

CS3354 Software Engineering

Final Project Deliverable 2

Book Review Library

Riya Patel
Benjamin Goff
Austin McGuire
James Castanho
Josephine Sun
Raashi Kulshrestha

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 the beginning of the project until the end. Please make sure to fairly distribute tasks in the team and remember that at the end of the semester, each member of a team will receive the same grade. See grading policy below for more detail.

Riya Patel

- Deliverable 1
 - i. Create a GitHub account.
 - ii. Use-Case diagram
 - iii. Register Sequence diagram
 - iv. Login Sequence diagram
 - v. First Commit to Repository
- Deliverable 2
 - i. Comparison of your work with similar designs.
 - ii. Presentation slides.

Benjamin Goff

- Deliverable 1
 - i. Create a GitHub account.
 - ii. Project Scope
 - iii. View List sequence diagram
 - iv. Create List sequence diagram
 - v. Edit List sequence diagram
 - vi. Delete List sequence diagram
- Deliverable 2
 - i. Project scheduling Graph
 - ii. Project Scheduling Table
 - iii. Presentation slides.

Austin McGuire

- Deliverable 1
 - i. Create a GitHub account.
 - ii. First commit to the repository.
 - iii. Non-functional requirements.
 - iv. Class diagram.
 - v. Sequence Diagram
- Deliverable 2
 - i. Project scheduling Graph

- ii. Project Scheduling Table
- iii. Presentation slides.

James Castanho

- Deliverable 1
 - i. Create a GitHub account.
 - ii. Non-functional requirements.
 - iii. Class diagram.
- Deliverable 2
 - i. Test Plan
 - ii. Presentation slides.

Josephine Sun

- Deliverable 1
 - i. Create a GitHub account.
 - ii. Create Github Repository & add all members and TA as collaborators
 - iii. Architectural Design
 - iv. Software Process Model
 - v. Search For Book Sequence Diagram
 - vi. Filter Book Sequence Diagram
 - vii. View Book Information Sequence Diagram
- Deliverable 2
 - i. Cost, effort and pricing estimation.
 - ii. Estimated cost of hardware products.
 - iii. Estimated cost of software products.
 - iv. Estimated cost of personnel.
 - v. User Interface Design Images
 - vi. Presentation slides.

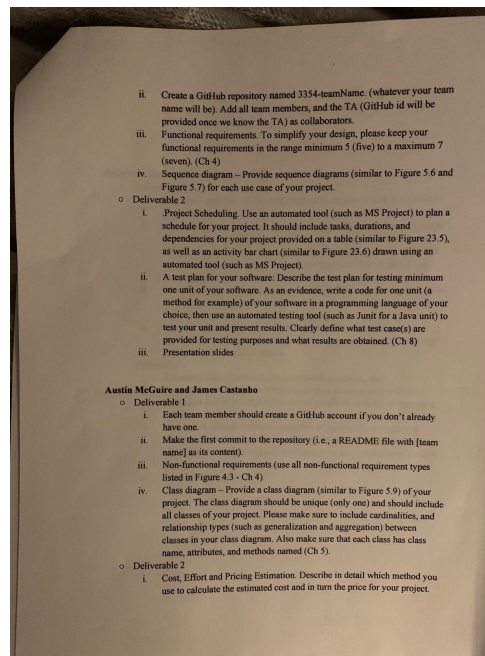
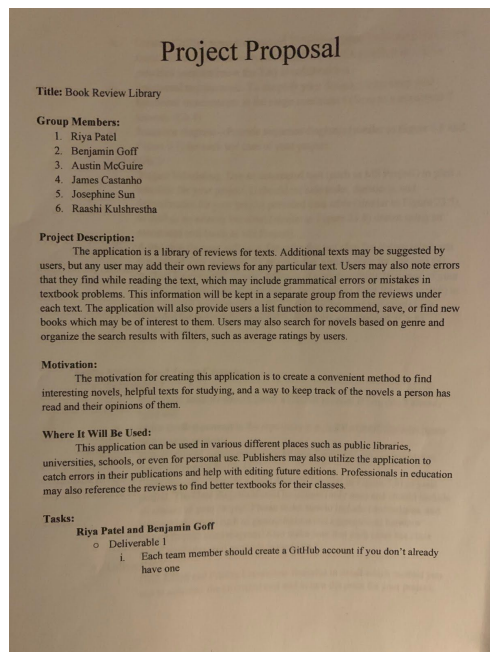
Raashi Kulshrestha

- Deliverable 1
 - i. Create a GitHub account.
 - ii. View Sequence Diagram
 - iii. Create Sequence Diagram
 - iv. Edit Delete Sequence Diagram
- Deliverable 2
 - i. Conclusion.
 - ii. Presentation slides.

2. [5 POINTS] Everything required and already submitted in Final Project Deliverable 1. Please specify this part as “Project Deliverable 1 content”.

Project Deliverable 1 content

1. Please attach here the Final Project draft description (that contains the instructor feedback). It is ok to include a picture of the original document. Address the feedback provided for your proposal by listing what you did to comply with the proposed changes/requests for additions to your project.



- ii. 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.
- iii. Presentation slides

Josephine Sun and Raashi Kulshrestha

- o Deliverable 1
 - i. Each team member should create a GitHub account if you don't already have one.
 - ii. Make another commit including a pdf/txt/doc file named "project_scope". If you choose a predefined topic (one of the 4 topics described in the "Project Topic Ideas" section of this document), the contents of the file should be identical to the corresponding project in this section. If you choose other topics, the contents should follow a similar structure.
 - iii. Use case diagram - Provide a use case diagram (similar to Figure 5.5) for your project. Please note that there should be more than one use cases depending on the complexity of your project. (Ch 5 and Ch 7)
 - iv. Architectural design - Provide an architectural design of your project. Based on the characteristics of your project, choose and apply only one appropriate architectural pattern
- o Deliverable 2
 - i. Estimated cost of hardware products (such as servers, etc.)
Estimated cost of software products (such as licensed software, etc.)
Estimated cost of personnel (number of people to code the end product, training cost after installation)
 - ii. 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.
 - iii. Presentation slides

1. Setting up a Github repository:

<https://github.com/JosephineSun4070/3354-BookReviewLibrary>

2. Delegation of tasks: Who is doing what

Riya Patel

- Deliverable 1
 - i. Create a GitHub account.
 - ii. Use-Case diagram
 - iii. Register Sequence diagram
 - iv. Login Sequence diagram
 - v. First Commit to Repository
- Deliverable 2
 - i. Project scheduling.
 - ii. Test plan.
 - iii. Presentation slides.

Benjamin Goff

- Deliverable 1
 - i. Create a GitHub account.
 - ii. Project Scope
 - iii. View List sequence diagram
 - iv. Create List sequence diagram
 - v. Edit List sequence diagram
 - vi. Delete List sequence diagram
- Deliverable 2
 - i. Project scheduling.
 - ii. Test plan.
 - iii. Presentation slides.

Austin McGuire

- Deliverable 1
 - i. Create a GitHub account.
 - ii. First commit to the repository.
 - iii. Non-functional requirements.
 - iv. Class diagram.
- Deliverable 2
 - i. Cost, effort and pricing estimation.
 - ii. Comparison of your work with similar designs.

- iii. Presentation slides.

James Castanho

- Deliverable 1
 - i. Create a GitHub account.
 - ii. Non-functional requirements.
 - iii. Class diagram.
- Deliverable 2
 - i. Cost, effort and pricing estimation.
 - ii. Comparison of your work with similar designs.
 - iii. Presentation slides.

Josephine Sun

- Deliverable 1
 - i. Create a GitHub account.
 - ii. Create Github Repository & add all members and TA as collaborators
 - iii. Architectural Design
 - iv. Software Process Model
 - v. Search For Book Sequence Diagram
 - vi. Filter Book Sequence Diagram
 - vii. View Book Information Sequence Diagram
- Deliverable 2
 - i. Estimated cost of hardware products.
 - ii. Estimated cost of software products.
 - iii. Estimated cost of personnel.
 - iv. Comparison of your work with similar designs.
 - v. Presentation slides.

Raashi Kulshrestha

- Deliverable 1
 - i. Create a GitHub account.
 - ii. View Sequence Diagram
 - iii. Create Sequence Diagram
 - iv. Edit Delete Sequence Diagram
- Deliverable 2
 - i. Cost, effort and pricing estimation
 - ii. Conclusion.
 - iii. Presentation slides.

3. Which software process model is employed in the project and why.

The software process model that we are employing in this project is the Incremental Process Model. We are using this model because we can create a delivery of the first increment, and then add more additional features to the project over time. This process model is a desirable choice for us to use, because our project can be used by a large and diverse audience. Thus it is a good idea to select a model that would give us the ability to add and change the software after the first delivery. With this model, we can also improve the features that were developed in older version, and optimize them for the more recent scenarios in which our system is being used for.

4.a.) Functional Requirements

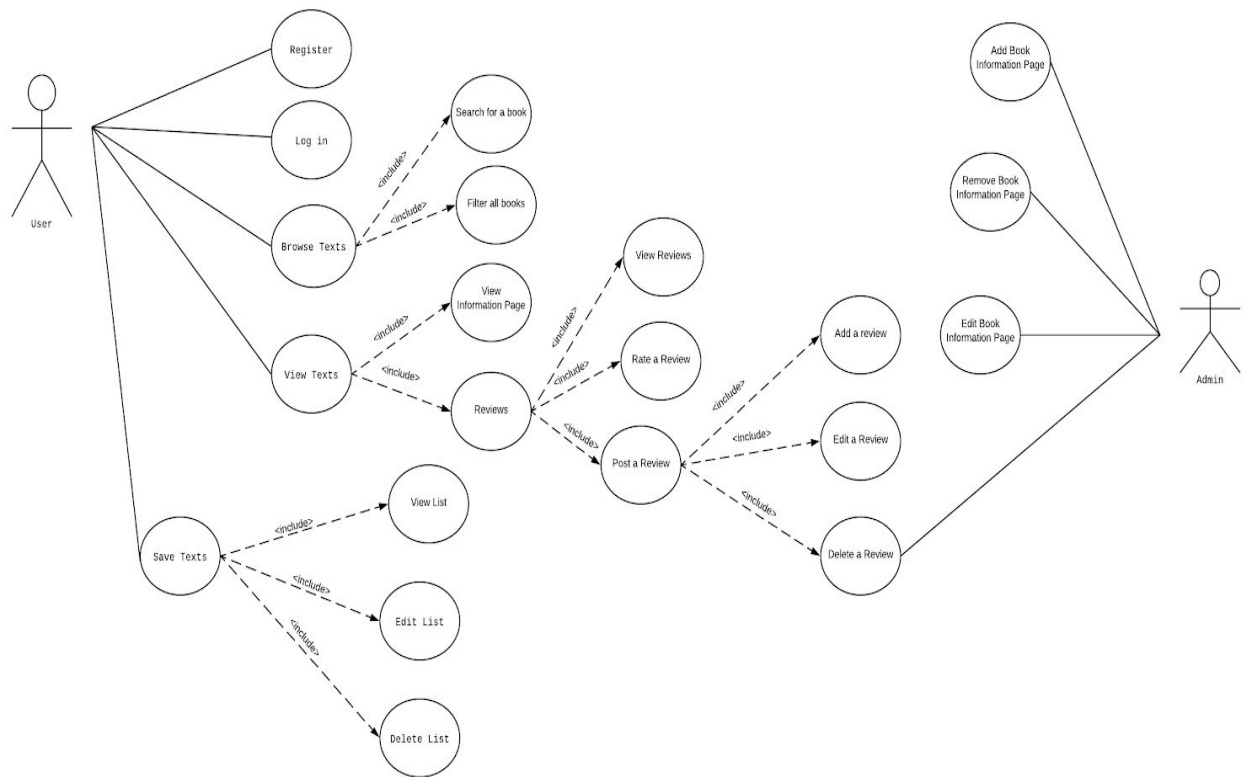
1. User shall be able to search through book database for a specific book by either Title, Author name, Series name, or ISBN number of the book.
2. User shall be able to view all published reviews about chosen book (with reviews with the most likes at the top of the list).
3. User shall be able to type a review and add it to the list of reviews.
4. User shall be able to "like" certain reviews.
5. Each user using the system shall be uniquely identified by a chosen username.
6. Each user shall access the system by entering their user name and chosen password.
7. Each user should be able to search for other users by username, and view all reviews published by that user.

