**CS3354 Software Engineering   
Final Project Deliverable 2**

**Group 10**

***The Bookaholics***

Katherine Goins

Felicity Pawlowski

Abhijeet Medtia

Jennifer Bohling

Noah Sims

Juan Cantu

Jonathan Hocevar

**1. [5 POINTS]** Well described delegation of tasks, i.e. who did what in the project.

Now that your project is complete, you are required to submit the delegation of tasks

from beginning of the project until the end. Please make sure to fairly distribute tasks

in the team and remember that in the end of the semester, each member of a team will

receive the same grade. See grading policy below for more detail. If no/poor

contribution by a member, please specify clearly so that we can grade each student

fairly.

**All team members**  **- DONE**

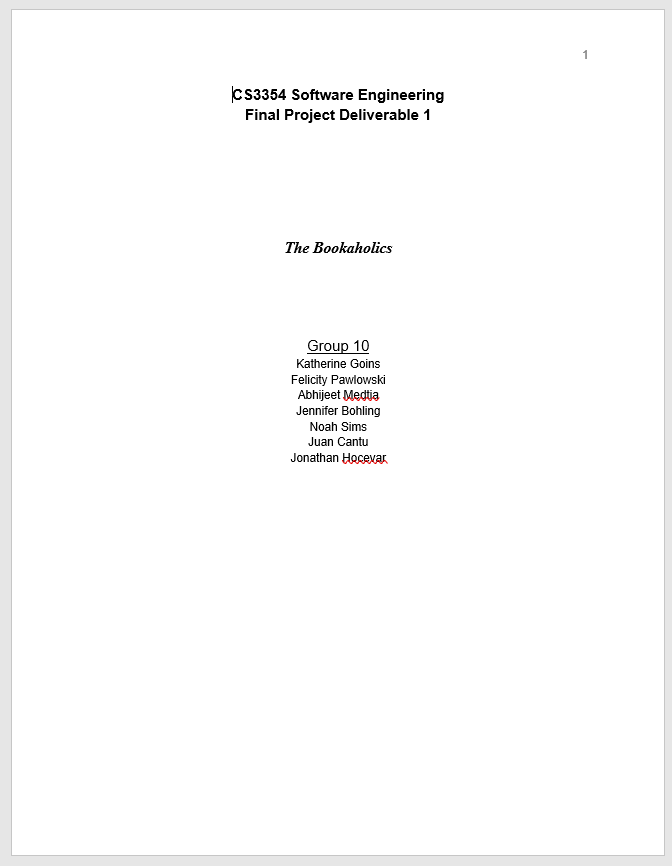
* Throughout the document team members are listed by what they did. Overall, there was an even distribution of work and everyone did their part.

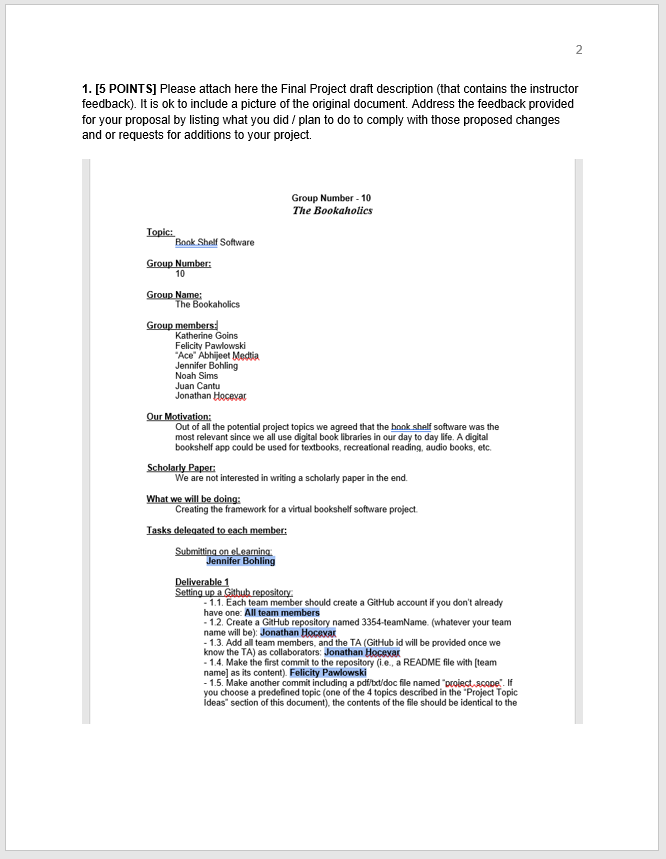
**2. [5 POINTS]** Everything required and already submitted in Final Project Deliverable

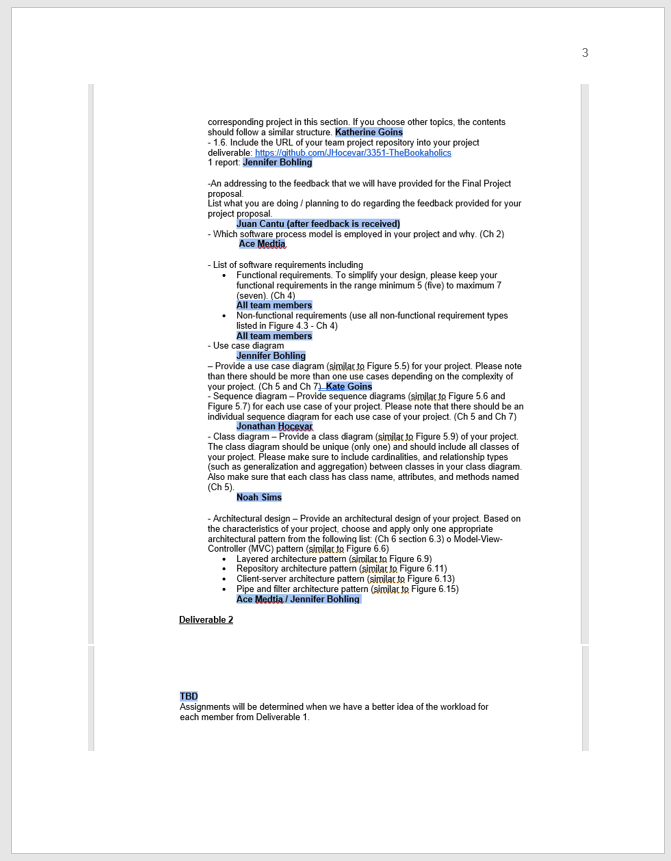
1. Please specify this part as “Project Deliverable 1 content”.

