# Report on Version Controlling the Class Diagram Using Git

Course: CCS3313 - Advanced Software Design

Date: 07/12/2024

Prepared by: **Group No: 02**

**Tharuka Karunarathne | 22UG2-0578**

**W. M. J. V. Weerasekara | 22UG1-0493**

**Geeth Dasanayaka | 22UG2-0121**

**Devanga Wettasinghe | 22UG2-0241**

**R. M. P. Gunarathne | 22UG1-0494**

**Nimesh Rashmika | 22UG2-0052**

**W. A. L. N. Madusanka | 22UG2-0593**

## 1. Introduction

This report documents the process of version-controlling a class diagram using Git, as part of the CCS3313 - Advanced Software Design course. The primary objective of the assignment was to simulate collaboration on a software project by using Git and GitHub. This process involved creating a local Git repository, adding a class diagram, pushing changes to a remote GitHub repository, and managing collaboration using branching and pull requests.

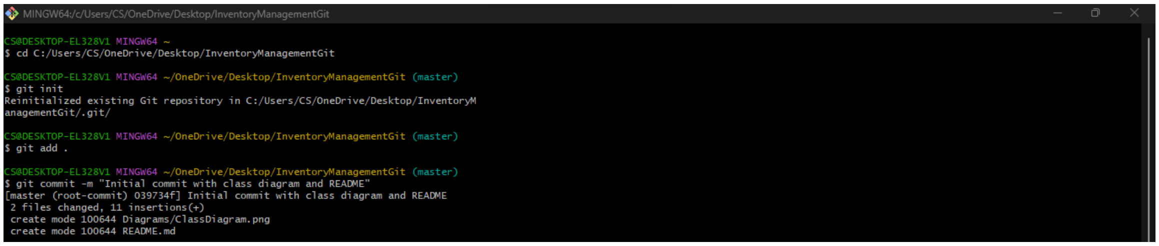
The project focused on an Inventory Management System, and as part of the task, a class diagram representing the system's entities and relationships was version-controlled. A Payment entity was added to the diagram, demonstrating the use of Git for version control and collaboration.

## 2. Tasks Completed

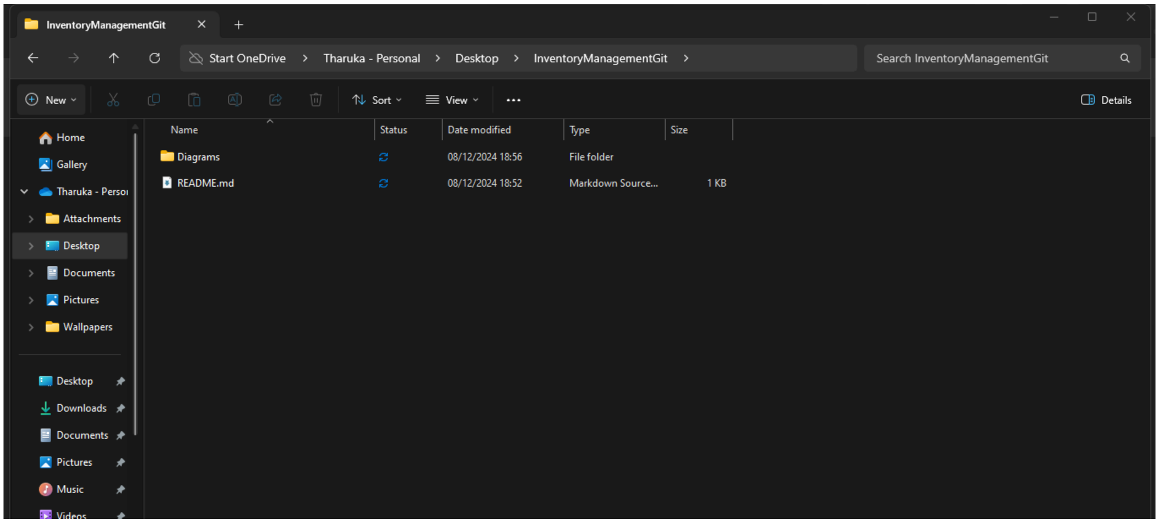
### 2.1. Repository Setup

- Created a local Git repository and organized the project files.

