Sprint Retrospective – Sprint 1

CS 4321 - Software Engineering 1

Group 1

Omar Batyah, John Bedene, Austin Hufstetler, Bethany Norton

October 4, 2017



1. **Progress Report**

*Provide a table listing the user stories you committed to for this sprint and their status.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **User Story** | **Github break down** | **Story Points** | **Status** |
| 12 | Edit Book Listings | #3 Create, Read, Update, Delete Commands, & #5 Create Book Table in Database, & #4 Create Book Class | 5 | Closed and completed |
| 1 | Search books | #19 Allow to Search Books, & #6 Search books, | 1 | Closed and Completed |
| 14 | Create Employee | #7 Create Employee Abstract Class, & #8 Create Associate Class, & #9 Create Manager Class | 5 | Closed and completed |
| 10 | Create Member | #13 Create Abstract User Class, & #14 Create Member Class | 5 | Closed and completed |
| 5 | Login/out | #22 Method to Authorize User | 8 | Closed and completed |
| 16 | Create pins/pwd | #20 Generate Pin/Password | 5 | Closed and Completed |
| 2 | Become Member | #7 Create Employee Abstract Class, & #8 Create Associate Class, & #9 Create Manager Class | 1 | Closed and completed |
| 3 | Check out book | #21 Checkout a book | 9 | Open, In Progress |

Completed Story Points: 30

1. **Retrospective.**

*Provide a bulleted list of specific steps you can take to improve your process of creating working software.*

* We had an issue with our guesses being off on story points. Somethings we thought would take longer, took less time, while other we took took more time than we thought it would take. To improve that, we just need more experience with working in sprints themselves to become better judges of assigning story points.
* Not using eclipse because the amount of header that eclipse creates makes it difficult to switch between coders/computers without issues.
* As a group, we could have worked together more efficiently if we had started from the same understanding of materials and tools. eg. github and databases
* Make sure we have close collaboration when working on closely related requirements. Although we met weekly and we were constantly helping each other, each member had an unrelated set of tasks. It will not be possible to work on completely unrelated stories in the next sprint, so we will need to stay updated on the details of each other's’ tasks so we don’t have conflicts.
* Create diagrams to show how each class relates to other classes. This will help us better understand what each class needs to implement, what methods need to be created, and what data needs to be stored.