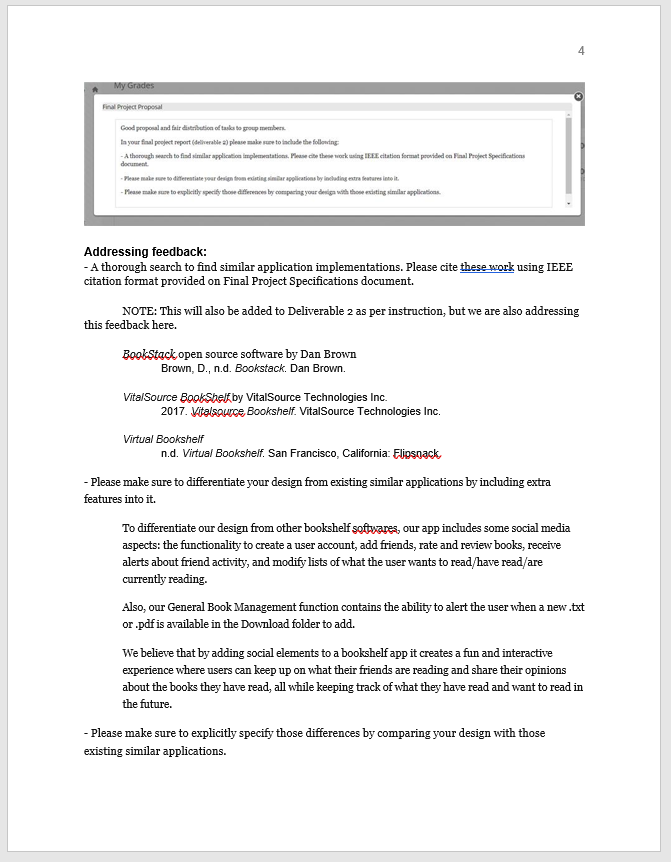
**Jennifer Bohling**  **- DONE**

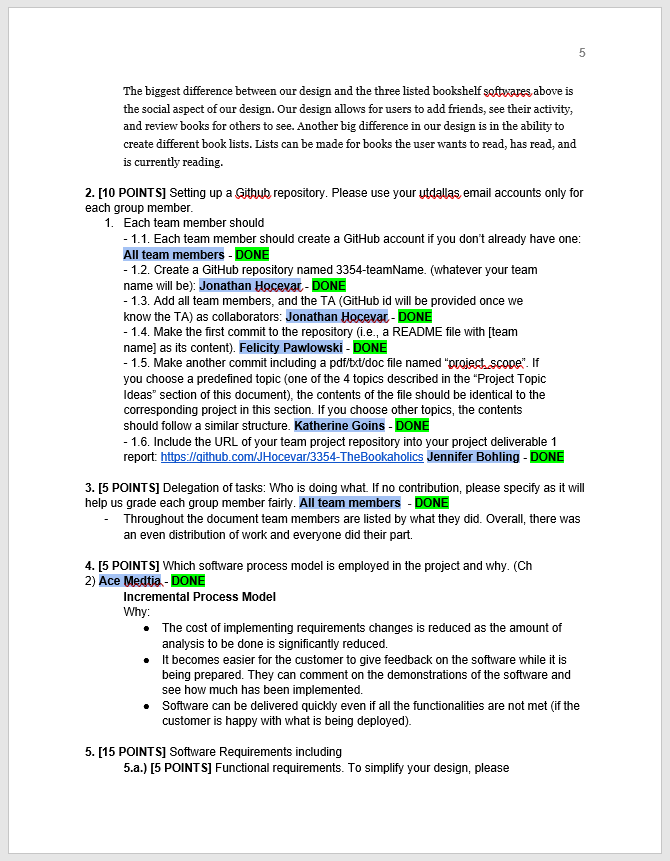
Project Deliverable 1 content:

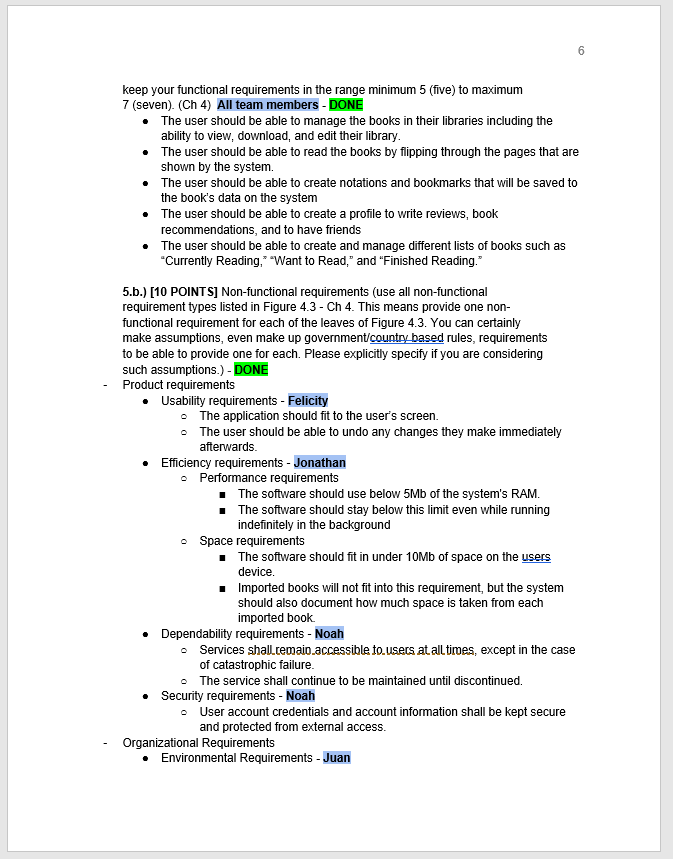


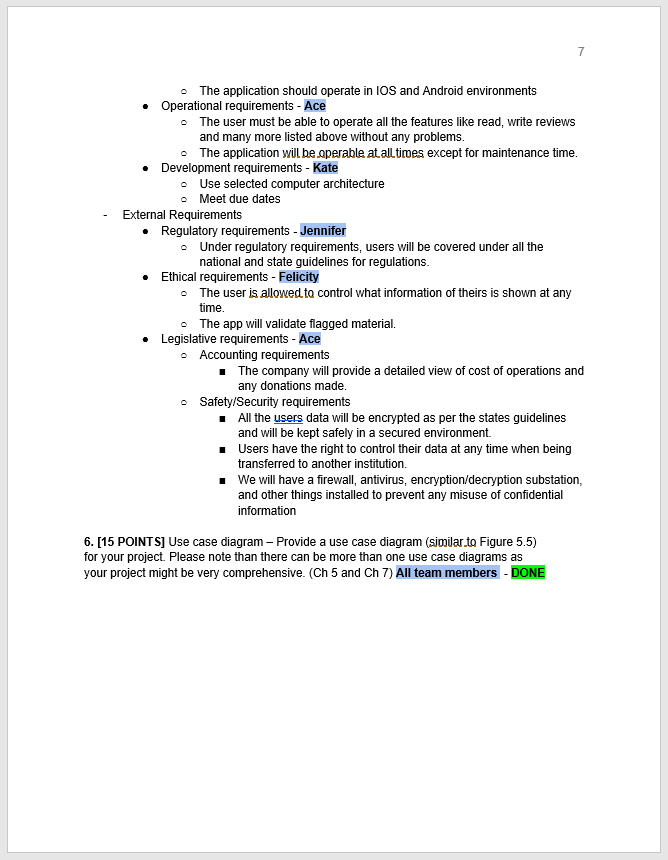


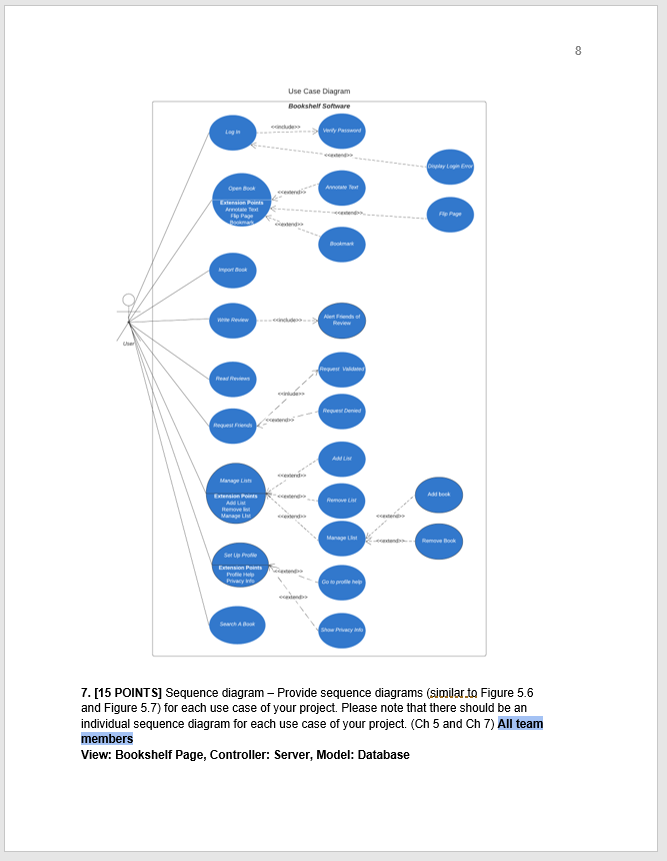


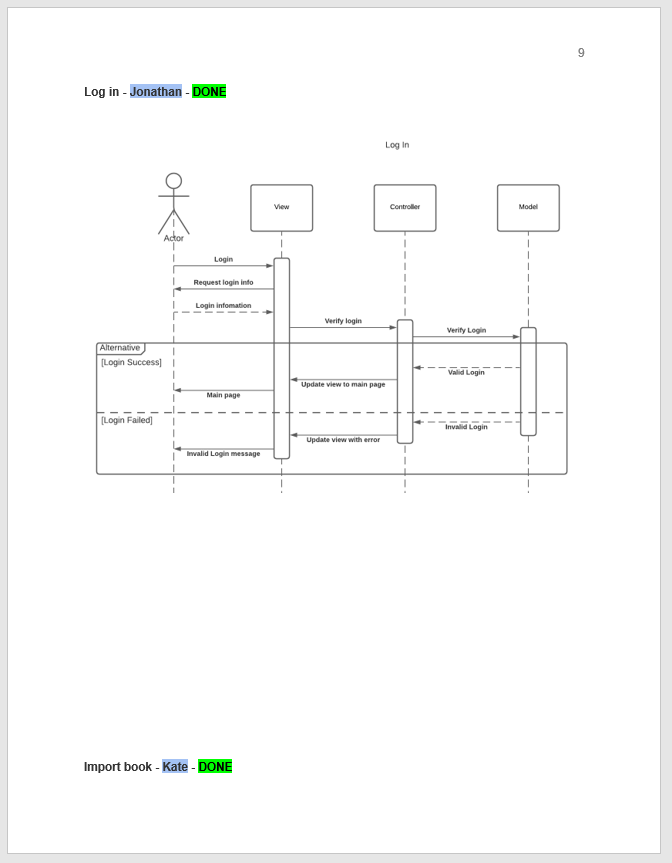


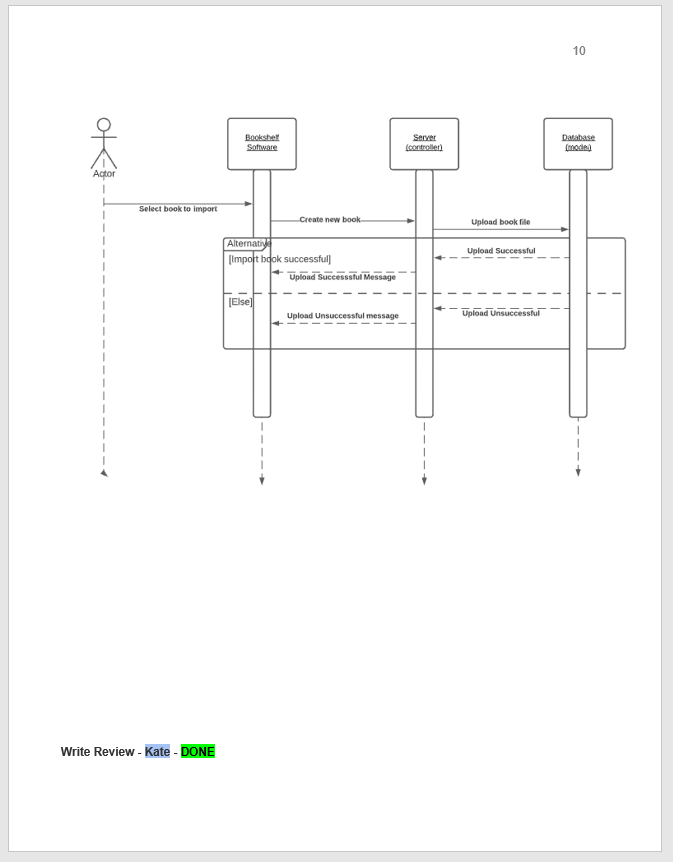


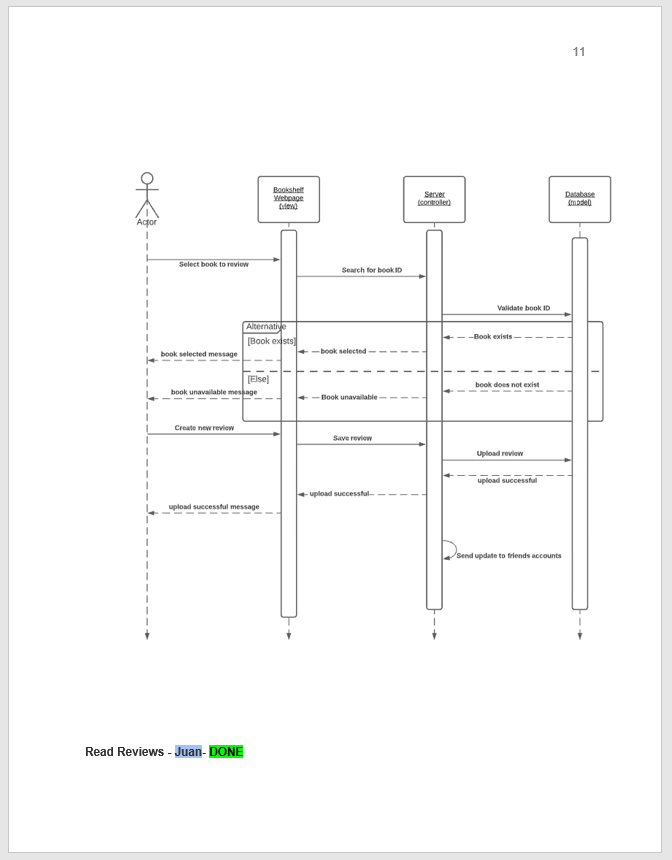


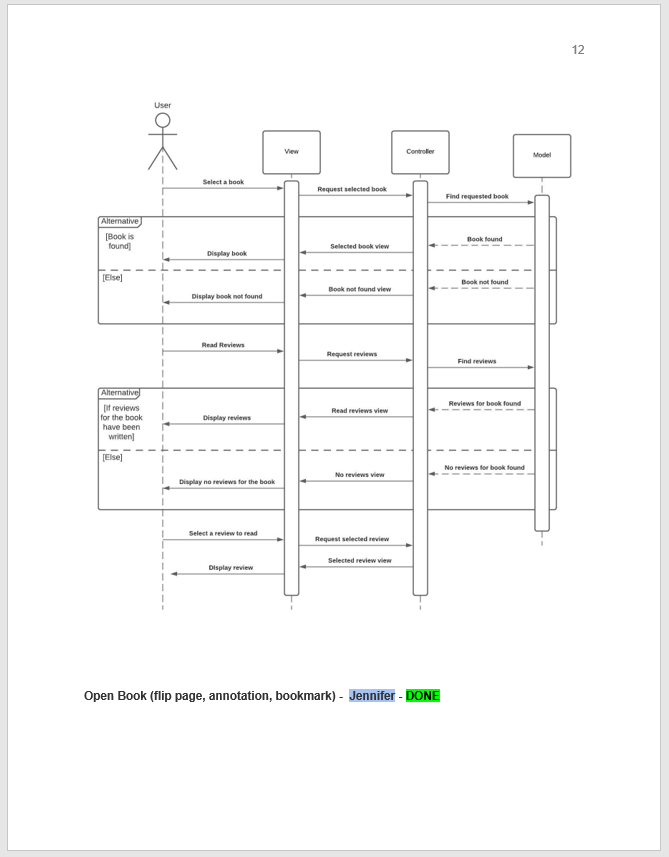


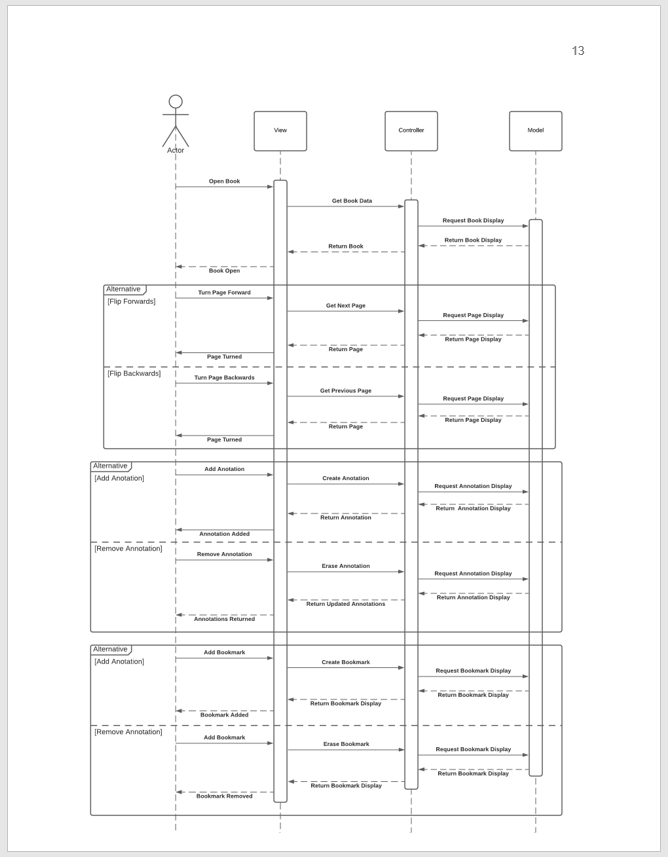


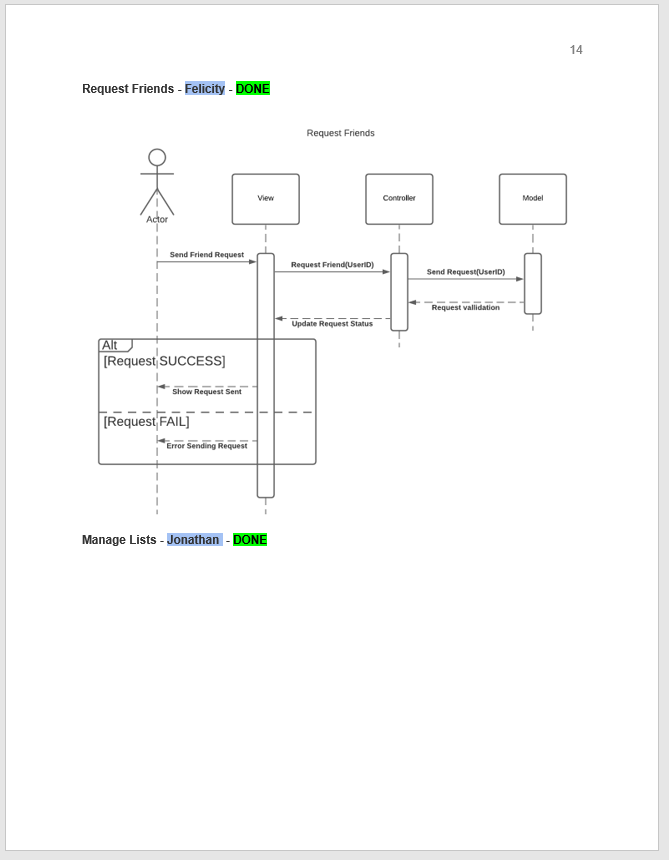


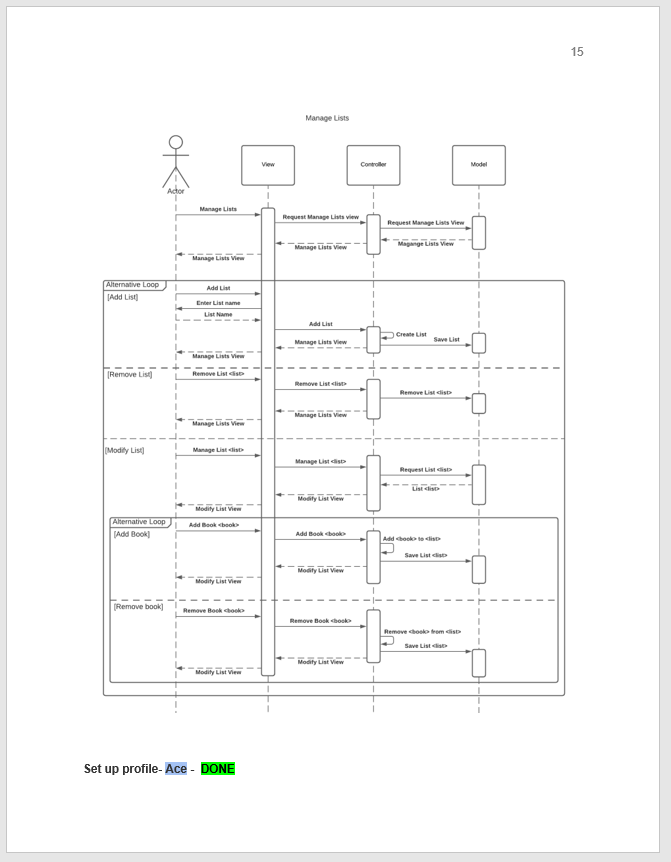


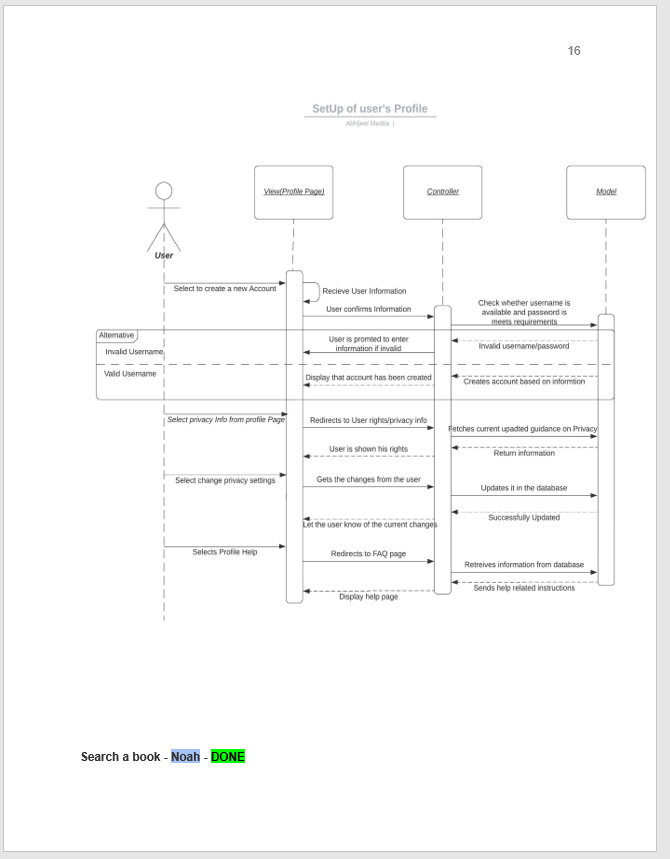


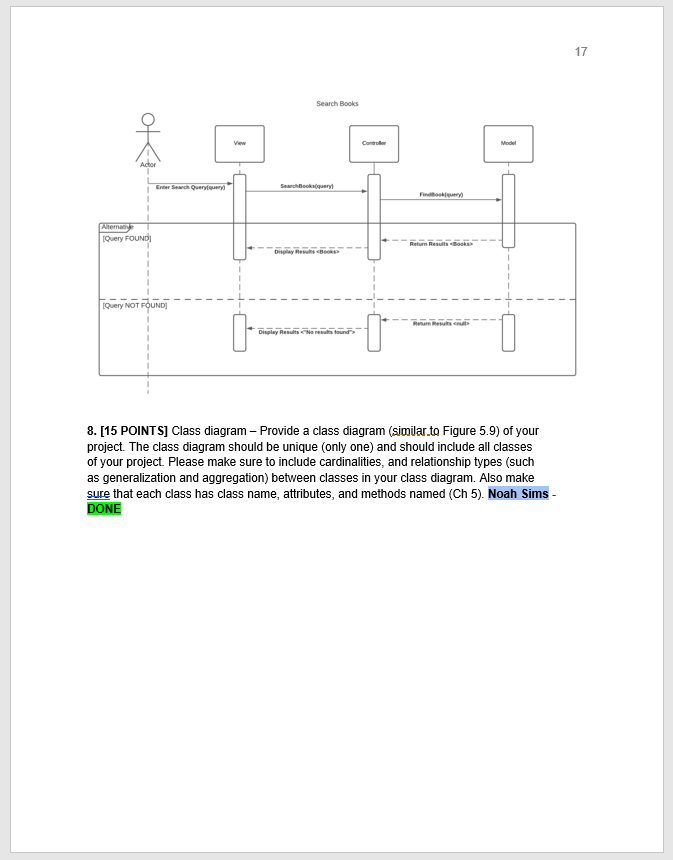


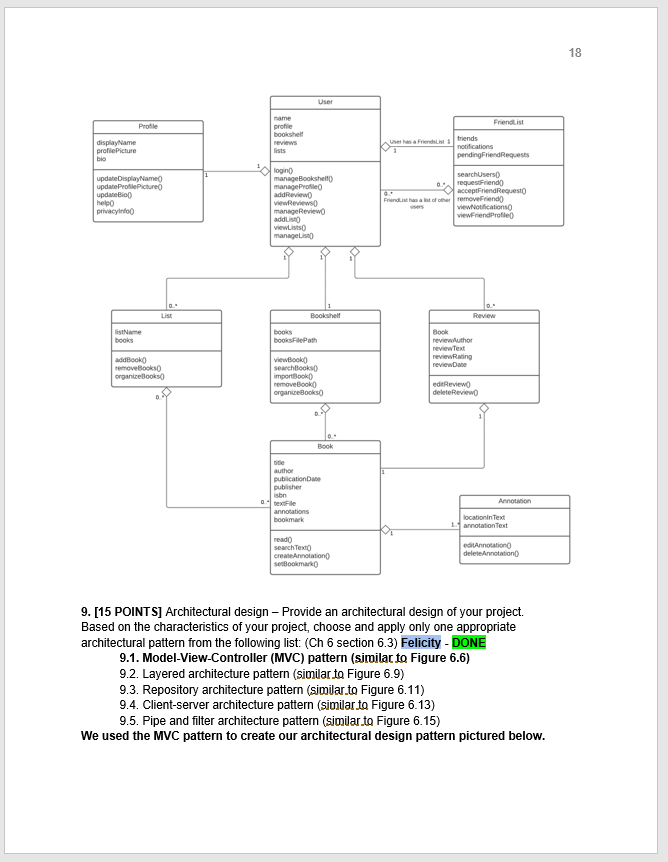


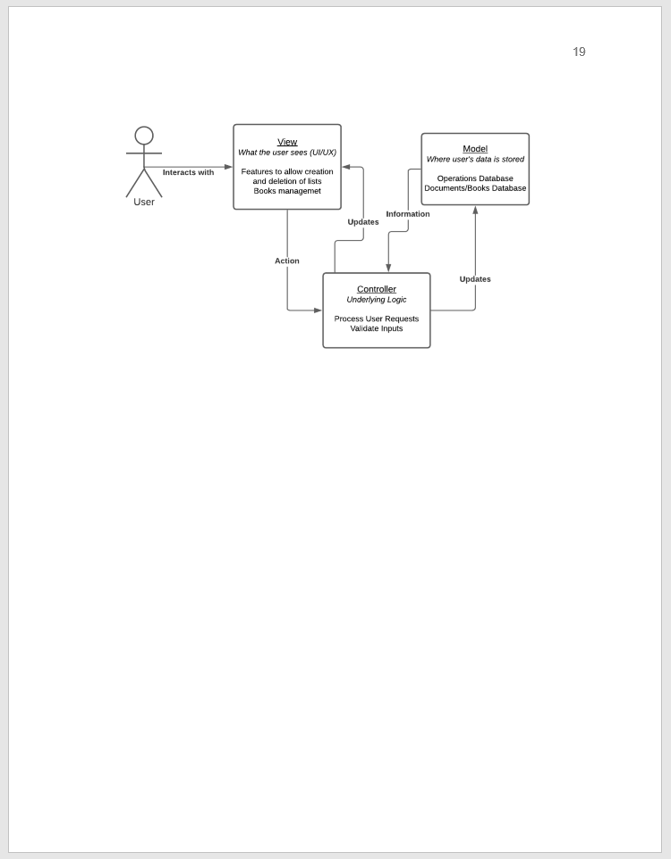


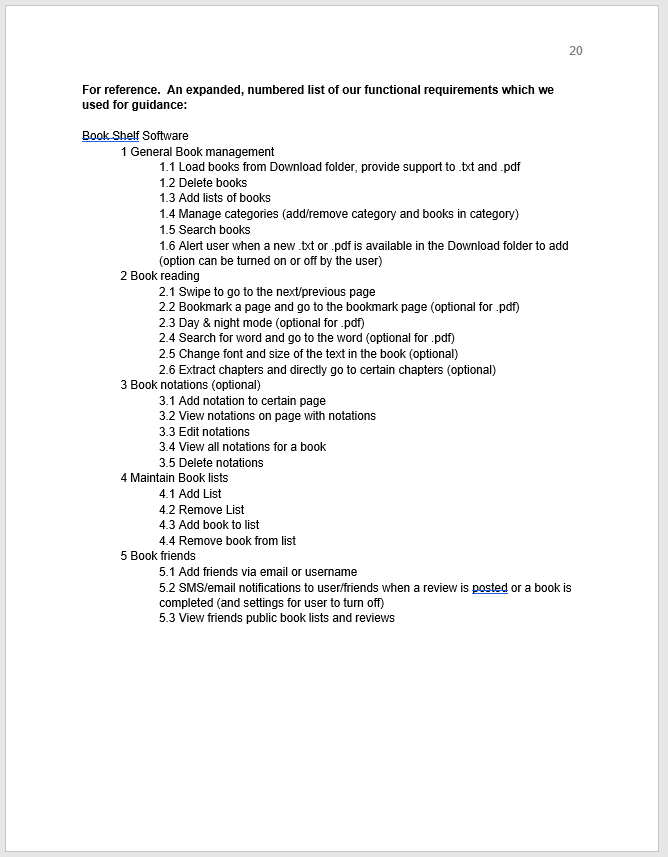












**3. [5 POINTS]** Choose only one of the following two options and specify clearly which

is your choice. This will help us post a live presentation schedule. Please understand

that we wouldn’t know which group will present live, and which group will prerecord instead before the due date of final project deliverable2. So, thank you very

much for your patience. We will post a presentation schedule once we have each

group’s submission.