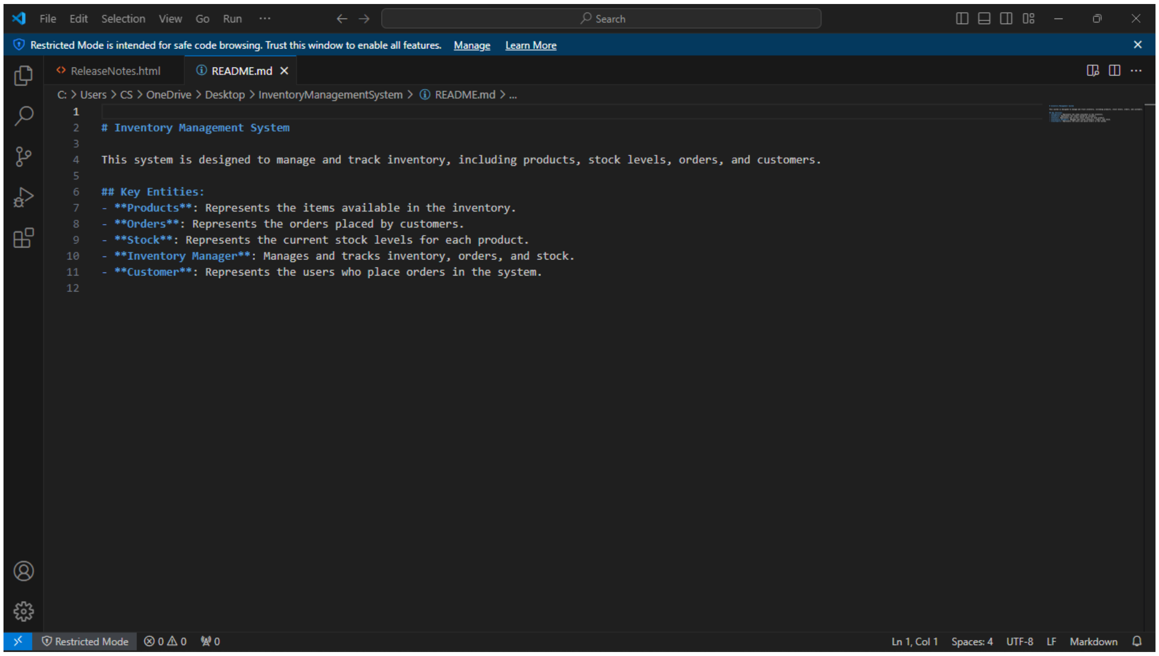
- Added a README.md file to describe the Inventory Management System.
- Created a subfolder called Diagrams and added the class diagram (ClassDiagram.png).
- Committed the initial files to the repository.



```
CS@DESKTOP-EL328V1 MINGW64 ~
$ cd C:/Users/CS/OneDrive/Desktop/InventoryManagementGit

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ git init
Reinitialized existing Git repository in C:/Users/CS/OneDrive/Desktop/InventoryM
anagementGit/.git/

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ git add .

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ git commit -m "Initial commit with class diagram and README"
[master (root-commit) 039734f] Initial commit with class diagram and README
 2 files changed, 11 insertions(+)
 create mode 100644 Diagrams/ClassDiagram.png
 create mode 100644 README.md
```



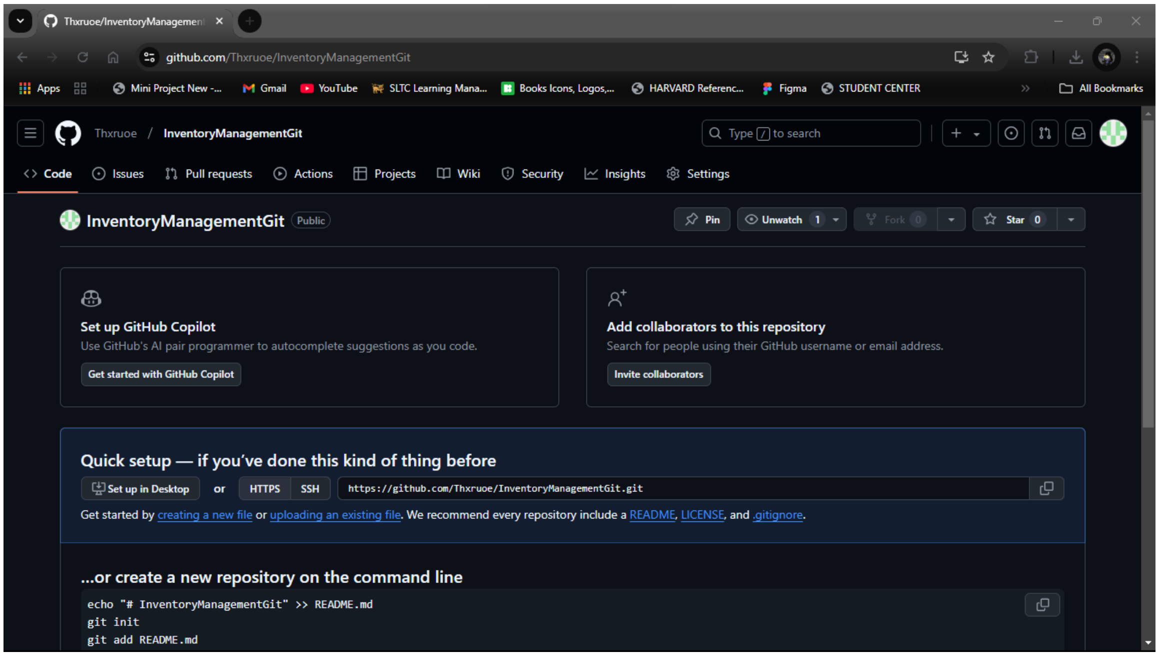
| Name | Status | Date modified | Type | Size |
|---|---|---|---|---|
| Diagrams | | 08/12/2024 18:56 | File folder | |
| README.md | | 08/12/2024 18:52 | Markdown Source... | 1 KB |


File
Edit
Selection
View
Go
Run
Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features.
Manage
Learn More
ReleaseNotes.html
README.md
C: > Users > CS > OneDrive > Desktop > InventoryManagementSystem > README.md > ...
Restricted Mode
0
0
0
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Markdown

```

# Inventory Management System

This system is designed to manage and track inventory, including products, stock levels, orders, and customers.

## Key Entities:
- **Products**: Represents the items available in the inventory.
- **Orders**: Represents the orders placed by customers.
- **Stock**: Represents the current stock levels for each product.
- **Inventory Manager**: Manages and tracks inventory, orders, and stock.
- **Customer**: Represents the users who place orders in the system.

