Iteration 1

Sprint Meeting

On 02/25/19, Team 12 held their Sprint meeting at Gallagher Library to set the team's goals for this sprint. That included the creation of the sprint backlog, and the tasklist for iteration 1. Group members Jacob Cuke, Matthew Allwright, Karim Beyk, Zander von Neudegg, and Jung Hyun Sohn attended. The meeting lasted for an hour.

Sprint Review Meeting

Team 12 held their sprint review meeting at 03/01/19, with every member in attendance. We tested the program as a group at Math Sciences to demonstrate the full functionality of the project so far. After reviewing, the team agreed that the program worked perfectly in terms of our expectations from the sprint meeting. We have discussed the next sprint backlog that we wish to implement, and have discussed how the team is functioning, and if any changes were needed, none of which came to mind. The meeting lasted one hour.

Retrospective Meeting

Team 12 held their retrospective meeting online through Slack on 03/01/19. The team discussed the specifics of the program and the databases, and how we wish to continue implementing our project using a text-based database. We also discussed if there would be anything that was not very efficient, and if there was anything we could stop doing. We discussed hence that less time was needed for the daily scrums, due to the team currently being very proactive and responsible such that if there was anything blocking progress, it would be put out when the problem is faced, not just during the meetings. We will implement this in the next iteration by maybe having our formal scrum meetings every 2 - 3 days. The meeting lasted 30 minutes.

Iteration 2

Sprint Meeting

On 03/10/19, Team 12 held their Sprint meeting at online through the slack platform to set the team's goals for this sprint. That included the updating of the sprint backlog, and the task list for iteration 2. We also discussed our UI and anything we may want to change. All group members attended. The meeting lasted for an hour.

Sprint Review Meeting

Team 12 held their sprint review meeting at 03/15/19, with every member in attendance. We tested the program as a group at Math Sciences to demonstrate the full functionality of the project so far, and to implement our code into the UI. After reviewing, the team agreed that the program worked beyond expectations from the sprint meeting and also from the first iteration. We implemented 11 user stories compared to 3 of iteration 1, and that was due to iteration 1 being a period of learning and creation of the UI, while iteration 2 we could fully implement only the user stories upon the foundation of iteration 1. We have discussed the next sprint backlog that we wish to implement, and have discussed how the team is functioning, and if any changes were needed, none of which came to mind. The meeting lasted two hours.

Retrospective Meeting

Team 12 held their retrospective meeting online through Slack on 03/16/19. The team discussed changes to the UI and sizeability of the UI to fit different resolutions. We also discussed how while we implemented everything individually and met together on Fridays to incorporate the modules into the UI, it may be best to start this finishing process closer to the midpoint and less towards the end, as we progress. Compared to iteration 1, we had less scrum meetings, but we still progressed very smoothly, implementing many user stories and communicating well. The meeting lasted 30 minutes with everyone in attendance.

Iteration 3

Software Development Process

Team 12 follows the “Open” development process. This software engineering paradigm is effective for us because it allows us all to better ourselves as software engineers by witnessing the innovation and new strategies of our team members and applying them ourselves individually, while still maintaining the traditional team structure and hierarchy that we were all comfortable with before becoming a team.

Complete Table of User Stories

Librarian User Story	Implementation Date
I want to order/add books and DVDs from the publisher.	Iteration 1

I want to add links to scholarly papers, articles and journal publications.	Iteration 1
I want to add eBook files to the system.	Iteration 1
I want to be able to browse user's information.	Iteration 2
I want to check whether a user is a registered user.	Iteration 2
I want to be able to check an item's current status.	Iteration 2

Clerk User Story	Implementation Date
I want to register a user in the database.	Iteration 1
I want to accept returned items and update their information in the system.	Iteration 2
I want to accept fine payments and adjust the amount a User owes in the system accordingly.	Iteration 2
I want to be able to take a user off of the blacklist if a fine of $50 or more was paid off.	Iteration 2
I want to check out items at the library.	Iteration 2

Student/Faculty User Story	Implementation Date
I want to check which section of the library a certain item is located.	Iteration 2
I want to be able to reserve items ahead of time to be picked up from the library.	Iteration 2
I want to view links to scholarly papers, articles and journal publications.	Iteration 2
I want to borrow and view electronic books.	Iteration 2
I want to renew a book if there is nobody else waiting for it.	Iteration 2
I want to view the books I currently have out.	Iteration 1
Specifically, for faculty, I want to restrict books to be library only, changing the due dates to be only two hours.	Iteration 3

Test Cases Description
Each of the test cases written are to be performed manually. This has the disadvantage of consuming more time while the software is being tested, but gains the advantage of a guaranteed human viewing and checking the output. This means that the tester has the adaptability and intuition to spot problems that couldn’t be checked with automated test suites. This also allows for the tester to create new test cases on the fly. These test cases are written with the techniques of Black Box, White Box, and Integration testing. Black box testing is used to come up with “universal” test cases that should be tested anyways, such as checking for empty input boxes. White box testing is used to specifically attack a potential weak point in the software, such as including a ‘*’ character in a text field, as these characters are used in a critical role in our database files. And integration testing is used to make sure that separate components of the software are interacting properly, such as how trying to add a resource with a duplicate ID is using both the GUI component, and the resource database component. Some of these tests were written without code inspection, and others were written with code inspection. This was done to create practical test via both the white box and black box testing strategies. This provides the most coverage of potential bug cases, and is a desirable property of a test suite.