Final Report
Methods and Tools in SW Development

I. Group Information

Group Number: Group 8
Group Member names/netIDs:
- Matthew Morgan
    - Dmm765
- Robert Gilliand
    - Rsg196
- Han Berdivev
    - Hb1136
- Brook Tebeje
    - Btt109

Group GitHub/GitLab repo link:
https://github.com/han191299/Methods-Tool-GroupProject-Fall2022

Who was assigned to do what?

Han Berdivev- Helped with both the design document. Helped with the code for the implementation by adding Comments and doing debugging for the code as well helped with the classes

Brook Tebeje- Did the bulk of the coding aspect of this project.

Matthew Morgan- Helped with design document and did the final report as well as minorly helping with code.

Robert Gilliand- Helped with the design document, added Comments to the Code as well as debugging code.

II. Updated Detailed Class Diagrams

When creating your project, did any of your classes change? Why did they have to change? Were the changes adding or removing items?

We have essentially changed all the class diagrams the only one not changing is the Account class diagram. We made one changes to Inventory class that being taking out the Show_electronic_inventory() because we realized that we only needed one category for a four person group and this will make the programing aspect a little lighter on our group. The biggest change that was made was the complete removal of the Item Category class the reason for this was because like mention before we moved from multiple categories of products to just one. Now that the Item Category is removed, we must meet the 4 class requirement so we made a Review class this class will allow User to leave reviews of the website.

Account has no change
Shopping Cart has no change
Inventory- remove the Show_electronics_inventory() function
Item Category- was completely removed
Review- replaced the Item Category

** after this point, add any class diagrams for classes that did change. Make sure it's the entire diagram, not just the items that were added/removed/edited -- one class diagram per page

Inventory
Cost ItemID Classification
Show_books_inventory() Edit_stok()

- Show_books_inventory()- shows the inventory of the books
- Edit_stock()- is to keep the stock updated

Review
Author of Review Rating Reviewer Name Content
Add_Review() Remove_Review() Organize()

Add_Review()- gives you the ability to add a review
Remove_Review()- lets you take down your review
Organize()- gives you the option to order the reviews in ascending or descending order

III. Conclusions

If you were to flesh out your project more, what requirements do you think could be added?

I would add a non-functional requirement that does not allow the service to use user's billing information except for the purpose of selling products and allowing the user to look and edit this information. Another requirement I would add would be to let users be able to contact a customer support team for any concerns this would be a functional requirement.

Are there any overall design choices you wish you could have changed in hindsight? What are they and why?

I think for the most part we changed a lot from our design and overall we are pretty happy with it but one thing is that we wish we could have made it more appealing to the eyes but we were not able to do this task. I wish we implemented more of a connection between inventory and review classes so that we could do reviews link to specific books.

What difficulties did your group have?

Most of our difficulties came from our lack coding skills in Python and the SQL connector syntax. This made it hard for everyone to participate in some parts of the project and thus led us to work on different parts of the projects and then come together with what everyone had.

What did your group learn overall from the project?

Overall, we learned how a simple ecommerce cite would look on the back end of things and thus gave us a little insight on what goes on in the background when we are doing online shopping.

Another important thing we learned is that since there was a drastic difference in ability of doing the programming portion that we each had to do separate parts of the project and then come back together with the finish product.

Lastly While it was more of a biproduct it help some of us become more comfortable with the Linux OS.