```

## 2.2. Remote Repository

- Created a remote repository on GitHub and linked it to the local repository.
- Pushed the initial files to GitHub to make the repository accessible for collaboration.



**Set up GitHub Copilot**

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot

**Add collaborators to this repository**

Search for people using their GitHub username or email address.

Invite collaborators

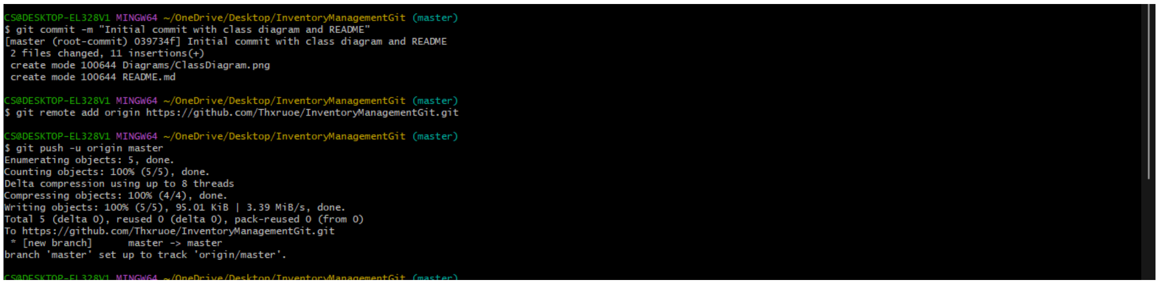
**Quick setup — if you've done this kind of thing before**

Set up in Desktop or HTTPS SSH `https://github.com/Thxruoe/InventoryManagementGit.git`

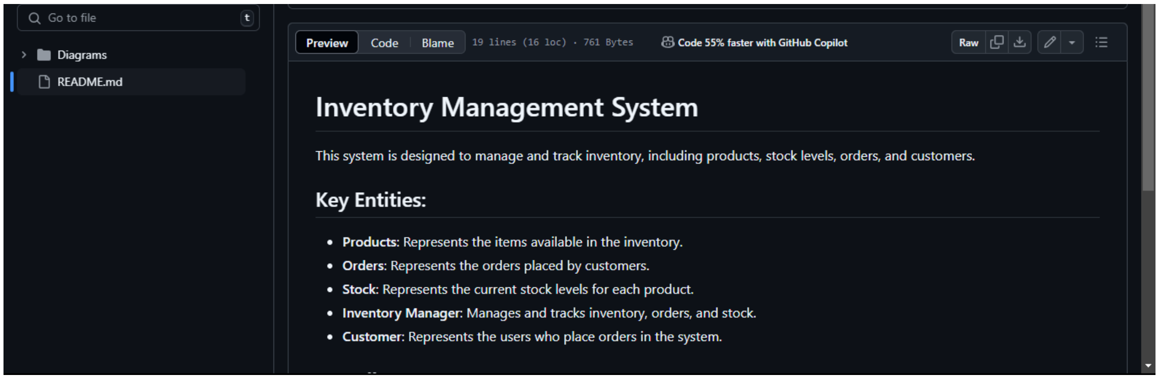
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

**...or create a new repository on the command line**

```
echo "# InventoryManagementGit" >> README.md
git init
git add README.md
```



```
CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ git commit -m "Initial commit with class diagram and README"
[master (root-commit) 039734f] Initial commit with class diagram and README
 2 files changed, 11 insertions(+)
 create mode 100644 Diagrams/ClassDiagram.png
 create mode 100644 README.md

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ git remote add origin https://github.com/Thxruoe/InventoryManagementGit.git

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 95.01 KiB | 3.39 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Thxruoe/InventoryManagementGit.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
```


Go to file
t
Diagrams
README.md
Preview
Code
Blame
19 lines (16 loc) · 761 Bytes
Code 55% faster with GitHub Copilot
Raw