4.b.) Non-Functional Requirements

- Product Requirements
 - Usability
 - Website shall be available for Android and iOS mobile devices.
 - Website shall be available in the following languages:
 - Mandarin
 - Spanish
 - English
 - Hindi
 - Bengali
 - Portuguese
 - Russian
 - Japanese
 - French
 - German
 - Website shall be accessible to users with poor eyesight or colorblindness.
 - No tutorial should be required prior to use, but will be available by request.
 - Efficiency
 - Performance
 - Site must load in less than 2 seconds.
 - Search results must return within 2 seconds of request.
 - Website should not crash or have any bugs that would reduce functionality.
 - Service should be available to 10,000 concurrent users within a 5 second timeframe.
 - Space
 - File sizes should not exceed 30 MB.
 - Dependability
 - Downtime/Maintenance should not last more than 3 hours per week.
 - Downtime/Maintenance should be done during off-peak hours (~3AM)
 - Database should be backed up during downtime/maintenance.
 - Security
 - End users' data shall be encrypted.
 - User data is confidential, will not be sold to 3rd parties.
- Organizational Requirements
 - Environmental
 - All communication with stakeholders will be through paperless communications.

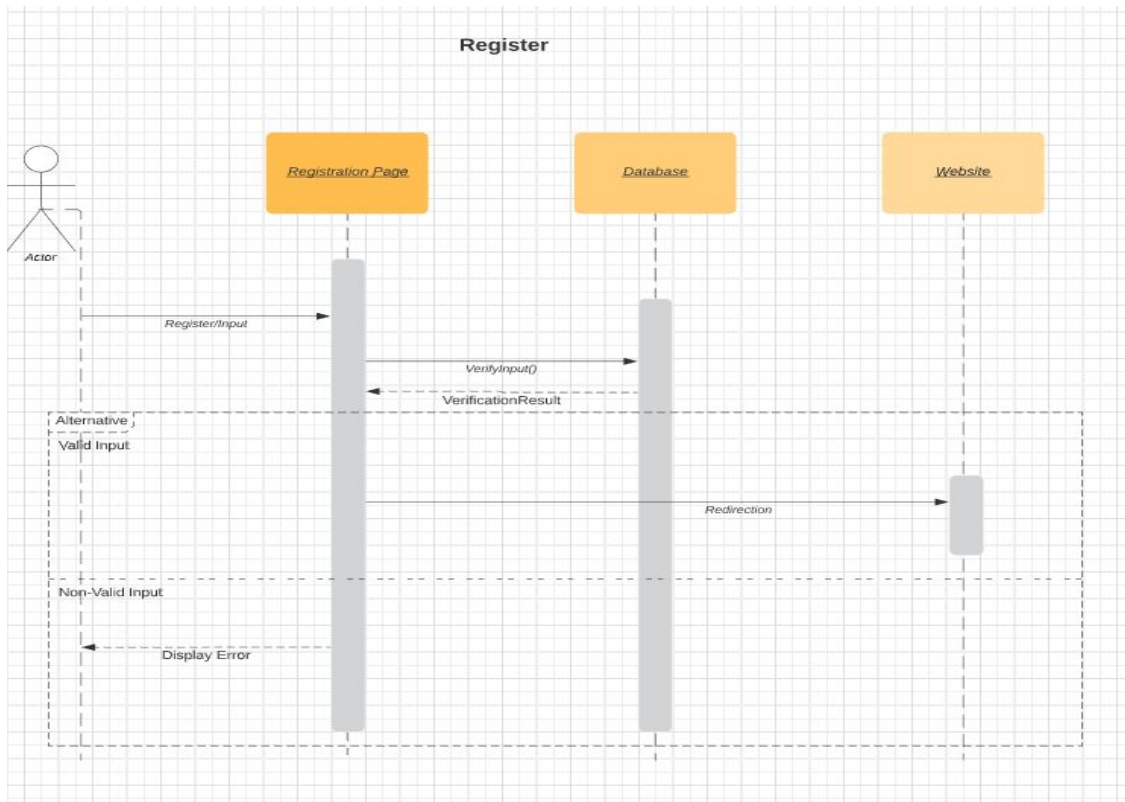
- Operational
 - End users must log in to the Website via unique username and password
 - Each End User may only create one review for each book.
 - End Users may like or dislike other User reviews only once per review.
- Development
 - Code will be written in the Java programming language
 - Website will be designed according to the MVC software architecture pattern
- External Requirements
 - Regulatory
 - Website will abide by all laws and regulatory requirements where its services will be available.
 - Ethical
 - Book reviews will be moderated to ensure legitimacy/appropriateness of user reviews.
 - Legislative
 - Safety
 - Website shall not have any vulnerabilities which would compromise the end user's mobile device or computer.