- Option 1: Present live during scheduled class time via BBC (Blackboard

collaborate). Each member of the group has to be present and participate in

this option. No need to prerecord your presentations if you choose option 1

as your live presentations will be recorded. Still, you should include your

(non-recorded) presentation slides into your project deliverable2 submission

bundle.

**We choose Option 2: Prerecord**

**All team members**-**DONE**

**Link to our presentation with captioned transcript:**

<https://web.microsoftstream.com/video/9f0d3a21-f12c-4e11-9e57-fb84ab3ef052>

- Option 2: Prerecord your captioned presentations. Save your captioned

recording at a URL. Then provide this URL (where you host your

prerecorded presentation) exactly here, inside your Final Project

Deliverable2 report so that we post it for students to access. Remember:

Captioned recordings are UTD requirement. Make sure each group member

talks in that recorded presentation. DO NOT SEND TO US your

presentation recordings, as your captioned pre-recording file will be too

large. Only provide its URL here. Still, you should include your (nonrecorded) presentation slides into your project deliverable2 submission

bundle.

**4. [35 POINTS]** Project Scheduling, Cost, Effort and Pricing Estimation, Project

duration and staffing: Include a detailed study of project scheduling, cost and pricing

estimation for your project. Please include the following for scheduling and

estimation studies:

4.1. **[5 POINTS]** Project Scheduling. Make an estimation on the schedule of your

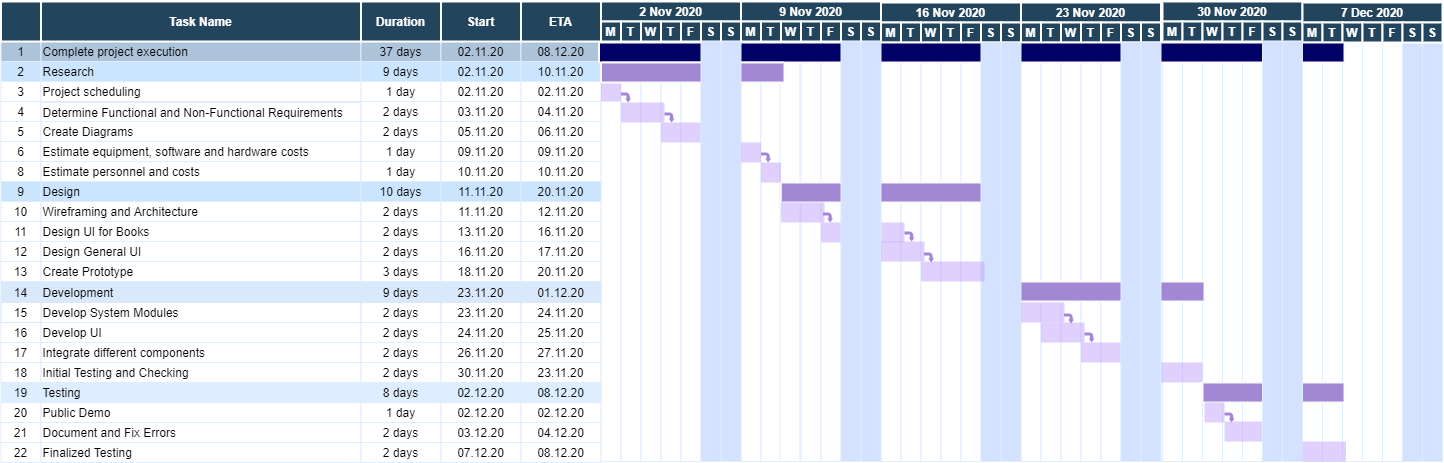
project. Please provide start date, end date by giving justifications about your

estimation. Also provide the details for: **Felicity Pawlowski-DONE**

- Whether weekends will be counted in your schedule or not

- What is the number of working hours per day for the project

We used the assumption of 35 hour work weeks for our calculations as well as 3 team members. This allows for 5 working hours per day per person for the project. Weekends will not be worked (as seen in the schedule below). Our calculations found in the Cost Estimation section gave us the approximation of 7 business days to implement the project. Using this as reference, below is the full project schedule.



4.2. **[15 POINTS]** Cost, Effort and Pricing Estimation. Describe in detail which

method you use to calculate the estimated cost and in turn the price for your

project. Please choose one of the two alternative cost modeling techniques and

apply that only:

- Function Point (FP)

- Application composition

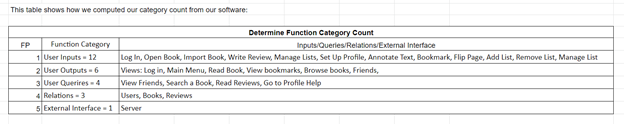
**All team members - DONE**

As a group we decided to use the Function Point algorithmic estimation technique. All team members agreed this was the most useful method for our project. We discussed Function Point versus Application Composition. Application Composition did not suit our project because it is better for estimating the costs involved with building a prototype. Also, it is more useful for projects which use commercial off-the-shelf software, which does not apply to our project. Function Point gained unanimous approval from our group because of how easily it estimates the effort needed to develop software, focused on quantifying the functions contained in our software.