**Inventory Management System**

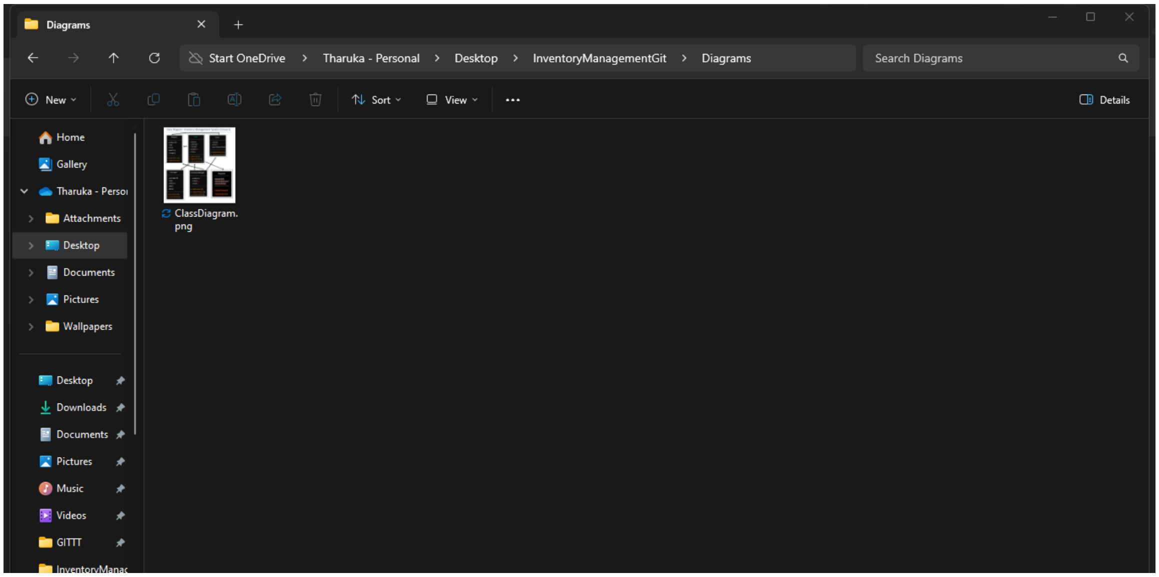
This system is designed to manage and track inventory, including products, stock levels, orders, and customers.

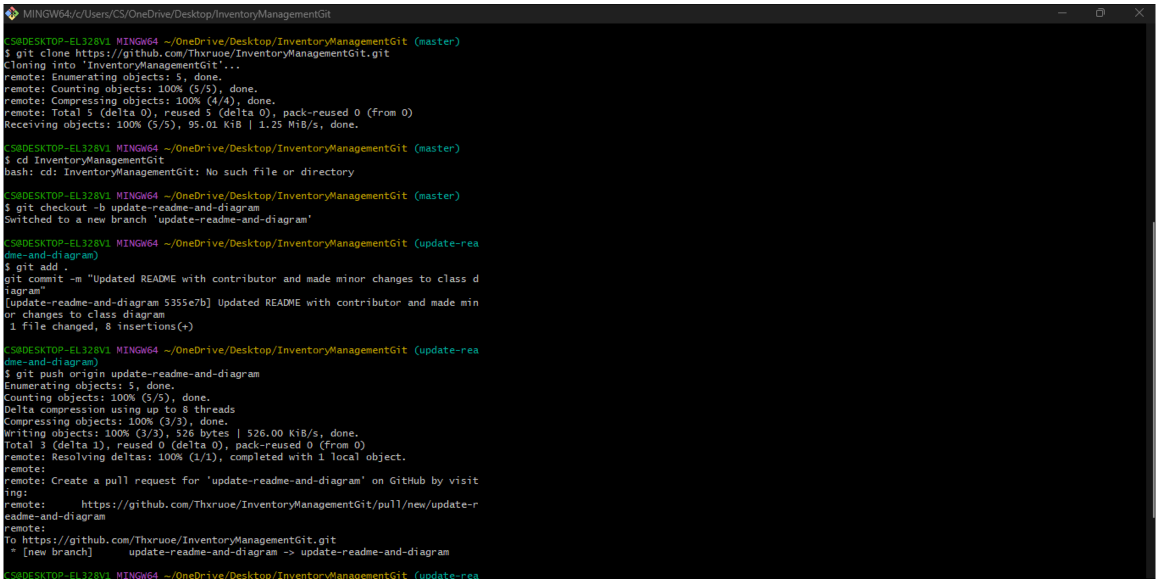
**Key Entities:**

- **Products**: Represents the items available in the inventory.
- **Orders**: Represents the orders placed by customers.
- **Stock**: Represents the current stock levels for each product.
- **Inventory Manager**: Manages and tracks inventory, orders, and stock.
- **Customer**: Represents the users who place orders in the system.

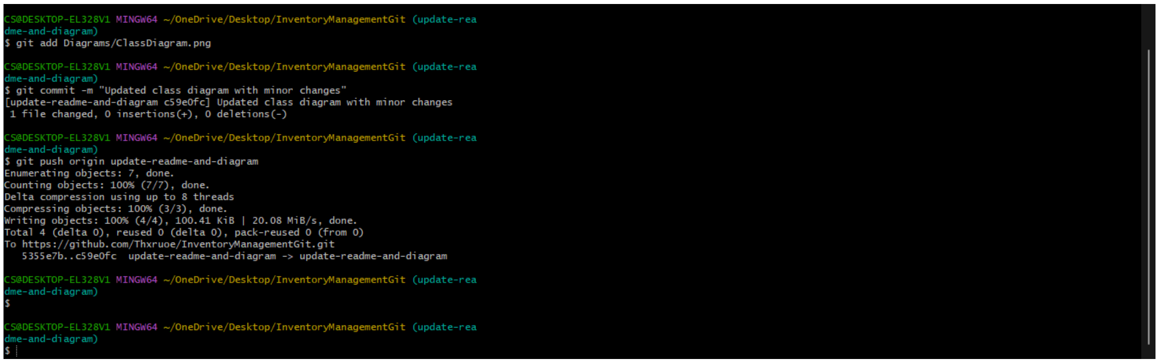
## 2.3. Collaboration:

- Simulated collaboration by creating a new branch (update-class-diagram).
- Added the Payment entity to the class diagram and established a relationship between Order and Payment.
- Committed and pushed the changes to the remote repository.