5. Use case diagram:

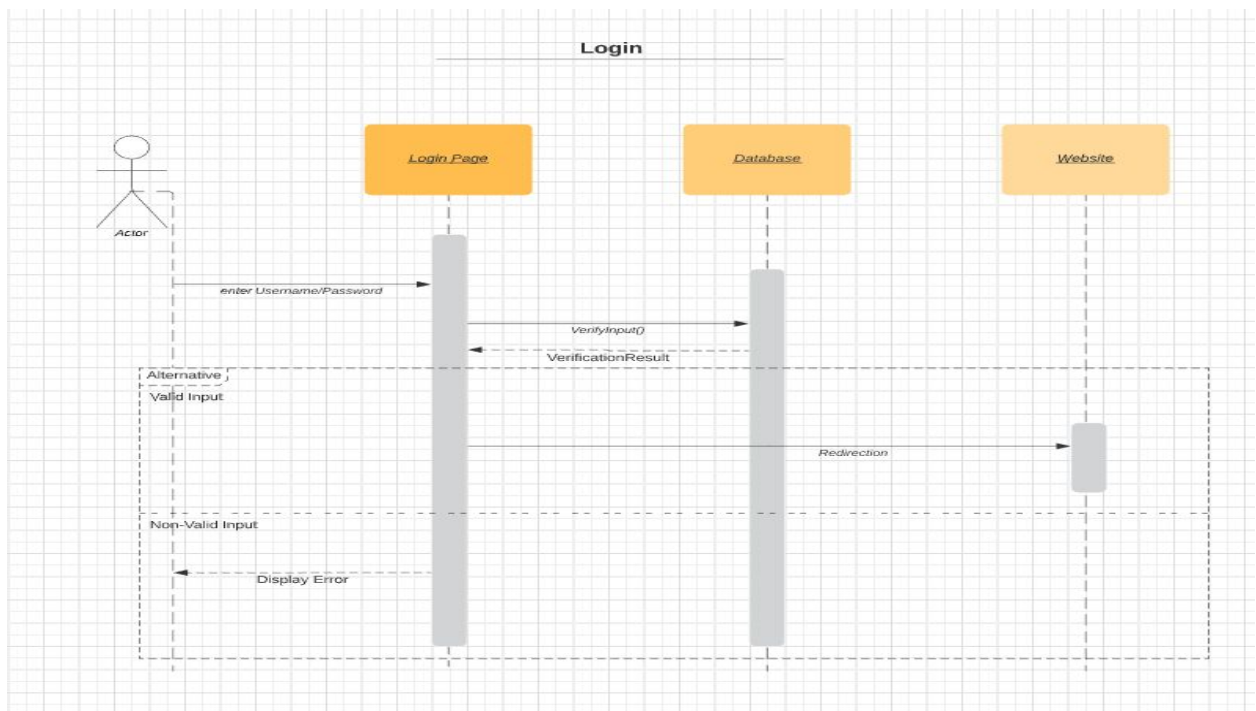


6. Sequence diagram:

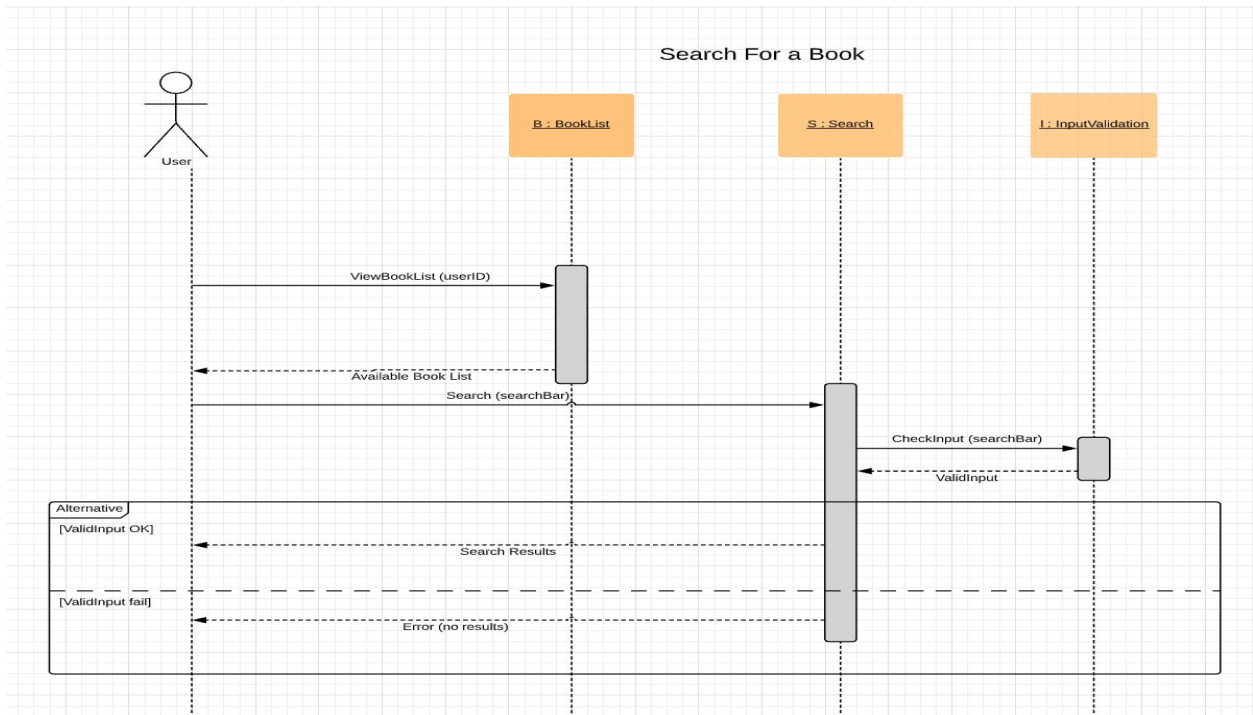
1. Register



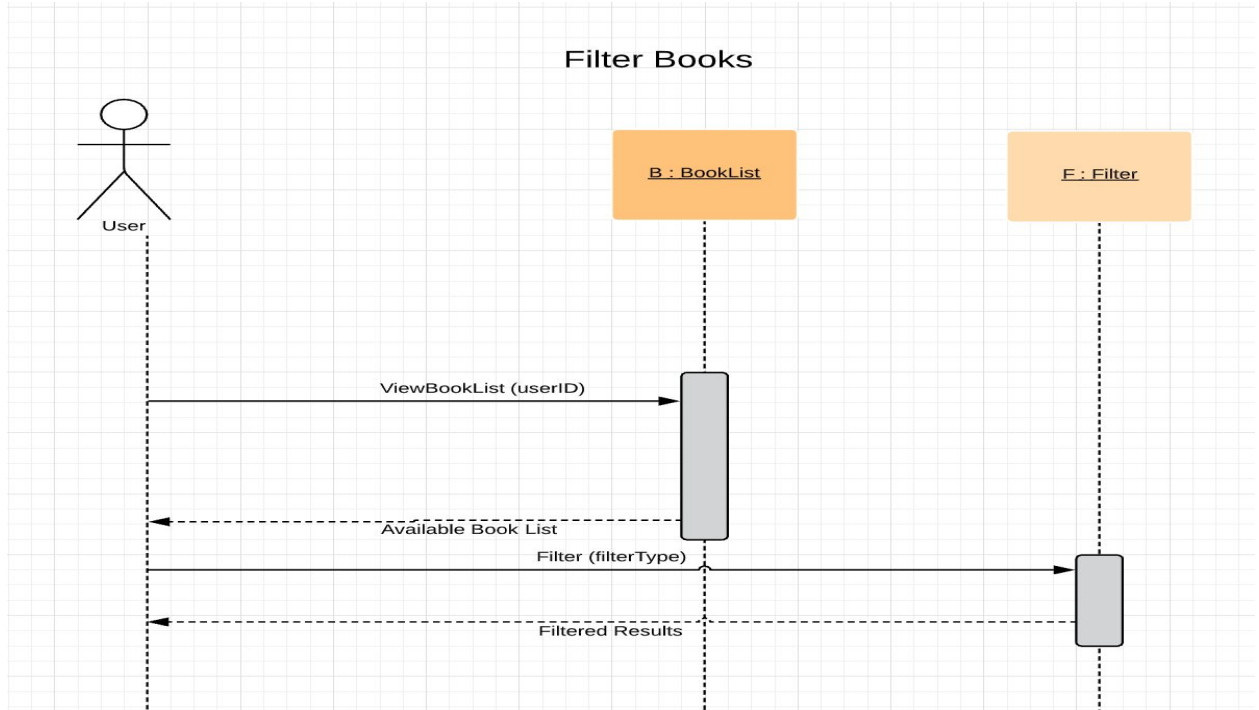
2. Login



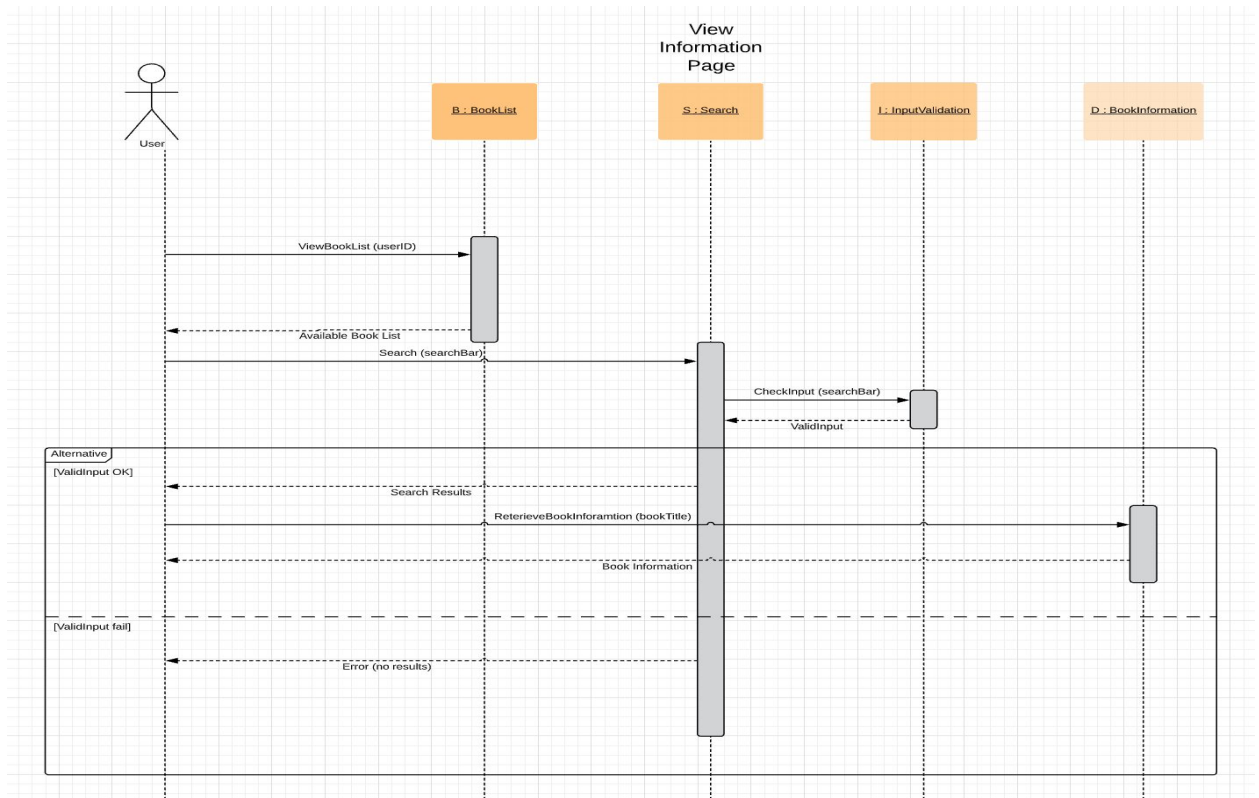
3. Search for book



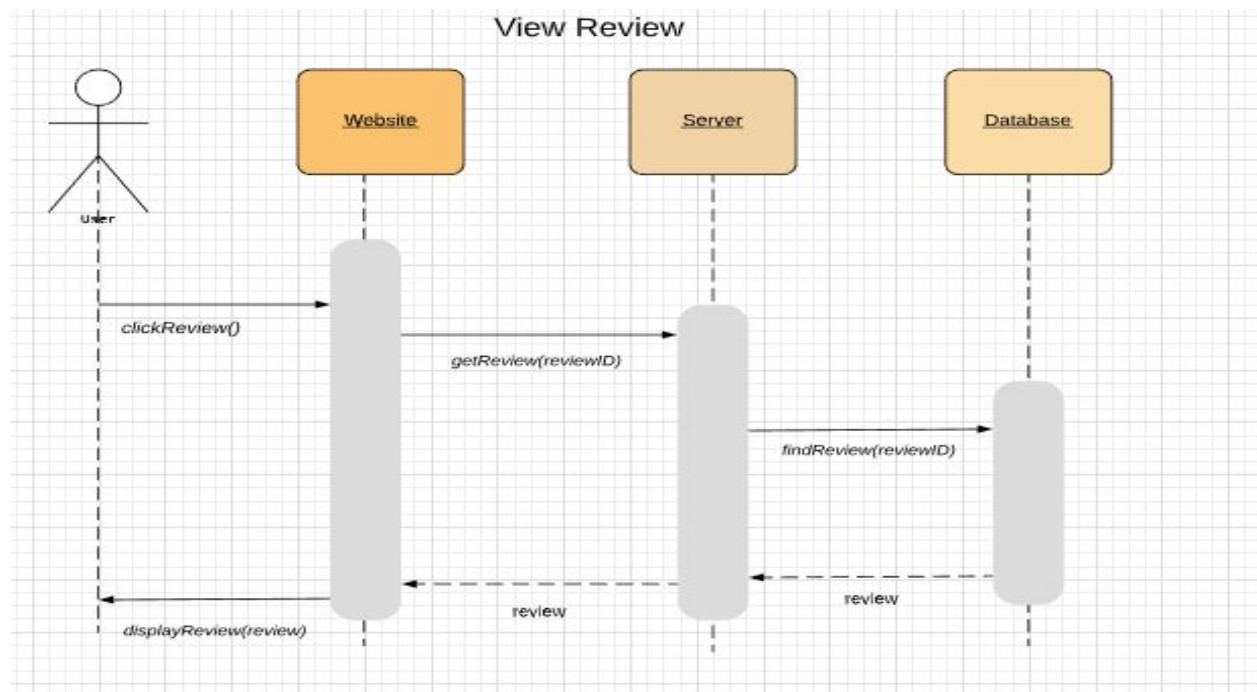
4. Filter books



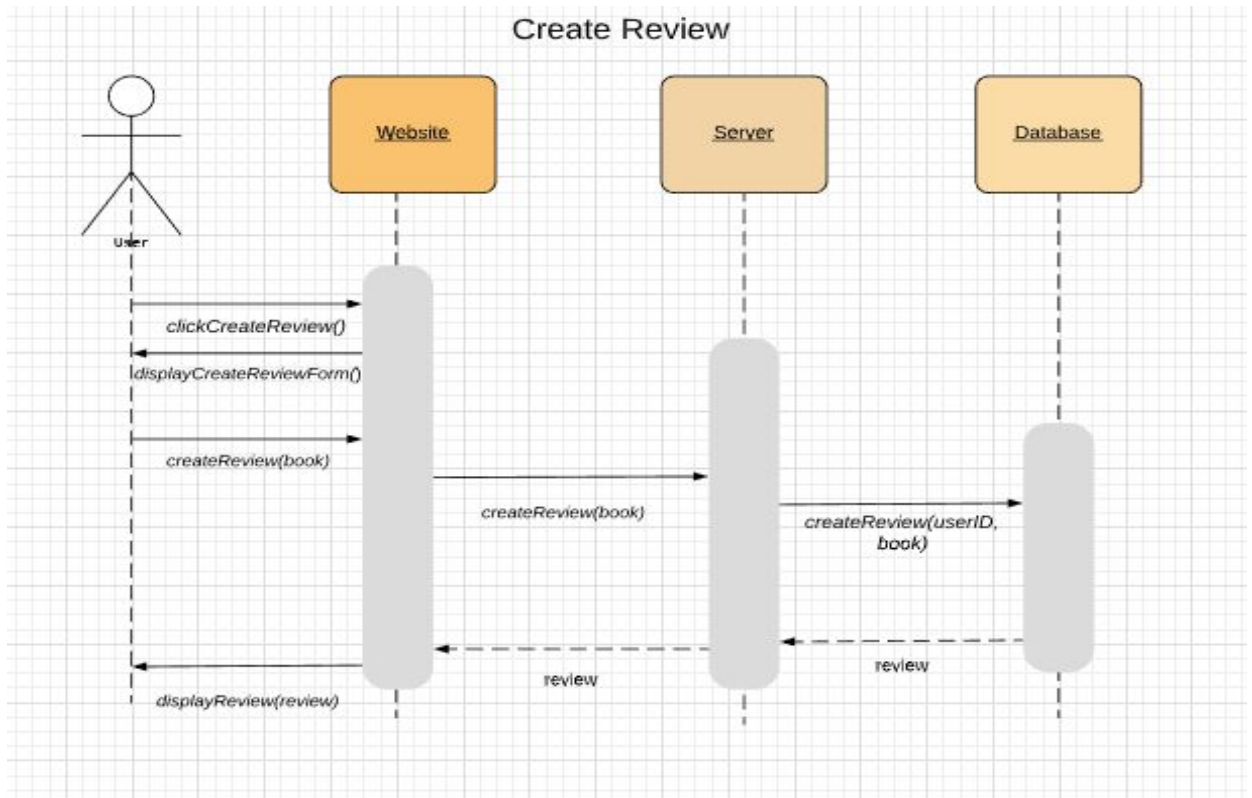
5. View info page



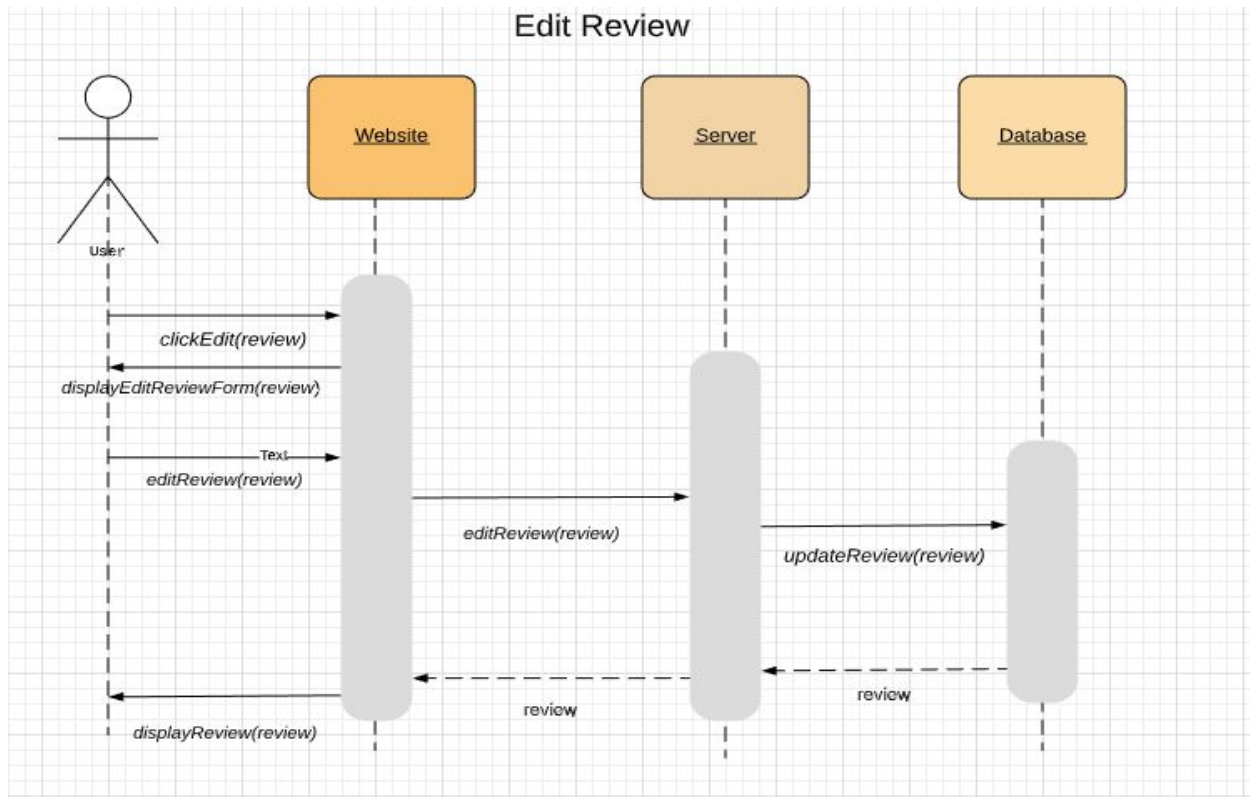
6. View reviews



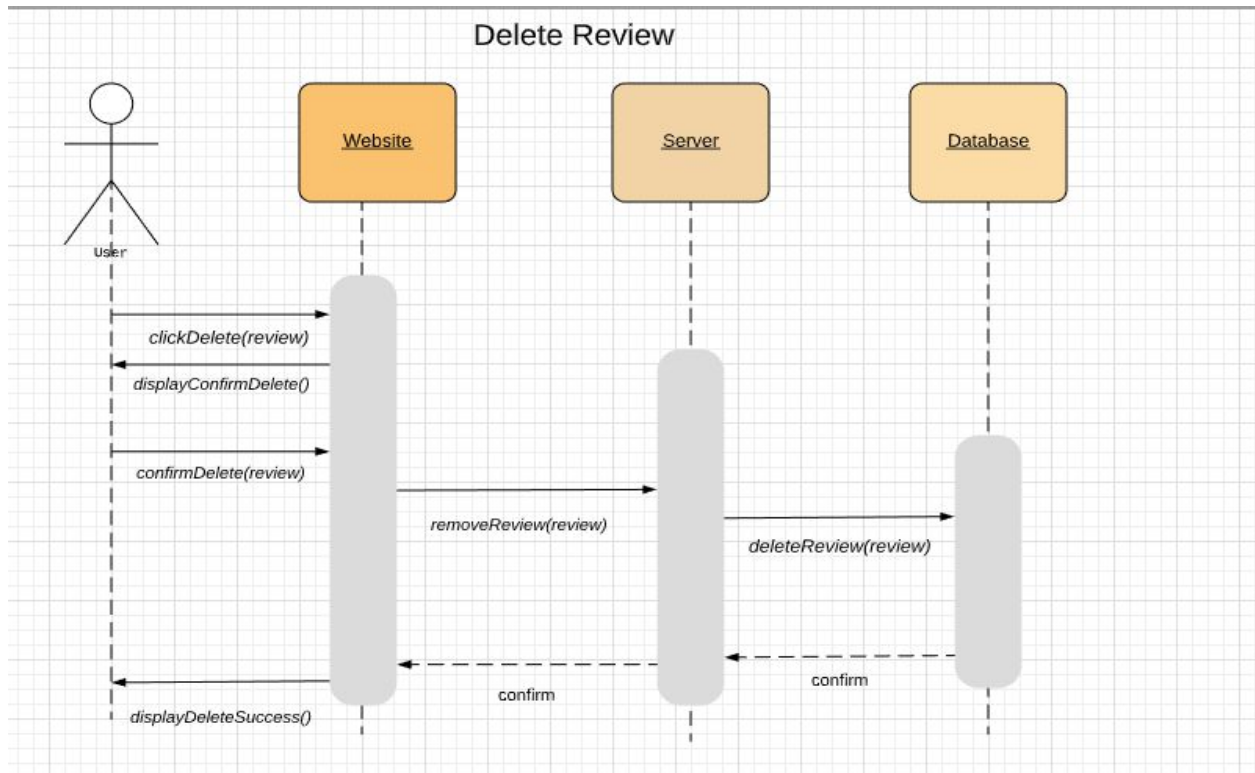
7. Create reviews



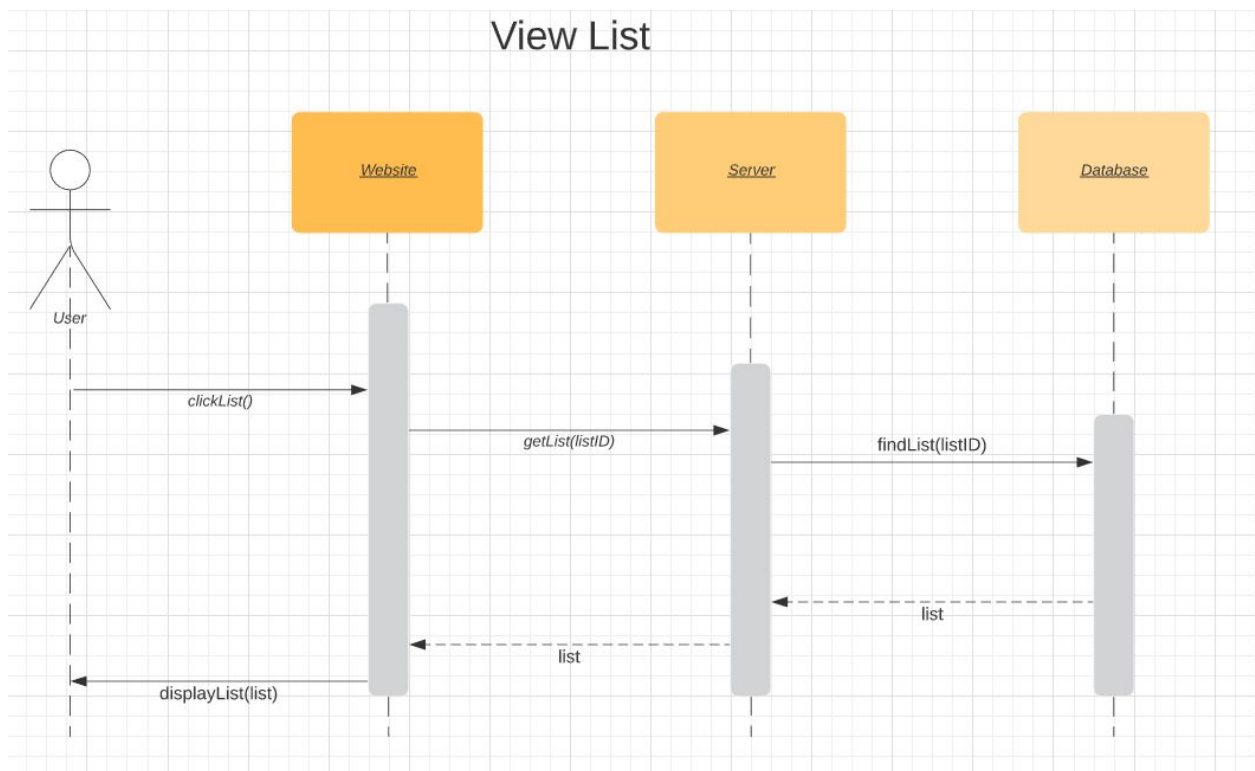
8. Edit reviews



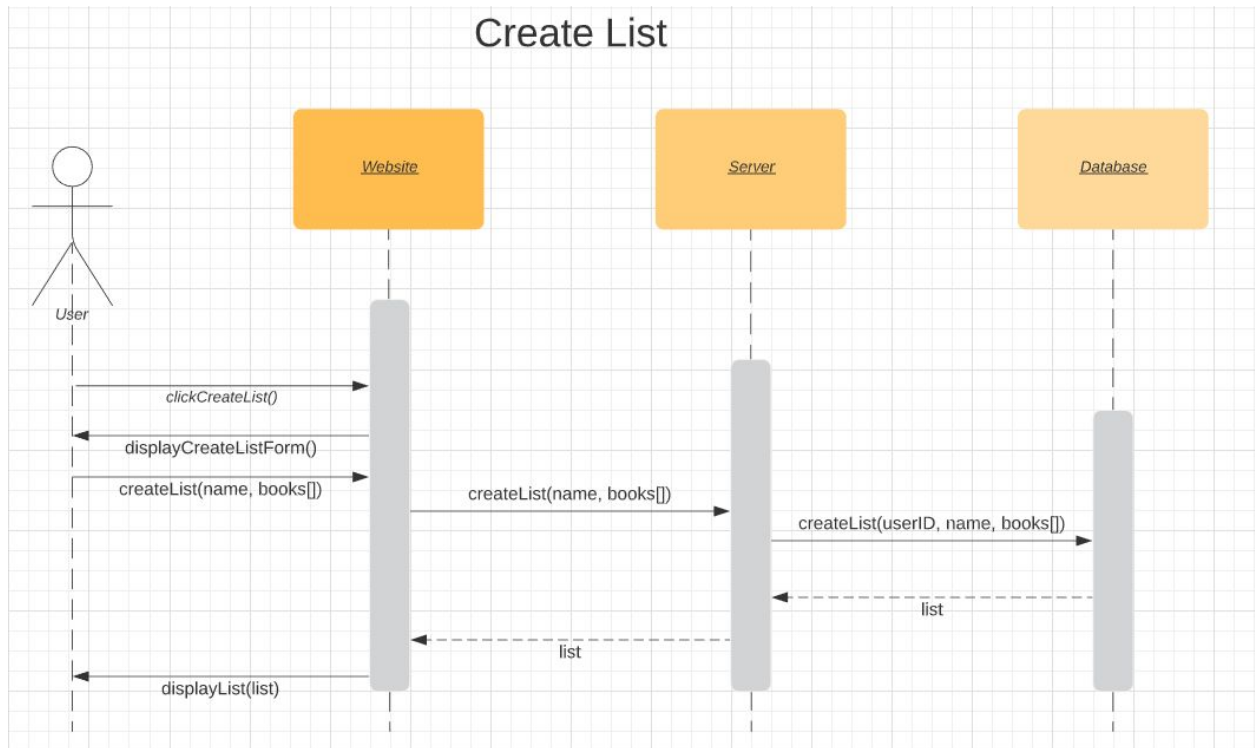
9. Delete reviews



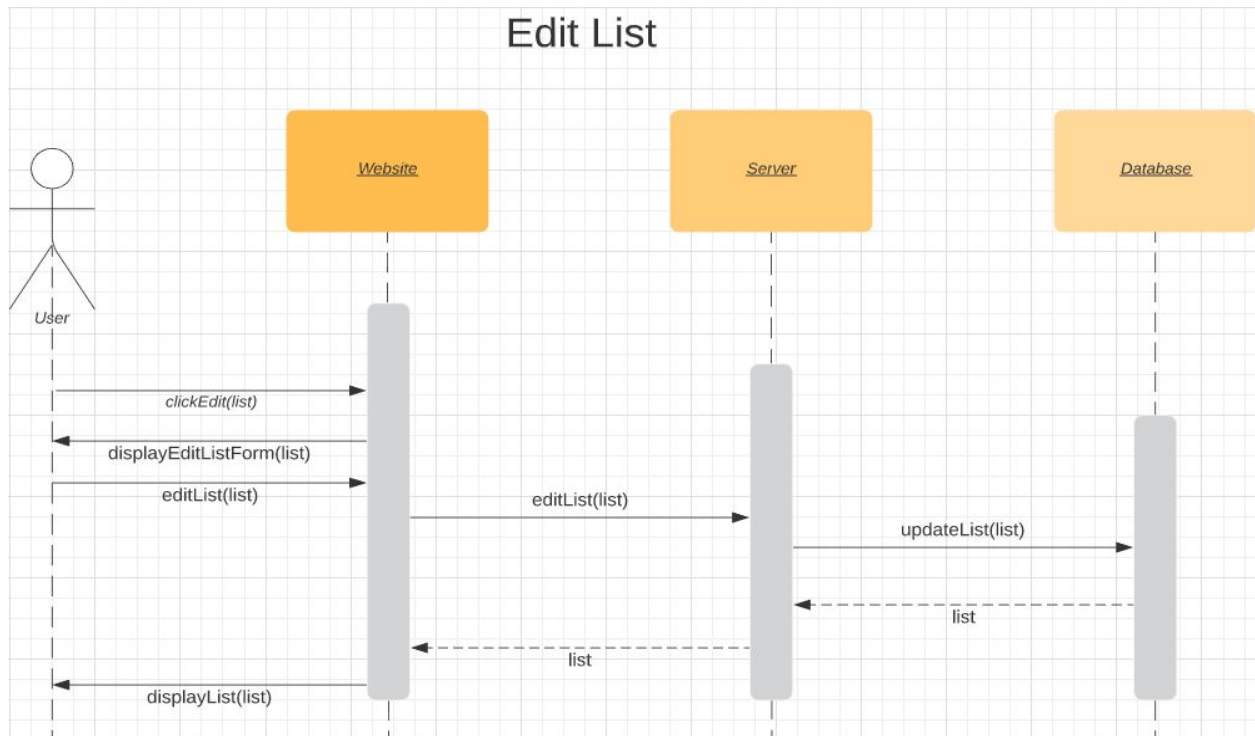
10. View List



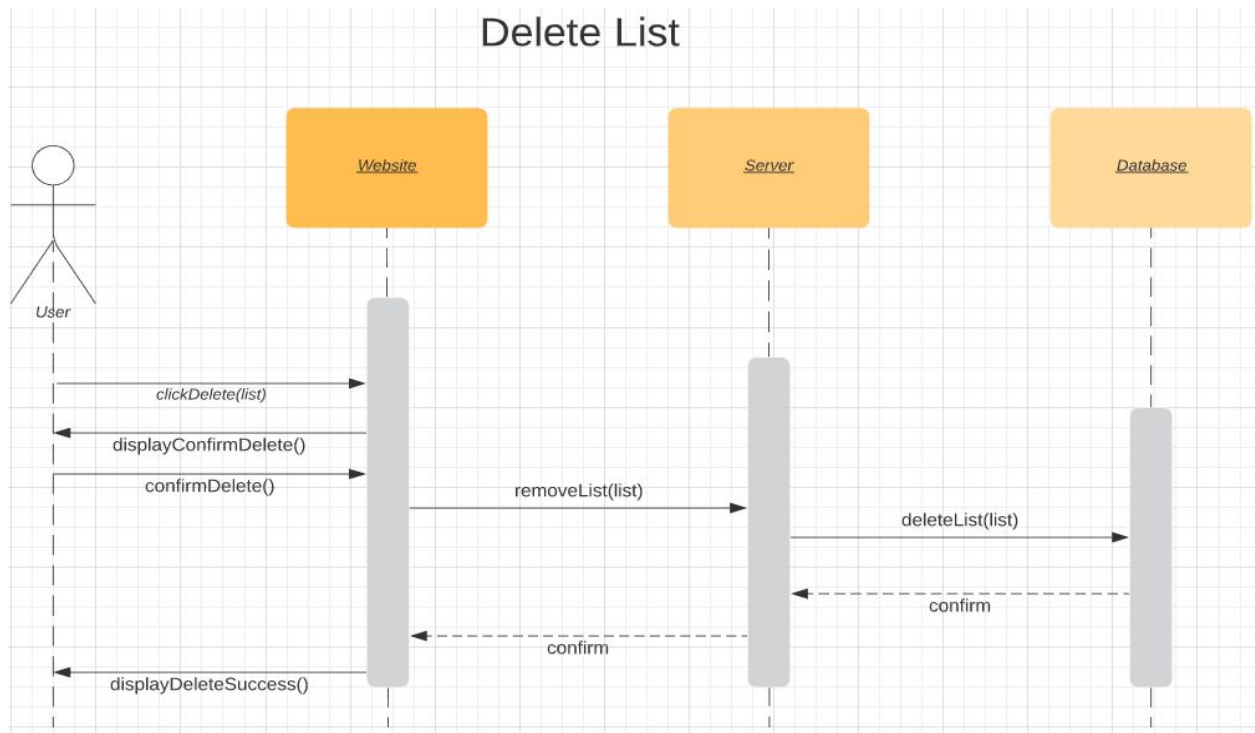
11. Create List



