Final Report

Methods and Tools in SW Development

**I. Group Information**

Group Number: *Section 501 (Online)*

Group Member names/netIDs:

●  *Spencer Hall*

*○ jsh278*

*● Molly Granger*

*o mmg423*

*● Jeffery Burns*

*o jcb1251*

Group GitHub/GitLab repo link: [spencerhall01/CSE-2213-501: Methods and Tools in Software Development (Online Group - Spring 2023) (github.com)](https://github.com/spencerhall01/CSE-2213-501)

Who was assigned to do what?

*Jeffery Burns - help code tables in setup\_db.py, view\_order\_history.py, code review, testing*

*Molly Granger - help code tables in setup\_db.py, code review, testing*

*Spencer Hall - functions, menus, testing*

**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 did not really implement the derived classes; this created more complexity than was needed given our situation and the time we had.*

*Our classes ended up being represented as elements in our database, while the associated methods ended up being functions coded in separate files and called in one main file.*

*Honestly, given our issues (explained more on the next page), we just tried to get something working.*

\*\* 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

*N/A*

**III. Conclusions**

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

*There definitely needs to be more fail-safes and error-checking. This became evident when performing manual functional testing; it quickly became aggravating to have to constantly run the code again and again and go through the same steps just to test a particular portion or error.*

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

*Not really, we think our design choices were mostly good. It was the implementation that was the problem; we waited too late to start and did not have time to properly implement our design. On top of this, there was poor communication and distribution of work within the group, which led to conflict.*

What difficulties did your group have?

*There was poor communication and coordination among group members. During the second phase, we had difficulty coordinating and assigning work. This led to some conflict within the group (pre-split).*

What did your group learn overall from the project?

*Despite our troubles, we did get some experience coding and integrating a backend/database with a simple console frontend. We also gained experience with object-oriented class and database design; it was interesting to see the idea of “classes” extends to and is implemented when databases are involved.*

*Most importantly we learned how critical team dynamics can be to the success or failure of a software project. Communication, coordination, regular meetings, and distribution of work are as important as coding.*