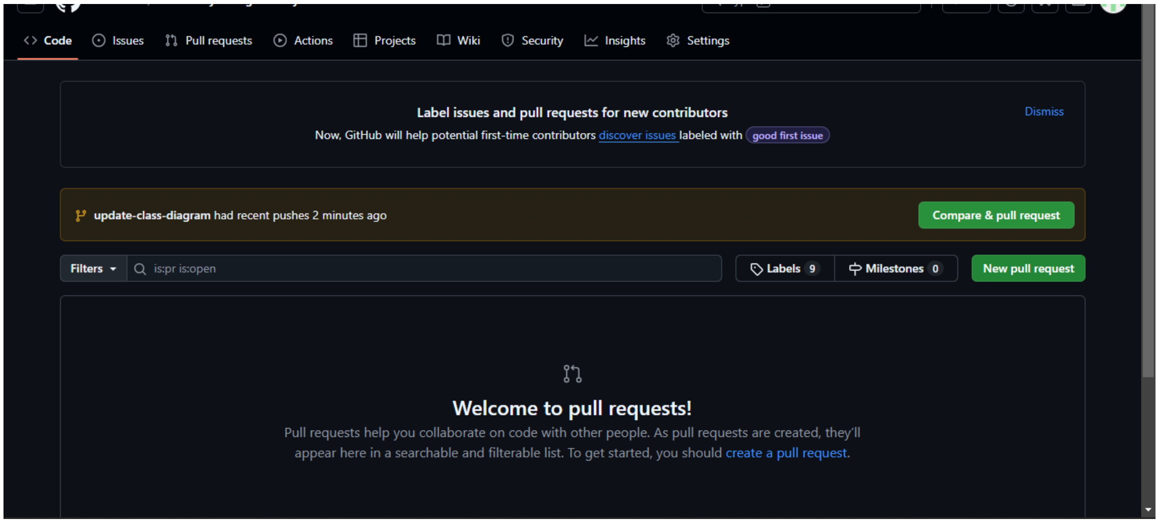
```
CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ git clone https://github.com/Thxruoe/InventoryManagementGit.git
Cloning into 'InventoryManagementGit'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), 95.01 KiB | 1.25 MiB/s, done.

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ cd InventoryManagementGit
bash: cd: InventoryManagementGit: No such file or directory

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (master)
$ git checkout -b update-readme-and-diagram
Switched to a new branch 'update-readme-and-diagram'

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$ git add .
git commit -m "Updated README with contributor and made minor changes to class d
iagram"
[update-readme-and-diagram 5355e7b] Updated README with contributor and made min
or changes to class diagram
 1 file changed, 8 insertions(+)

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$ git push origin update-readme-and-diagram
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 526 bytes | 526.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'update-readme-and-diagram' on GitHub by visit
ing:
remote:      https://github.com/Thxruoe/InventoryManagementGit/pull/new/update-r
eadme-and-diagram
remote:
To https://github.com/Thxruoe/InventoryManagementGit.git
 * [new branch]      update-readme-and-diagram -> update-readme-and-diagram

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
```



```
CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$ git add Diagrams/ClassDiagram.png

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$ git commit -m "Updated class diagram with minor changes"
[update-readme-and-diagram c59e0fc] Updated class diagram with minor changes
 1 file changed, 0 insertions(+), 0 deletions(-)

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$ git push origin update-readme-and-diagram
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 100.41 KiB | 20.08 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Thxruoe/InventoryManagementGit.git
   5355e7b..c59e0fc  update-readme-and-diagram -> update-readme-and-diagram

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$
```

## 2.4. Pull Request & Merge

- Created a Pull Request (PR) for merging the update-class-diagram branch into the master branch.
- Reviewed and merged the PR after confirmation.