We decided on the parameters as follows:

1. Determining function category count.

For count, we discussed at length which parts of our software performed user input, user output, user query, relation, and external interface. We used our previous Use Case and Sequence diagrams for reference and put the results into a table for clarity:

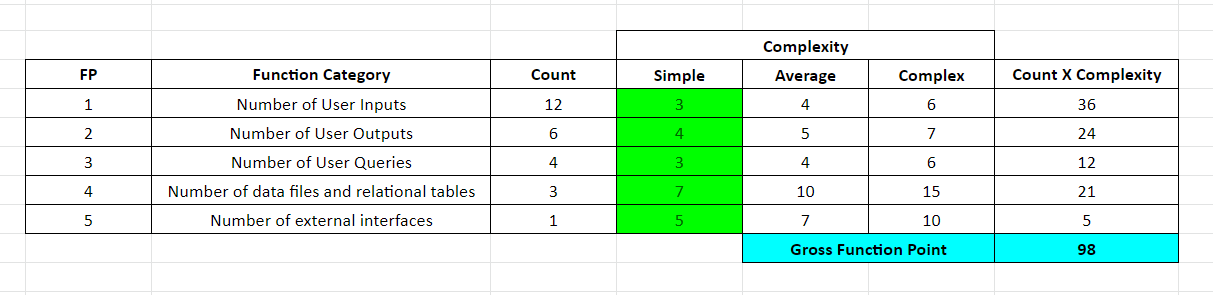


2. Determining complexity.

We decided that the complexity parameter of our project was **simple**, due to the way it only manipulates data in small, simple ways.

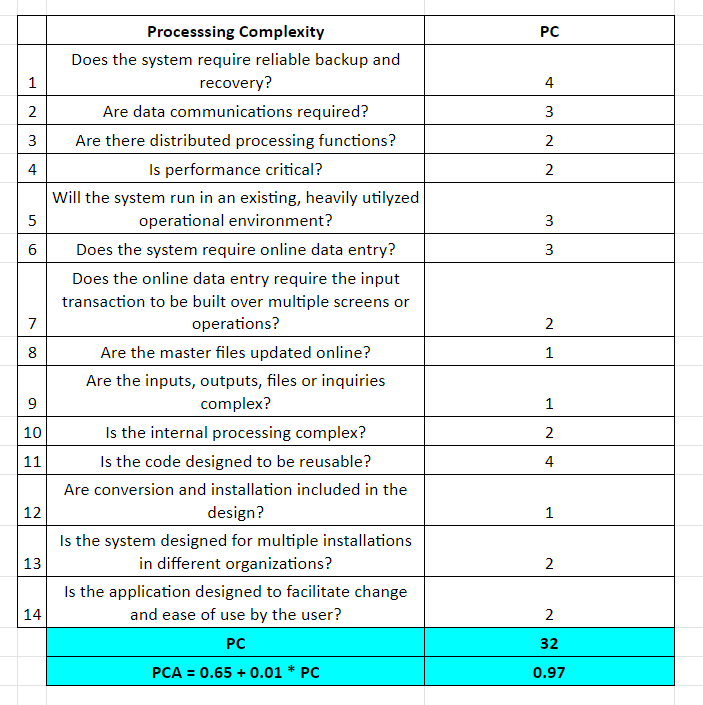
3. Compute gross function point (GFP).

With our complexity and our count values decided upon, we calculated our Gross Function Point value:



4. Determine processing complexity (PC).

As a group, we discussed each processing complexity, considering how it related to our software, and came up with the following results:



5. Compute processing complexity adjustment (PCA).

As seen above, the sum of our PC values equals 32.  
 PCA = 0.65 + 0.01(32)

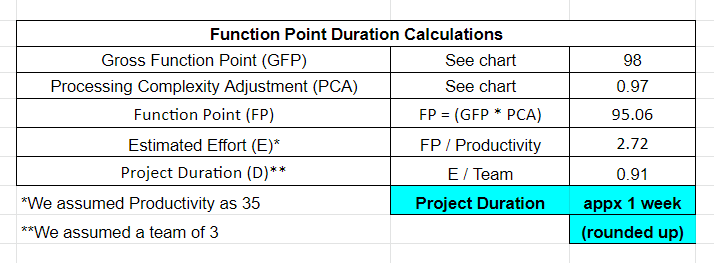
Therefore, our PCA = 0.97



6. Compute function point (FP) using the formula: FP = GFP × PCA

Use FP and estimated effort(E) to find Project Duration (D), rounded up.

As indicated in the table below, we assumed a small team of 3 people worked on this simple project with a productivity level of 35, resulting in approximately 1 week of project duration.



4.3. **[5 POINTS**] Estimated cost of hardware products (such as servers, etc.)

**Ace Medtia -** **DONE**

|  |  |
| --- | --- |
| **Hardware Cost** | |
| Things Needed: | Cost: |
| Hard Drive: 10\*1TB SSD | $1,200 |
| Power Supply | $100 |
| Memory : 64GB ECC RAM | $329 |
| SUPERMICROServer Motherboard | $929 |
| Chassis: Supermicro 1U Chassis | $160 |
| CPU:Xeon W-2245 | $667 |
| Server Software | $250 |
| Processor | $0 (Included with MotherBoard) |
| **Total Cost** | **$3,635** |

4.4. **[5 POINTS]** Estimated cost of software products (such as licensed software, etc.) **Ace Medtia -** **DONE**

|  |  |
| --- | --- |
| **Software Licence/Product Cost** | |
| Apple App Store Fee: | $99 |
| Google Play Store Fee: | $25 |
| Microsoft Windows Server Standard License | $525 |
| Mac OS | $129 |
| **Total Cost** | **$778** |

4.5. **[5 POINTS]** Estimated cost of personnel (number of people to code the end

product, training cost after installation)

**Ace Medtia -** **DONE**

|  |  |
| --- | --- |
| **Cost of Personnel** | |
| No. of Engineers Required | 3 |
| Total Duration of Work | 7 business days |
| Per/Hour Salary of an Engineer | $80 |
| **Total Personnel Cost** | **$13,440** |

**5. [10 POINTS]** A test plan for your software: Describe the test plan for testing

minimum one unit of your software. As an evidence, write a code for one unit (a

method for example) of your software in a programming language of your choice,

then use an automated testing tool (such as JUnit for a Java unit) to test your unit and

present results. Clearly define what test case(s) are provided for testing purposes and

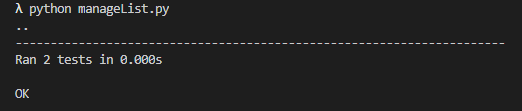
what results are obtained (Ch 8). Include your test code as additional document in

your zip file submitted.

**Jonathan Hocevar-** **DONE**

Most of the methods in our application will be taken straight from the use case diagram as these are the major features. After the GUI is designed and implemented, we will begin work on these major methods and any of their sub/helper methods. These Major methods will be the ones to be tested with unit tests. Most of these tests will be requests or updates to the server. The application will check to make sure the right object was returned/created. In our example, a test case was written for the add\_list method. This method takes the name of a new list to create and adds it to the main application. It also checks to make sure a list with the same name does not already exist before creating the new one. Our test case focused on creating the list and adding it to the application. Below you can see the result of the two test cases, Creating a new list, and trying to create a list that already exists. The code and this image can also be found in the Test Cases folder on the GitHub and is included with this document in the zip folder.

GitHub can be found at: <https://github.com/JHocevar/3354-TheBookaholics>



**6. [10 POINTS]** Comparison of your work with similar designs. This step requires a

thorough search in the field of your project domain. Please cite any references you

make.