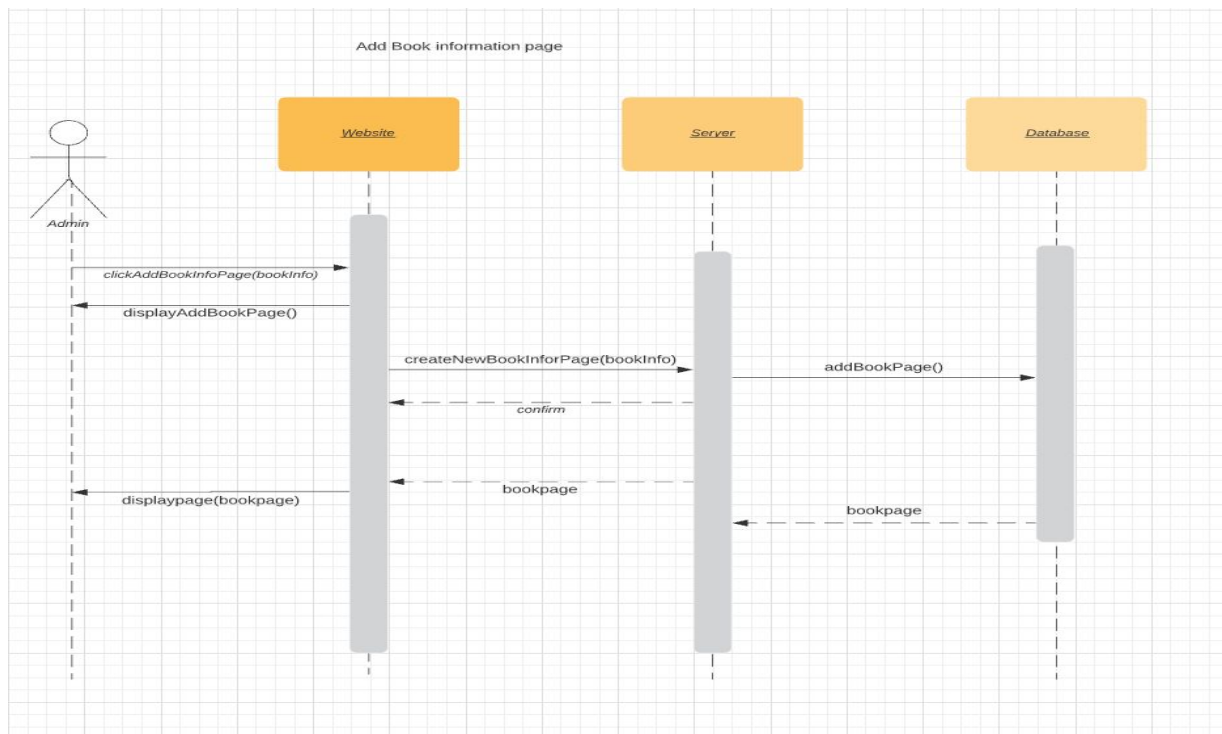
12. Edit List



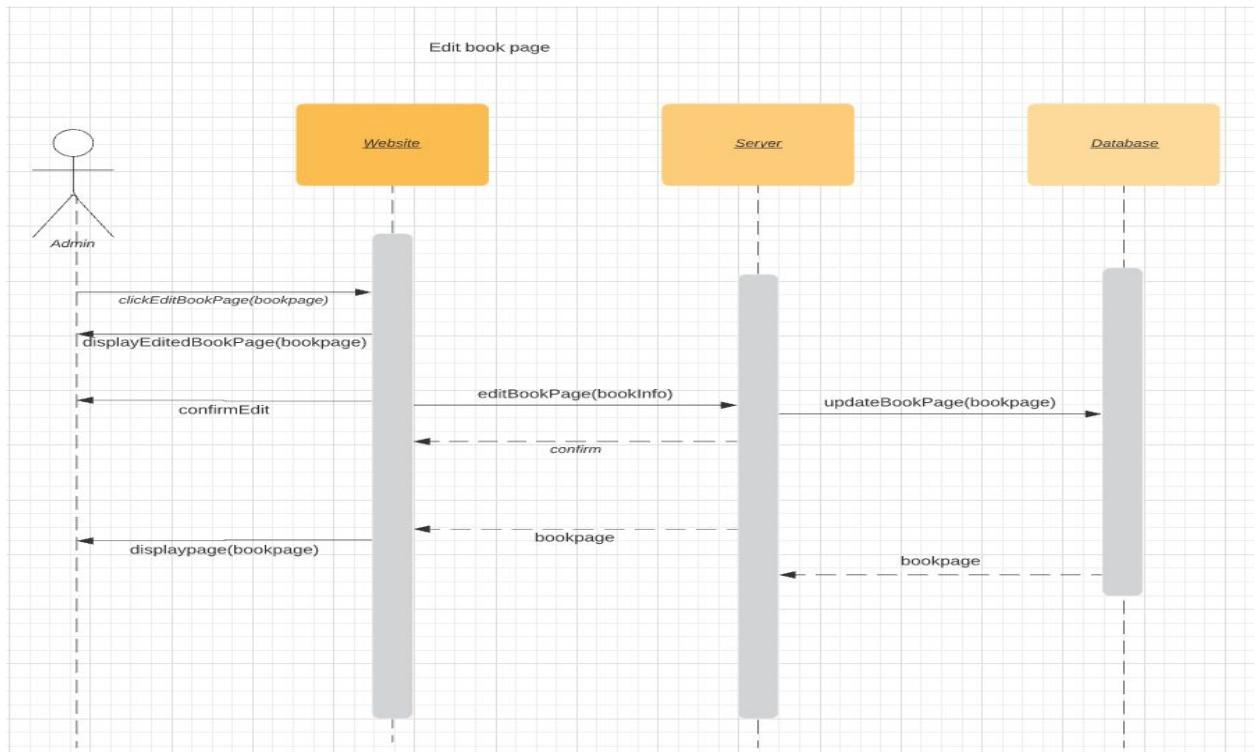
13. Delete List



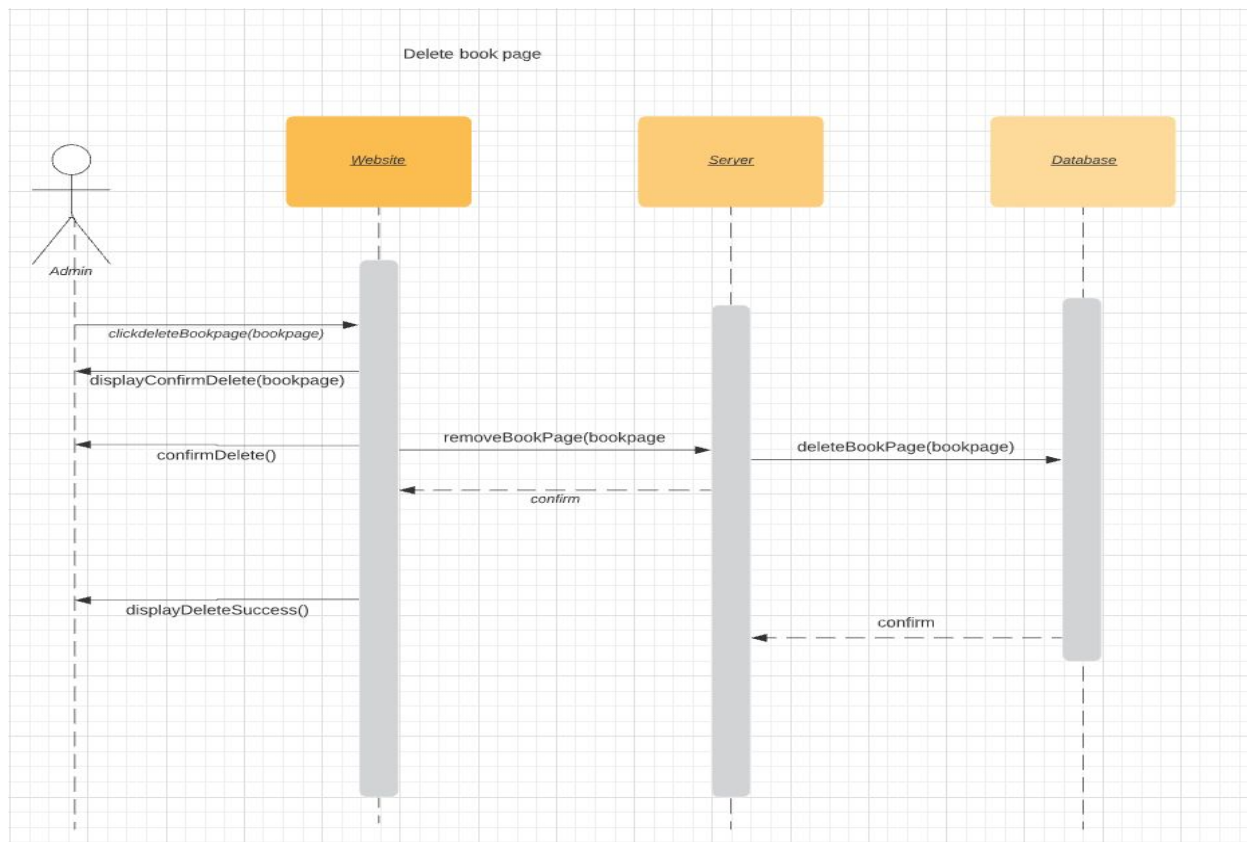
14. Add Books/Admin



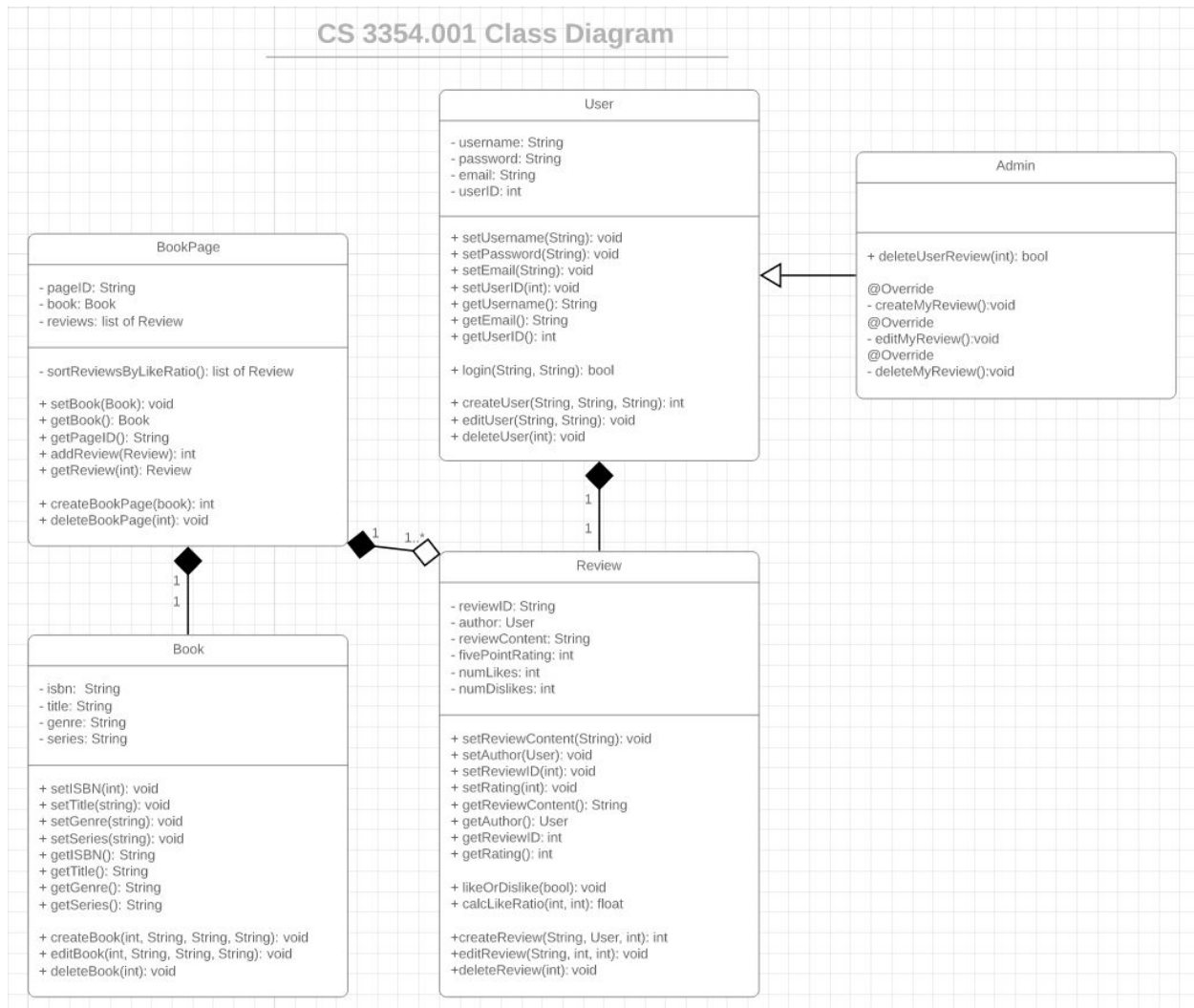
15. Edit Books/Admin



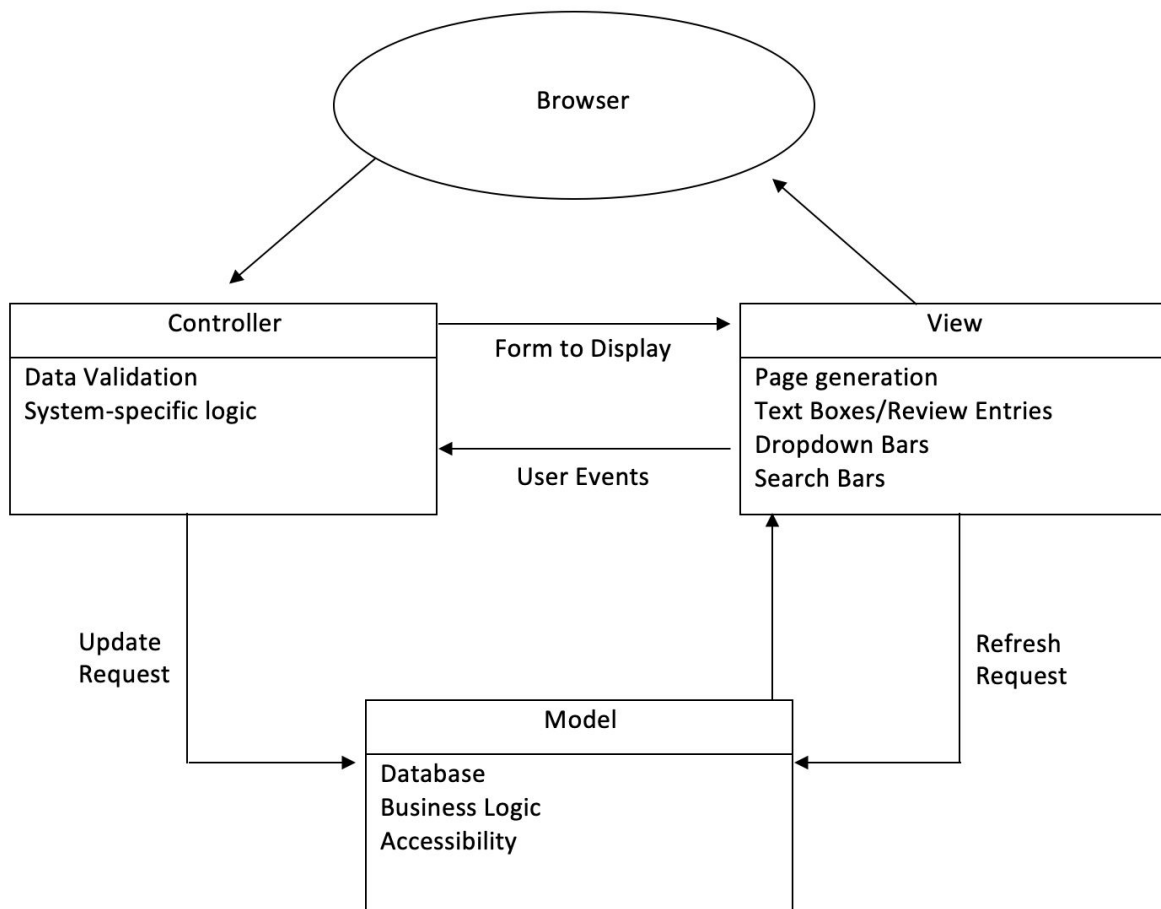
16. Remove Book/Admin



7. Class diagram:



8. **Architectural design** - Model-View-Controller (MVC) pattern (similar to Figure 6.6)

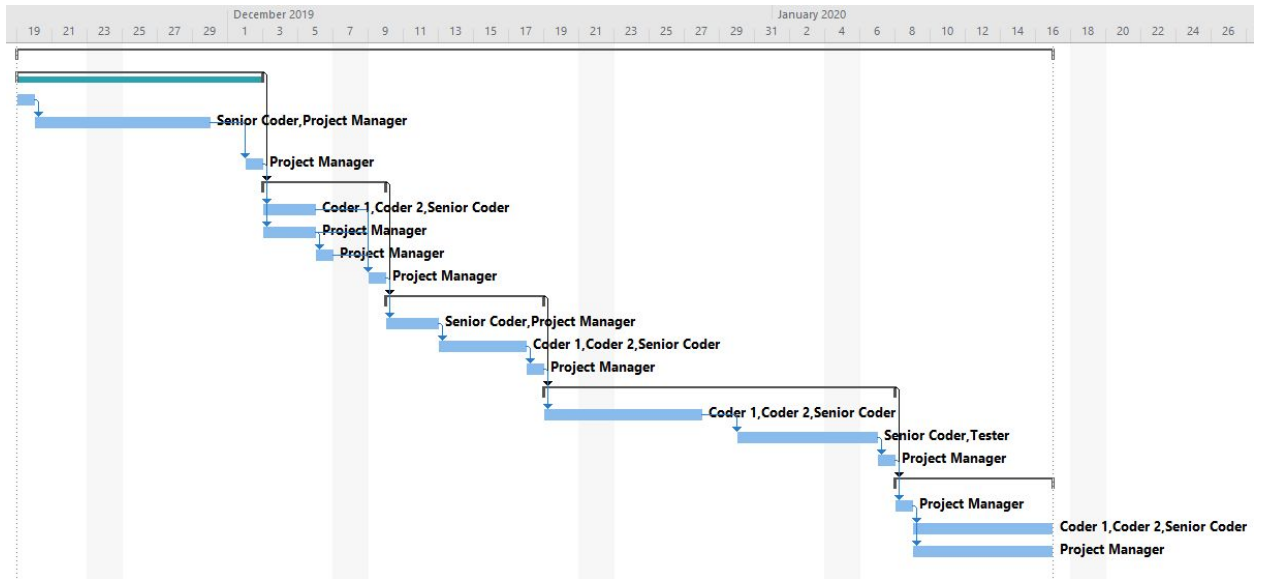


**Project Deliverable 1 content
end**

3. [50 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:

3.1. [20 POINTS] Project Scheduling. Please note that what you present should be the timeline of the project designed, NOT the time you've spent on it. Use an automated tool (such as MS Project) to plan a schedule for your project. It should include tasks, durations, and dependencies for your project provided on a table (similar to Figure 23.5), as well as an activity bar chart (similar to Figure 23.6) drawn using an automated tool (such as MS Project). Remember that MS Project is installed and ready for use in UTD Open Lab computers. Or, you may install the freely available version of this tool to your computers. Please include your MS Project file as an additional document to your project report submission. You may zip all your files in one single zip file afterwards.

Task Name	Duration	Start	Finish	Predecessors	Resource Names	Cost
Group All	43 days?	Tue 11/19/19	Thu 1/16/20			\$34,673.92
Communication	10 days	Tue 11/19/19	Mon 12/2/19			\$4,712.96
Project Initiation	1 day	Tue 11/19/19	Tue 11/19/19			\$0.00
Requirements Gathering	8 days	Wed 11/20/19	Fri 11/29/19	3	Senior Coder, Project Manager	\$4,376.32
Validation	1 day	Mon 12/2/19	Mon 12/2/19	4	Project Manager	\$336.64
Planning	5 days	Tue 12/3/19	Mon 12/9/19	2		\$4,712.96
Estimating	3 days	Tue 12/3/19	Thu 12/5/19	5	Coder 1, Coder 2, Senior Coder	\$3,029.76
Scheduling	3 days	Tue 12/3/19	Thu 12/5/19	5	Project Manager	\$1,009.92
Tracking	1 day	Fri 12/6/19	Fri 12/6/19	8	Project Manager	\$336.64
Validation	1 day	Mon 12/9/19	Mon 12/9/19	7,8,9	Project Manager	\$336.64
Modeling	7 days?	Tue 12/10/19	Wed 12/18/19	6		\$5,386.24
Analysis	3 days	Tue 12/10/19	Thu 12/12/19	10	Senior Coder, Project Manager	\$2,019.84
Design	3 days	Fri 12/13/19	Tue 12/17/19	12	Coder 1, Coder 2, Senior Coder	\$3,029.76
Validation	1 day?	Wed 12/18/19	Wed 12/18/19	13	Project Manager	\$336.64
Construction	14 days?	Thu 12/19/19	Tue 1/7/20	11		\$11,445.76
Code	7 days	Thu 12/19/19	Fri 12/27/19	14	Coder 1, Coder 2, Senior Coder	\$7,069.44
Test	6 days	Mon 12/30/19	Mon 1/6/20	16	Senior Coder, Tester	\$4,039.68
Validation	1 day?	Tue 1/7/20	Tue 1/7/20	17	Project Manager	\$336.64
Deployment	7 days?	Wed 1/8/20	Thu 1/16/20	15		\$8,416.00
Delivery	1 day?	Wed 1/8/20	Wed 1/8/20	18	Project Manager	\$336.64
Support	6 days	Thu 1/9/20	Thu 1/16/20	20	Coder 1, Coder 2, Senior Coder	\$6,059.52
Feedback	6 days	Thu 1/9/20	Thu 1/16/20	20	Project Manager	\$2,019.84