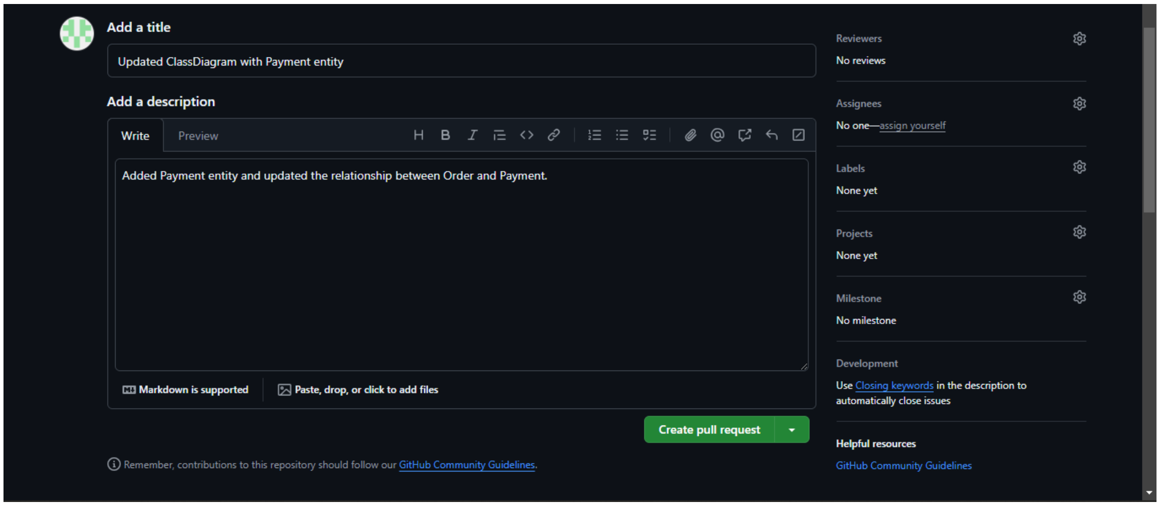
Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

update-class-diagram had recent pushes 2 minutes ago

Welcome to pull requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should create a pull request.



Add a title

Updated ClassDiagram with Payment entity

Add a description

Added Payment entity and updated the relationship between Order and Payment.

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Reviewers
No reviews

Assignees
No one—assign yourself

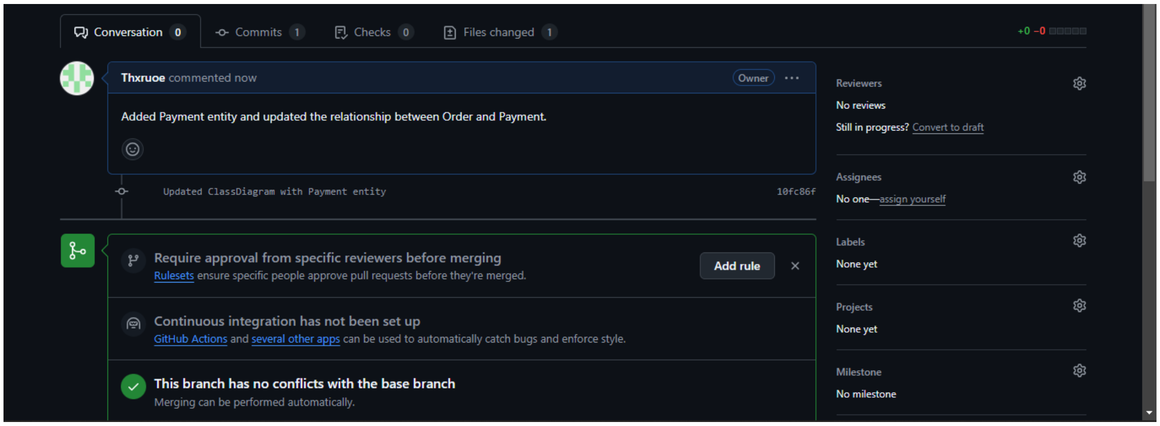
Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Use Closing keywords in the description to automatically close issues

Helpful resources
GitHub Community Guidelines



**Thxruoe** commented now · Owner

Added Payment entity and updated the relationship between Order and Payment.

`Updated ClassDiagram with Payment entity` 10fc86f

**Require approval from specific reviewers before merging**
Rulesets ensure specific people approve pull requests before they're merged.

**Continuous integration has not been set up**
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

**This branch has no conflicts with the base branch**
Merging can be performed automatically.

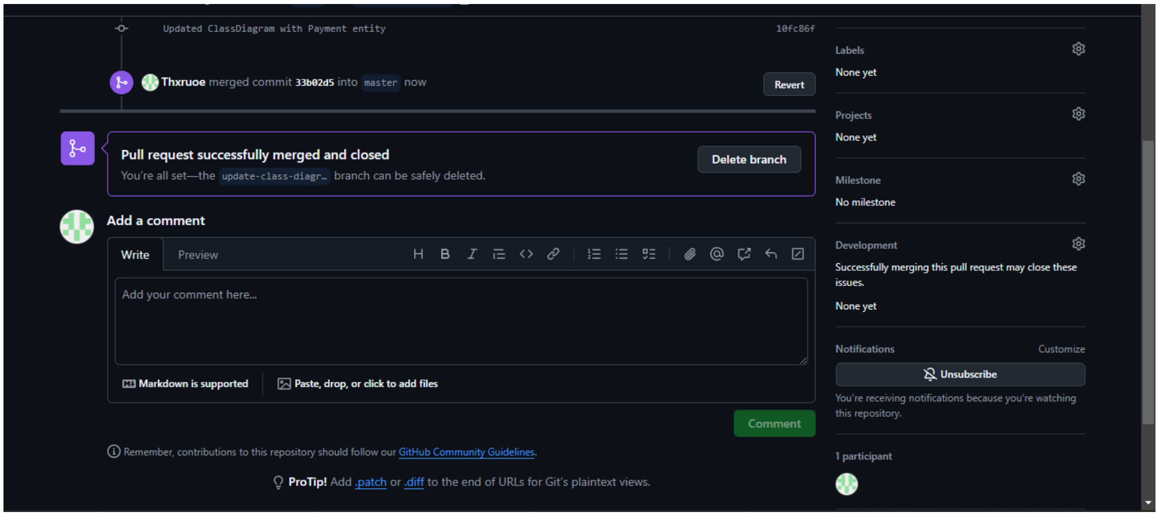
Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone



`Updated ClassDiagram with Payment entity` 10fc86f

**Thxruoe** merged commit **33b02d5** into `master` now

**Pull request successfully merged and closed**
You're all set—the `update-class-diagr…` branch can be safely deleted.

**Add a comment**

Remember, contributions to this repository should follow our GitHub Community Guidelines.

