**Zobia Asad**

**Hager Ben Mansour**

**Jadon Duby**

**Rory Skipper**

**Prit Patel**

**Parsa Honarmand**

**Final Project Report**

**Iteration 1**

**Sprint Planning Meeting**

**Date:**February 25

**Time:**13:00

**Location:** TFDL 260J

**Duration:**1 Hour

**Attendance:**All

**Decisions:**Created user stories, brainstormed on ways to structure the software, chose to implement UML diagram for first iteration for overall structure of the project. Decided any user stories and/or further functional implementations for this iteration are optional.

**User Stories**

1. As a user, I want to sign into my account so I can borrow books/ magazines/ CDs
2. As a user, I want to sign into my account to I can check my overdue balance.
3. As a professor, I want to sign into my account so I can check if my textbooks for the class I’m teaching are available.
4. As a user, I want to sign into my account to see how much time I have left to return my books
5. As a librarian, I want to login to order more books.
6. As a librarian, I want to be able to approve a blacklist at for instructors
7. As a student I want to check a book’s availability
8. As a student I want to look up how many books I check out so I can tell how many more books I have to return.
9. As a student, I want to sign into my account so I can put books on hold
10. As a professor, I want to put books on reserve so only my students can access them
11. As an admin, I want students who have overdue fees exceeding $50 to be automatically blacklisted.

**Sprint Review**

**Date:**February 28

**Time:**19:00

**Location:**TFDL 350G

**Duration:**1 Hour

**Decision:**Showed team complete UML diagram, and confirmed further plans for the next iteration. Decided to start working with Gitlab for effective and easier communication.

**Attendance:**All

**Retrospective**

**Date:**February 28

**Time:**20:00

**Location:**TFDL 350G

**Duration:** 30 min

**Decision for future:**Decide on further implementations for GUI. Use a task tracking software such as trello. Follow sprint planning more precisely in order to get more work done.

**Things to keep doing:**Regular scrum meetings. Use gitlab.

**Things to stop doing:**N/A

**Attendance:**All

**Iteration 2**

**Sprint planning meeting**

**Date:** March 6

**Time:** 13:00

**Location:**ENGG 207C

**Duration:**1 Hour

**Attendance**: All

**Decisions:**Decided which user stories we would like to implement for this iteration. Decided to split up the work among pairs or 3 members. Anyone willing to work alone was also welcome to. Created a task log that broke down the tasks that needed to be completed in order to implement the chosen user stories. Thought of ways to design the GUI using the UML diagram created which outlines all the backend code. Decided on a tentative 2 week plan for this iteration, taking into consideration everyone’s upcoming busy schedule.

**User Stories**

1. As a user, I want to sign into my account so I can place books/ magazines/ CDs on hold

2. As a student, I want to log in to check if my account is valid.

3. As a user, I want to sign into my account so I can check my overdue balance.

4. As a professor, I want to sign into my account so I can check if my textbooks for the class I’m teaching are available.

5. As a user, I want to sign into my account to see how much time I have left to return my books

6. As a librarian, I want to update book repository.

7. As a librarian, I want to login to order more books.

8. As a librarian, I want to be able to approve a blacklist for instructors

9. As a student I want to check a book’s availability

10. As a student I want to look up how many books I check out so I can tell how many more books I have to return.

11. As a student, I want to sign into my account so I can put books on hold

12. As a professor, I want to put books on reserve so only my students can access them

13. As an admin, I want students who have overdue fees exceeding $50 to be automatically blacklisted.

There are no further user stories added. Throughout iteration 2, user stories 1,2,6 and 7 will be implemented.

**Sprint Review Meeting**

Date: March 15

Time: 13:00

Location: EEEL

Duration: 1 Hour

Attendance: Rory, Prit, Parsa, Jadon, Zobia

Hajer - Away, but updated through Whatsapp

Decisions: Showed team the code written and how it is implemented using Java Swing. Decided we need to rename the variables used in Swing to meaningful variables for the next iteration. Prit and Parsa will continue working on the “hold items” function with Rory’s help for any last minute fixes over the weekend. Jadon will work on the Librarian GUI and “putting books on hold” function for any last minute fixes over the weekend. Zobia will complete the Scrum report. Reminded everyone to update the task log. Decided to use Gitlab more often, post any changes in code soon enough for the team to see.

**Retrospective Meeting**

**Date:**March 15

**Time:**14:00

**Location:**EEEL

**Duration:**30 min

**Attendance:**Rory, Prit, Parsa, Jadon, Zobia

Hajer - Away, but updated through Whatsapp

**Decisions for future:** We decided we need to come up with a weekly time for sprint meetings instead of having to plan them a day or two in advance. This weekly time should work for everyone until the end of the semester. For the next iteration, we will start implementing the user stories required for administrators and professors, and complete the functionality required for students and librarians. We will also start focusing on making the GUI more aesthetically pleasing.

**Things to keep doing:**Daily scrum meetings. Keeping the team updated on the latest work done. Using Gitlab continuously, and asking for help when needed.