3.2. [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. Some cost modeling techniques you may use are listed as follows:

1. Function Point

Or any of the following COCOMO II estimation models

2. Application composition

3. Early design

4. Post-architecture

Algorithmic Estimation technique: Function Point (FP) Method

Step 1. Determine Function Category Count

Step 2. Determine Complexity

Step 3. Compute Gross Function Point (GFP)

	Function Category	Count	Complexity			Count x Complexity
			Simple	Average	Complex	
1	Number of User Input	13	3	4	6	52
2	Number of User Output	7	4	5	7	49
3	Number of User Queries	10	3	4	6	40
4	Number of Data Files and Relational Tables	5	7	10	15	75
5	Number of External Interfaces	0	5	7	10	0
					GFP	216

Step 4. Determine Processing Complexity (PC)

PC = 0 (no influence), 1 (incidental), 2 (moderate), 3 (average), 4 (significant), 5 (essential)

(1) Does the system require reliable backup and recovery? **PC = 3**

(2) Are data communications required? **PC = 2**

(3) Are there distributed processing functions? **PC = 0**

- (4) Is performance critical? **PC = 1**
- (5) Will the system run in an existing, heavily utilized operational environment? **PC = 2**
- (6) Does the system require online data entry? **PC = 5**
- (7) Does the online data entry require the input transaction to be built over multiple screens or operations? **PC = 4**
- (8) Are the master files updated online? **PC = 1**
- (9) Are the inputs, outputs, files, or inquiries complex? **PC = 3**
- (10) Is the internal processing complex? **PC = 1**
- (11) Is the code designed to be reusable? **PC = 4**
- (12) Are conversion and installation included in the design? **PC = 1**
- (13) Is the system designed for multiple installations in different organizations? **PC = 0**
- (14) Is the application designed to facilitate change and ease of use by the user? **PC = 5**

Step 5. Compute Processing Complexity Adjustment (PCA).

$$\begin{aligned}
 PCA &= 0.65 + 0.01 (PC1 + PC2 + \dots + PCn) \text{ where } 0.65 \text{ is the empirical constant} \\
 PCA &= 0.65 + 0.01 (3 + 2 + 0 + 1 + 2 + 5 + 4 + 1 + 3 + 1 + 4 + 1 + 0 + 5) \\
 &= 0.65 + 0.01 (32) \\
 &= \mathbf{0.97}
 \end{aligned}$$

*Step 6. Compute Function Point (FP) using the formula $FP = GFP * PCA$*

$$GFP = 216$$

$$PCA = 0.97$$

$$FP = 216 * 0.97 = \mathbf{209.52}$$

The method we used to calculate the estimated cost, effort, and price for our project is the function point method. Assuming that the productivity of the development team is fifty function points per person-week, the estimated effort is obtained through the calculation,
 $Effort = FP / Productivity = 209.52 / 50 = 4.1904$

In conclusion, the estimated effort is approximately 4 person-weeks, so if the team size is 4, the project duration would be calculated such that,
 $Duration = Effort / Team Size = 4 / 4 = 1 \text{ week}$

The estimated cost of the project is derived below and we concluded that the estimated price for the software we are planning will be 3.5K annually. In order to compensate for this cost, we will rely on the use of add revenues to collect the required payment to maintain the website on its servers. The pricing for the software is ideally free to the general public.

3.3. [5 POINTS] Estimated cost of hardware products (such as servers, etc.)

A similar software to the one described in our project is the GoodReads Software. This software operated using the Nginx servers prior to their merger with amazon. [1] The pricing for this particular servers' is listed to begin at 2.5K per year. [2] Since our software is new, the amount of users would not be too large and can likely be accommodated by just one server. Although a backup server or two might be ideal in

case one of them is down, Nginx provides support at an increased price. Taking this option, the pricing for the server would be 3.5K per year for less critical deployments. [2]

3.4. [5 POINTS] Estimated cost of software products (such as licensed software, etc.)

Since we do not plan to utilize other software products, such as open source or licensed software, the estimated cost of the software products is zero cost. Other software involved in the construction of our software includes the IDEs and other development tools. In terms of the IDE alone, there is no cost since there are freely available versions. Other involved software includes software to aid project scheduling, such as MS project. This task, however can be accomplished manually and the cost of the software is unnecessary.

3.5. [5 POINTS] Estimated cost of personnel (number of people to code the end product, training cost after installation)

The average hourly wage for programmers in 2017 was \$42.08. [3] Assuming that every worker works for 40 hours a week, and that there will be four workers on the team, the estimated cost of personnel can be calculated as,

Estimated cost of Personnel = 40 hrs/week * 4 personnel * \$42.08/hr = \$6732.80

This cost represents the cost to create the software. We then plan on keeping the same personnel to maintain the software after the release.

4. [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.

The 'searchISBN(String): String' method is used to search the database via a book's ISBN, and if found, returns the pageID of the BookPage which contains that book. The formatting of a book ISBN is as follows: "ISBNs were 10 digits in length up to the end of December 2006, but since 1 January 2007 they now always consist of 13 digits" [4]. Knowing this, the ISBN should be stored as a String containing digit characters only, and its length should be either ten or thirteen characters only. This requirements allow the searchISBN() input String to be easily divided into disjoint subsets, so we have decided to implement the Equivalence Partitioning Black-Box Testing technique.

Valid Input

A range of consecutive strings

0000000000-9999999999

or

00000000000000-99999999999999

Invalid Input

All strings except those consisting of either ten or thirteen character digits only

Test Cases

- | | | |
|-----|-------------|--------------------------------|
| 1) | N = 10 | digit characters |
| 2) | N = 13 | digit characters |
| 3) | N < 10 | digit characters |
| 4) | 10 < N < 13 | digit characters |
| 5) | 13 < N | digit characters |
| 6) | N = 10 | non-digit characters |
| 7) | N = 13 | non-digit characters |
| 8) | N < 10 | non-digit characters |
| 9) | 10 < N < 13 | non-digit characters |
| 10) | 13 < N | non-digit characters |
| 11) | N = 10 | digit and non-digit characters |
| 12) | N = 13 | digit and non-digit characters |
| 13) | N < 10 | digit and non-digit characters |
| 14) | 10 < N < 13 | digit and non-digit characters |
| 15) | 13 < N | digit and non-digit characters |

5. [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.

Even though there are many designs similar to the Book Review Library, the most common design that we came across was GoodReads. GoodReads is a website that was launched in 2007 for a community of book lovers [5]. It consists of a database which contains all the published books and the user can rate as well as review the book. GoodReads also consists of links to purchase the book. Furthermore, the user has to login through a username and a password in order to rate the books. The user can also create a community by adding their friends through a username. If the user wishes to look at their friend's review, they can do so just by searching their friend's username. The website also lets you browse and filter books based on your preferences. In addition, it consists of blogs, challenges, awards, etc [5]. GoodReads has many other features; however, it differs from the Book Review Library in many ways. Book Review Library is a simple website that is only meant for viewing the reviews of various books. The user can filter the book based on their preference, and they can view the reviews of that particular book as well write their own reviews. In comparison to GoodReads, Book Review Library makes it easier for the users to find the books and does not consist of challenges, blogs, awards, etc. This design is simple and only useful for users who are looking to find a review for a particular book. Since the website is simple, it is much faster than goodreads

6. [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.

We planned out and made all the estimates needed to implement a book review library, a website designed to give people a platform to review books and share those reviews with the world via the internet. A few of our functional requirements include the added features that make it possible to create, update, and remove reviews, as well as search for books and pull up all reviews published about that book. We originally wanted to add a feature where users would be able to “like” reviews; however, we decided to create lists for the user to save certain reviews under instead. Users are able to save specific reviews to their lists, which they can create, update, and remove. Having lists instead of “likes” is a more efficient way for users to have fast access to the reviews they want to save, and it allows for organization, as users can create more than one list.

Furthermore, following the incremental process model, we made a decision to have our first release of this product be a website, rather than an app. We figured our first increment should keep its primary focus on functionality and accessibility, rather than concerns with varying installations on different platforms. Creating a well functioning first increment of our product also allows for room to develop, possibly into an application in the future, as many applications start off and progress in this way.

All in all though, we have prepared ourselves for a first release of our book review library, with calculations made to estimate the costs (for effort, time, and money) that give us a good idea of the necessary requirements to get this product finalized. Our cost estimates were calculated using the function-point method, as we saw that algorithmic estimation method to be the best fit, and through research of current hardware/ software pricing, and shown above in this deliverable.

7. [5 POINTS] References: Please include properly cited references in IEEE paper referencing format.

Please review the IEEE referencing format document at the URL:

<https://ieeeditaport.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.

References

- [1] Fiberadmin, “Goodreads Case Study,” *NetHosting*, 19-Apr-2013. [Online]. Available: <https://nethosting.com/goodreads-case-study/>. [Accessed: 15-Nov-2019].
- [2] “Nginx Pricing,” g2. [Online]. Available: <https://www.g2.com/products/nginx/pricing>. [Accessed: 15-Nov-2019].
- [3] “How Much Can a Computer Programmer Expect to Get Paid?,” *U.S. News & World Report*. [Online]. Available: <https://money.usnews.com/careers/best-jobs/computer-programmer/salary>. [Accessed: 15-Nov-2019].
- [4] “What is an ISBN?,” *International ISBN Agency*. [Online]. Available: <https://www.isbn-international.org/content/what-isbn>. [Accessed: 14-Nov-2019].
- [5] Garden, H., HowStuffWorks, Tech, Computer, Internet, Networking and Networks (2019). *How Goodreads Works*. [online] HowStuffWorks. Available at: <https://computer.howstuffworks.com/internet/social-networking/networks/goodreads.htm> [Accessed 15 Nov. 2019].

Also include: 8. [5 POINTS] Your presentation slides. No min/max number of slides enforced. Please make sure that you can complete presentation within 12 (twelve) 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
- Objective of the project designed
- Cost estimation
- Project timeline (timeline of the project designed, NOT the time you've spent on it)
- Functional and non-functional requirements. If too long, select representative items.
- Use case diagram
- Sequence diagram for a selected representative operation of the project.
- Class diagram
- Architectural design
 - Model-View-Controller (MVC) pattern (similar to Figure 6.6) 4
 - 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.
- OPTIONAL: IF implemented the project, a demo of your implementation.