**ProTip!** Add .patch or .diff to the end of URLs for Git's plaintext views.

Labels
None yet

Projects
None yet

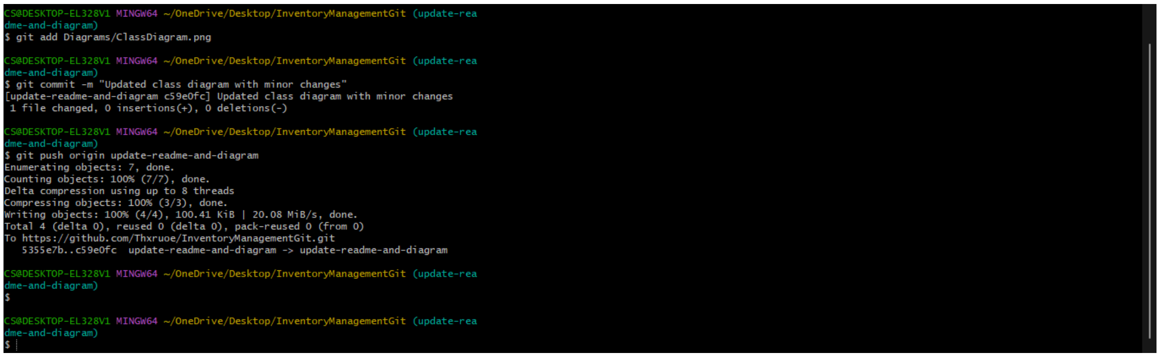
Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

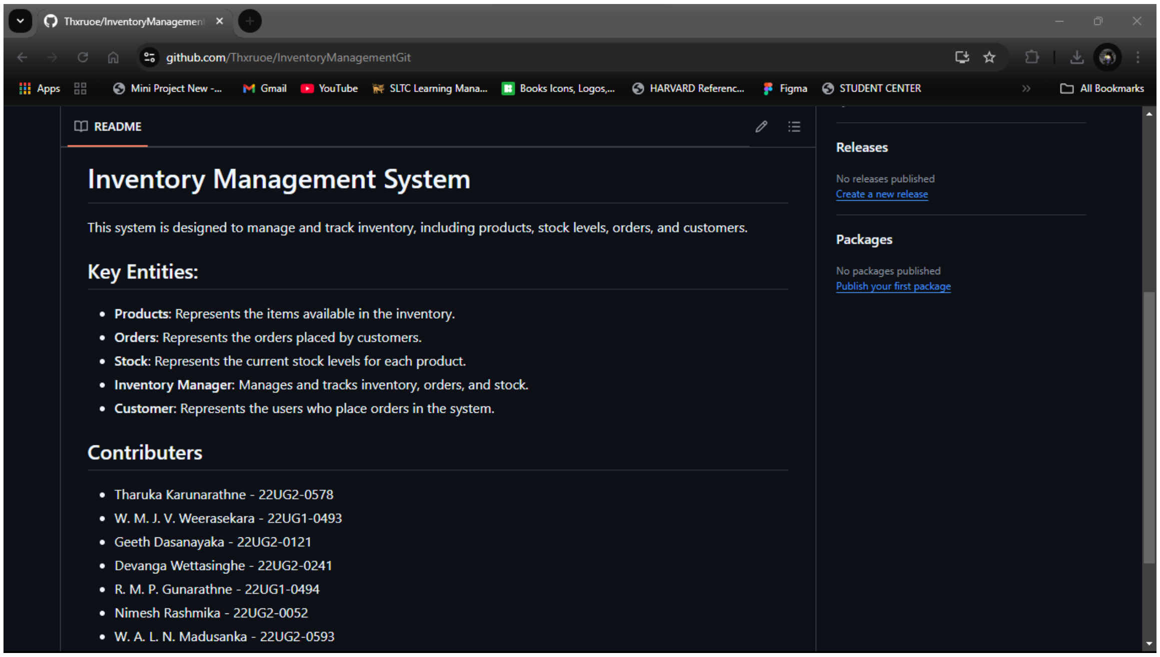
Notifications · Customize

You're receiving notifications because you're watching this repository.