**Things to stop doing:** Booking meeting times a day or two before the meeting, we will try to come up with an organized meeting schedule that we will try to follow until the end of the semester.

**Iteration 3**

**Incremental Model**

For this iteration, our team decided to use the incremental model as the software development process. The incremental product applies the waterfall method in an iterative way. We decided to use this process because after each previous iterative, our core product was evaluated by the client (Professor, TA), where we were given feedback on our “core product”, thus allowing us to create a plan and modifications for the next iteration. This process continues until the product is complete. This approach follows the way our project is structured. Three iterations + a final submission, and feedback after each iteration perfectly follows the incremental model. This model is also useful when staffing is low, which works well since we only have 6 people to create an entire product. This process works with creating small portions of the product at a time, which is demonstrated through our user stories.

**User Stories**

1. As a user, I want to sign into my account so I can place books/ magazines/ CDs on hold.

2. As a student, I want to log in to check if my account is valid.

3. As a user, I want to sign into my account so I can check my overdue balance.

4. As a professor, I want to sign into my account so I can check if my textbooks for the class I’m teaching are available.

5. As a user, I want to sign into my account to see how much time I have left to return my books

6. As a librarian, I want to update book repository.

7. As a librarian, I want to login to order more books.

8. As a librarian, I want to be able to approve a blacklist for instructor

9. As a student I want to check a book’s availability

10. As a student I want to look up how many books I check out so I can tell how many more books I have to return.

11. As a student, I want to sign into my account so I can put books on hold

12. As a professor, I want to put books on reserve so only my students can access them

13. As an admin, I want students who have overdue fees exceeding $50 to be automatically blacklisted.

There are no further user stories added.

**Iteration 1:**overall structure of the project was created, UML diagram, no user stories were implemented during this iteration.

**Iteration 2:**user stories 1,2,6 and 7 were implemented.

**Iteration 3:**user stories 3, 9, 10, 11, 12 were implemented.

**Final Submission:**4, 5, 8, 13 will be implemented.

**Final Submission**

**Updated User Stories List**

1. As a user, I want to sign into my account so I can place books/ magazines/ CDs on hold.

2. As a student, I want to log in to check if my account is valid.

3. As a user, I want to sign into my account so I can check my overdue balance.

4. As a professor, I want to sign into my account so I can check if my textbooks for the class I’m teaching are available.

5. As a user, I want to sign into my account to see how much time I have left to return my books

6. As a librarian, I want to update book repository.

7. As a librarian, I want to login to order more books.

8. As a librarian, I want to be able to approve a blacklist for instructor

9. As a student I want to check a book’s availability

10. As a student I want to look up how many books I check out so I can tell how many more books I have to return.

11. As a student, I want to sign into my account so I can put books on hold

12. As a professor, I want to put books on reserve so only my students can access them

13. As an admin, I want students who have overdue fees exceeding $50 to be automatically blacklisted.

**Final Submission:**User stories 4 5, and 13 will be implemented. Testing will be implemented. User interface will be implemented in a user friendly and appealing manner. Now Icons are added to all tables to easily show which type of material you’re looking at.

**Update on User Stories and Requirements**

For our final submission, no new user stories were added. However, during the requirements gathering meeting, the client stated that a new user should be able to create a new account to access the library system. During the meeting, it was stated that an administrator should also be able to create new roles. These requirements will be implemented in the future when new skills will be developed, and more resources and time will be available. User stories 8 and 12 were not fully implemented due to lack of resources and time. These user stories can be easily fully implemented in the future with new skills and more time and resources available. For example, the part in user story 12 that’s not implemented is “only my students”. A professor is able to put books on reserve, but not only so his students can access it. This can be implemented once we’re able to learn databases, where we can organize students under a specific professor. The rest of the user stories that needed to be implemented for the final iteration have been successfully completed.

**Meetings**

Our team had their weekly meeting on Wednesday, April 3rd and Wednesday, April 10th. Throughout the meetings, we discussed the remaining tasks that needed to be completed. Tasks were split between the team members, and any extra help needed was provided by other members. A demo was booked with the TA during one of the meetings as well to display the final product. All team members attended the two final meetings.

**Test Suites**

The methods of testing that were used to test the back end software were JUnit test cases and code inspection. The classes that were tested using a code inspection list were UserManager and TextDatabase which hold a large bulk of the back end operations required for the user interface to work properly. MaterialManager was tested using JUnit test cases. This class was tested using automated testing as opposed to the others due to its method contents and the testability of these methods due to their return data characteristics. Code inspection was used instead of automated testing as automated testing would test methods for obvious output(UserManager) or it would not thoroughly test the output of a method at all(TextDatabase). Using automated unit testing it is easier to see a methods functionality as well as it’s reliability as expected outcomes are tested with actual outcomes. Code inspections help test the quality, functionality and all around efficiency of the code as they help in critically analysing the white box characteristics of any given piece of code used in a larger ultimate system. The automated unit test cases can be found in the librarysystem.testing package. By testing these three backend classes a bulk of the backend operations are tested for quality, functionality satisfaction, efficiency and reliability.

**Deployment date:** Friday, April 12th, 2019