* **Juan Cantu, Jennifer Bohling -** **DONE**

The biggest difference between Bookstack [1], Bookshelf [2], Virtual Bookshelf [3], and our bookshelf project is the social aspect of our design. Our design allows for users to add friends, see their activity, and review books for others to see. Another big difference in our design is in the ability to create different book lists. Lists can be made for books the user wants to read, has read, and is currently reading.

To differentiate our design from other bookshelf softwares, our app includes some social media aspects: the functionality to create a user account, add friends, rate and review books, receive alerts about friend activity, and modify lists of what the user wants to read/have read/are currently reading.

Also, our General Book Management function contains the ability to alert the user when a new .txt or .pdf is available in the Download folder to add.

We believe that by adding social elements to a bookshelf app it creates a fun and interactive experience where users can keep up on what their friends are reading and share their opinions about the books they have read, all while keeping track of what they have read and want to read in the future.

**7. [10 POINTS]** Conclusion - Please make an evaluation of your work, describe any

changes that you needed to make (if any), if things have deviated from what you had

originally planned for and try to give justification for such changes.

**Jennifer Bohling -** **DONE**

At the beginning of our project, we tried to assign each task individually. We quickly realized that it was more efficient and produced better results to work on the largest tasks as a group (or in pairs). By re-assigning things to all team members, we were able to work as a team more effectively. The best example of this was our work on the Use Case diagram. That was the most involved part of our project and having all team members contributing to it insured that everyone’s area of focus was clearly represented.

The scope of our project changed a few times when we tried to determine how to make our project stand out from other bookshelf softwares. We toyed with the idea of being able to make purchases for some time. In the end we decided against doing that, because it was a very common idea, and we preferred the more unique idea of having a social aspect to the bookshelf. Additionally, we thought it beyond the scope of this project. The ability to make reading something shareable, trackable, and more interactive, was most appealing. Further, it allowed us to keep the scope of our project simple.

**8. [5 POINTS]** References: Please include properly cited references in IEEE paper

referencing format. Please review the IEEE referencing format document at the URL:

https://ieeedataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf).

It means that your references should be numbered, and these numbers properly cited

in your project report.

[1] D. Brown, “BookStack,” March, 2016. [online]. Available: <https://www.bookstackapp.com>. [Accessed: Oct. 13, 2020].

[2] VitalSource Technologies, “Bookshelf,” *VitalSource Technologies,* 2017. [online]. Available: <https://bookshelf.vitalsource.com/#/user/signin>. [Accessed: Oct. 13, 2020].

[3] Flipsnack, “Virtual Bookshelf,” *Flipsnack,* n.d. [online]. Available: <https://www.flipsnack.com/virtual-bookshelf>. [Accessed: Oct. 13, 2020].

**Juan Cantu -** **DONE**

**Also include:**

**9. [10 POINTS]** Non-recorded (no-voice) presentation slides. No min/max number of

slides enforced. Please make sure that you can complete presentation within **15**

**(fifteen)** minutes.

Following template could be a good start to prepare your presentations. As each

project topic is different, a variety in presentation style is expected and welcome.

- Title of your project together with participants **Everyone**

- Objective of the project designed **Jennifer Bohling**

- Cost estimation **Ace**

- Project timeline (timeline of the project designed, NOT the time you’ve

spent on it) **Felicity Pawlowski**

- Functional and non-functional requirements. If too long, select

representative items.

**Functional: Kate**

**Nonfunctional: Ace**

- Use case diagram **Juan Cantu**

- Sequence diagram for a selected representative operation of the project. **Jonathan Hocevar**

- Class **Noah Sims**

- Architectural design **Jonathan Hocevar**

- Model-View-Controller (MVC) pattern (similar to Figure 6.6)

- Layered architecture pattern (similar to Figure 6.9)

- Repository architecture pattern (similar to Figure 6.11)

- Client-server architecture pattern (similar to Figure 6.13)

- Pipe and filter architecture pattern (similar to Figure 6.15)

- Preferably a demo of user interface design that shows screen to screen

transitions though no full functionality is required. **Felicity Pawlowski**

- OPTIONAL: IF implemented the project, a demo of your implementation. **N/A**

- Conclusion **Jennifer Bohling**

10. OPTIONAL PART. Your program code **(if fully implemented the project, not**

**required otherwise)**. Please note that **implementation is not required for the final**

**project**. Groups are welcome to implement their work, if they choose to do so.

**[This part may qualify for extra credit, if you implement and submit the**

**implementation code together with your project. The extra credit will be**

**determined based on the quality of your implementation]**

**NOT IMPLEMENTED**

**11. [5 POINTS]** GitHub requirement:

GitHub can be found at: <https://github.com/JHocevar/3354-TheBookaholics>

Make sure at least one member of your group commits everything for project deliverable

2 to your GitHub repository, i.e.

- Your final project deliverable2 report **DONE**

- Unit test code for a sample unit of your project **DONE**

- Implementation code (if you have implemented your project)**DONE**

- Non-recorded (no voice) presentation slides**DONE**

Still, one member of your team should also submit the required project deliverable 2

materials to eLearning.

**Jonathan Hocevar**

**About Presentation of your Project:**

Regardless of which option you choose, presentation should take maximum **15 (fifteen)**

**minutes**. No minimum time restriction. As we have many groups and limited time for

presentations, we will have to time each presentation. So, thank you for your attention on

the matter. **Each team member is expected to talk during presentation.**

The presentation dates are:

11/12

11/17

11/19

11/24

The presentation schedule will be posted to eLearning once students submit their final

deliverables.

You may use any style in your presentations. A slide show is recommended as it helps

with displaying a summary of content you talk about to the audience as well as to

yourself. A suggested outline for presentations is listed below:

A brief introduction to your project topic

List of requirements

Use case diagram that contains use cases

Sequence diagram

Design Class Diagram (DCD)

User Interface Design

Comparison with similar work (if any), or emphasizing its significance and uniqueness (if

there is not any)

Conclusion and Future Work

You are welcome to enhance the minimum content listed above, provided that you stay

within maximum 15 minute presentation time requirement. We will listen to multiple

presentations per day, so please try not to exceed the allocated time for your presentations

so as not to steal from other groups’ presentation time.

Feel free to enrich your presentation with supporting figures, charts, documents, tables,

similar work, etc.

Contribute from yourselves: Employ your own design layouts, color selections,

animations, artistic perspectives to your presentations. Try to make them attractive. Think

of commercials: We only remember the “interesting” ones.

It is a suggested tactic that in a presentation, each slide should remain min. 1 minute on

display so that everybody reads and understands it. So, not too many slides maybe a good

idea to start with.

Rehearsal will prevent unexpected surprises. Make sure you rehearse and time your

presentation before you actually present it.

**Project Submission:**

Each group should designate one team member to submit his work via eLearning only.

PLEASE SUBMIT ONE PROJECT PER TEAM. Each member of the group will receive

the same grade, unless group members report a poor member performance.

**What to submit? DONE**

Please zip the following as one single file:

- Final project deliverable2 report (Please note that your deliverable2 report should also

include your deliverable1 report as required in section 2 above.) **DONE**

- Test code (section 5 above) **DONE**

- Non-recorded (no voice) presentation slides (section 9 above) **DONE**

- [Optional] If you have fully implemented your project, include your implementation

code (section 10 above) **NOT IMPLEMENTED**