1 participant



```
CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$ git add Diagrams/ClassDiagram.png

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$ git commit -m "Updated class diagram with minor changes"
[update-readme-and-diagram c59e0fc] Updated class diagram with minor changes
 1 file changed, 0 insertions(+), 0 deletions(-)

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$ git push origin update-readme-and-diagram
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 100.41 KiB | 20.08 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Thxruoe/InventoryManagementGit.git
   5355e7b..c59e0fc  update-readme-and-diagram -> update-readme-and-diagram

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$

CS@DESKTOP-EL328V1 MINGW64 ~/OneDrive/Desktop/InventoryManagementGit (update-rea
dme-and-diagram)
$
```



## Inventory Management System

This system is designed to manage and track inventory, including products, stock levels, orders, and customers.

### Key Entities:

- **Products**: Represents the items available in the inventory.
- **Orders**: Represents the orders placed by customers.
- **Stock**: Represents the current stock levels for each product.
- **Inventory Manager**: Manages and tracks inventory, orders, and stock.
- **Customer**: Represents the users who place orders in the system.

### Contributers

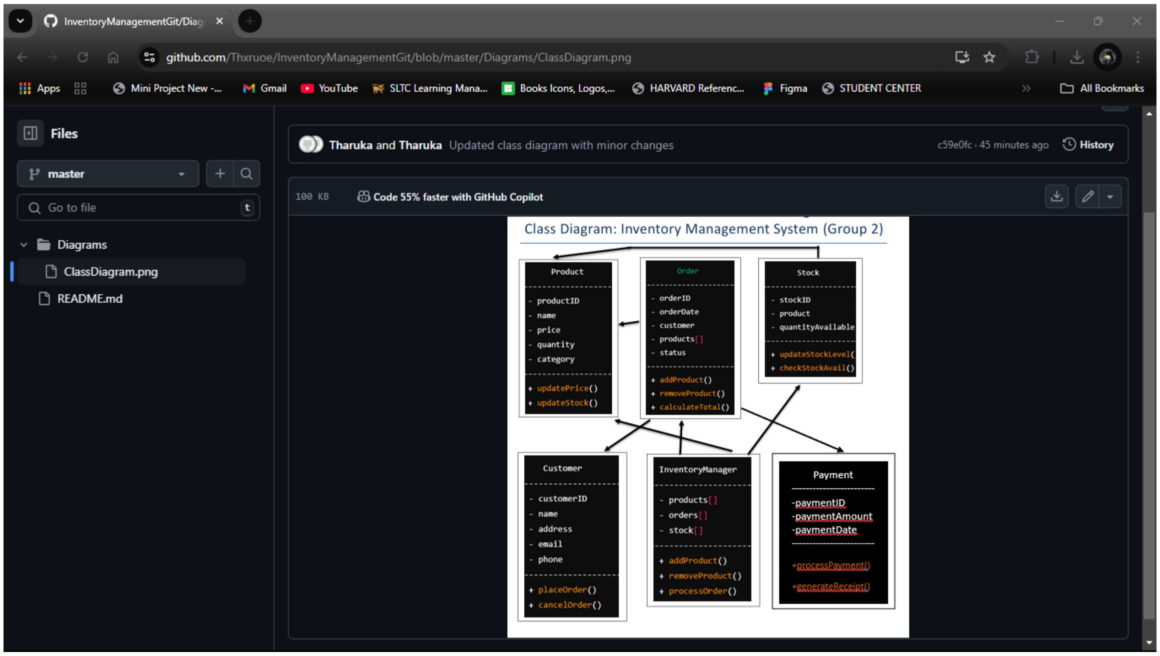
- Tharuka Karunarathne - 22UG2-0578
- W. M. J. V. Weerasekara - 22UG1-0493
- Geeth Dasanayaka - 22UG2-0121
- Devanga Wettasinghe - 22UG2-0241
- R. M. P. Gunarathne - 22UG1-0494
- Nimesh Rashmika - 22UG2-0052
- W. A. L. N. Madusanka - 22UG2-0593

**Releases**

No releases published
Create a new release

**Packages**

No packages published
Publish your first package



Tharuka and Tharuka Updated class diagram with minor changes c59e0fc · 45 minutes ago

Class Diagram: Inventory Management System (Group 2)

## 3. Challenges Faced

- Branching and Merging: Initially, managing branches and ensuring changes were correctly pushed to the remote repository was a challenge. However, this was overcome by creating a new branch and using Git commands such as git checkout, git commit, and git push.

- Conflict Handling: While no actual conflicts were encountered, learning about how to handle merge conflicts (using git merge and resolving issues locally) was important to ensure smoother future collaboration.

## 4. Key Learnings from Using Git:

- Version Control: Understanding the importance of version control for collaborative projects. Git allows easy tracking of changes, collaboration across team members, and reverting to previous versions if necessary.

- Branching and Merging: Gained hands-on experience in using branches to implement changes independently and merging them back into the main branch, which is essential for team collaboration.

- Pull Requests: Learned how to create and review Pull Requests, a key aspect of collaboration in GitHub. PRs ensure that code changes are reviewed before being merged into the main codebase.

- Conflict Resolution: While no conflicts arose, We learned how to handle them effectively. Understanding how to use Git to manage conflicting changes in the code is an essential skill for working in teams.

## 6. Conclusion

This task provided a comprehensive understanding of how to use Git for version control, collaboration, and handling changes in a project. It demonstrated the power of GitHub for managing a software development project, even for a simple task like version-controlling a class diagram.