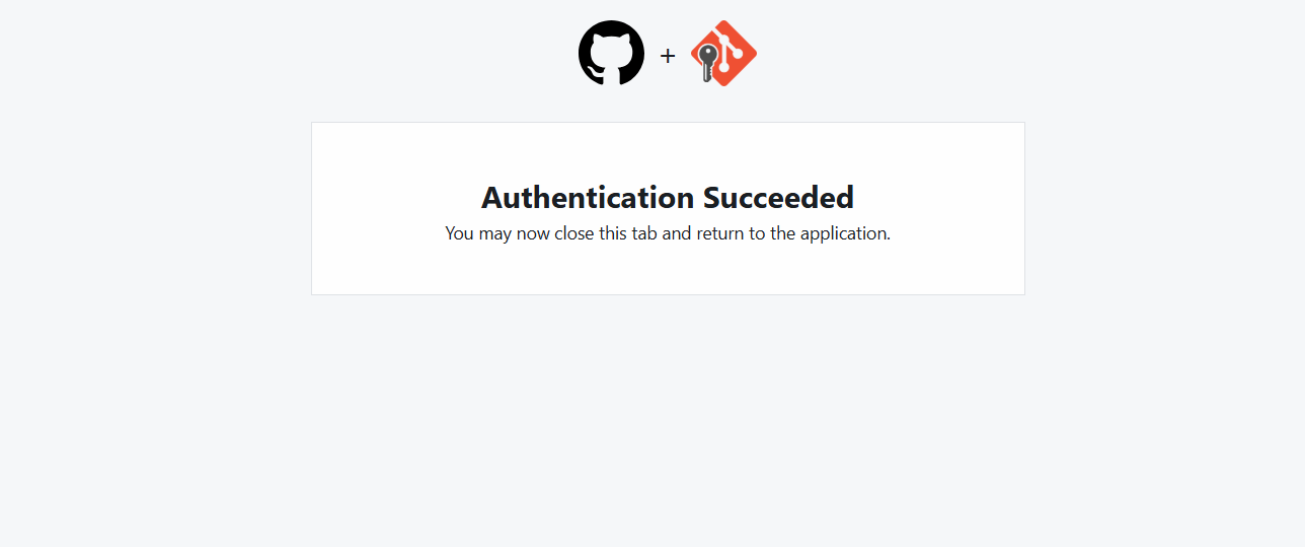
- Created a new branch named `update README.md` and made the changes.



- Committed and pushed the changes to the branch.

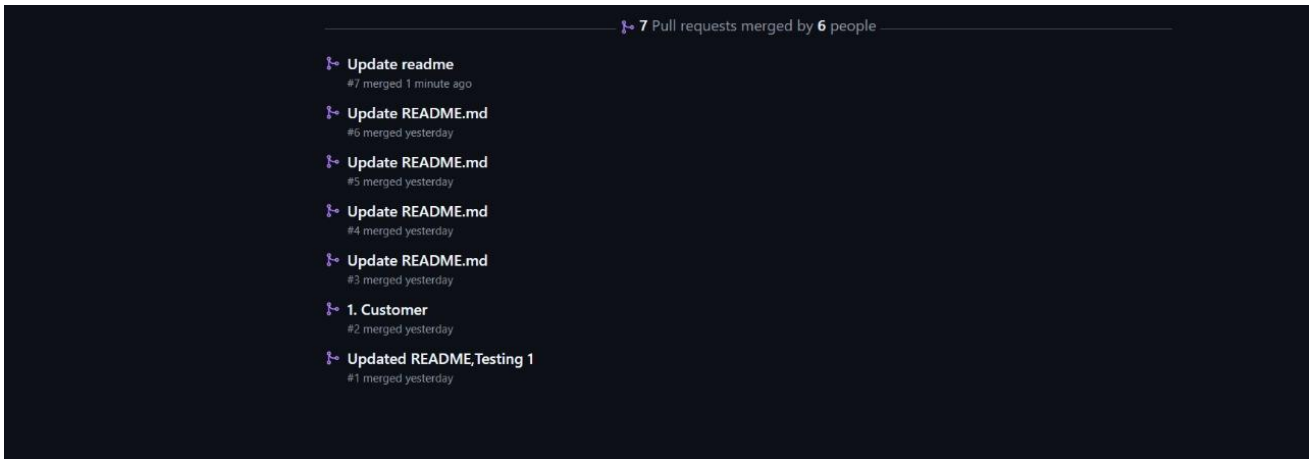
```
C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>git push origin update-README
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 760 bytes | 760.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/DhanukaRathnayaka/E-CommercePlatform.git
   1195239..7b013aa  update-README -> update-README

C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>
```



4. Review and Merge

Opened a pull request to merge the branch `README.md` 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

During the process of committing and pushing updates to the update-README branch in our project, we encountered a **non-fast-forward push error**. This challenge arose when attempting to push changes from the local repository to the remote branch, only to find that the remote branch was ahead of the local branch.

```
C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>git push origin update-README
To https://github.com/DhanukaRathnayaka/E-CommercePlatform.git
 ! [rejected]        update-README -> update-README (non-fast-forward)
error: failed to push some refs to 'https://github.com/DhanukaRathnayaka/E-CommercePlatform.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>git pull origin update-README
error: Pulling is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
```

How we resolved it,

```
C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>git commit -m "Updated README,Customer"
[update-README 7b013aa] Updated README,Customer

C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>git pull origin update-README
From https://github.com/DhanukaRathnayaka/E-CommercePlatform
 * branch            update-README -> FETCH_HEAD
Already up to date.

C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>git push origin update-README
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 760 bytes | 760.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/DhanukaRathnayaka/E-CommercePlatform.git
   1195239..7b013aa  update-README -> update-README

C:\Users\h p\OneDrive\Desktop\Advanced software\E-CommercePlatform>
```

Key Learnings of the Assignment

- Understanding the importance of structuring a project with a clear folder hierarchy, including subfolders like Diagrams for easier navigation and maintenance.
- Practical experience in initializing a Git repository, tracking files with git add, and committing changes with meaningful messages to create a reliable project history.
- Gained hands-on experience linking a local repository to GitHub, enabling seamless collaboration.
- Understanding how to clone remote repositories for collaboration, ensuring that all team members start from the same codebase.
- Practical experience with creating pull requests to merge feature branches into the main branch.
- Working in a team environment with Git fosters collaboration and ensures project changes are well-tracked.

This assignment reinforced practical skills in version control, improved understanding of collaborative workflows, and highlighted the importance of communication and organization in software development. These skills are foundational for professional software development and teamwork in real-world projects.

Conclusion

This Git assignment was a comprehensive exercise in understanding and applying version control principles. Through tasks like repository setup, branching, collaborative editing, pull requests, and conflict resolution, we gained hands-on experience in managing a shared codebase.

The practical challenges, such as resolving merge conflicts and synchronizing local and remote repositories, emphasized the importance of structured workflows and clear communication in collaborative software development. Additionally, the assignment reinforced best practices such as writing meaningful commit messages, using branches to isolate tasks, and conducting thorough code reviews to maintain quality.

Overall, the assignment provided a solid foundation in using Git and GitHub effectively, equipping us with essential skills to manage real-world software projects efficiently and collaboratively.