Wolf-Share: Book Sharing Application Phase II

Abhinandan Deshpande[1], Deepak Ravindra Patil[2], Kanchan Bisht[3] and Prashant Nagdeve[4]

***Abstract*— Have you ever felt disheartened when you went to the library to check out a book, but due to its limited number of available copies were not granted the request? There are certain books in the library for which the demand is more and the copies are less to meet these demands. Since getting access to these books is rare, so there are chances that once a student gets access to one of these, he may wish to retain it for a longer time (by renewing it the allowed number of times). This would hinder the chances of others trying to get access to that book. But, during the span that the student keeps a book, he may not use it every day. There are days when he may have assignments, midterms, and quizzes due in other subjects. So, he would end up not using this book. This application seeks to exploit this period to help other students gain access to such books that are checked-out but are not being used by the borrower. But, wouldnt you be skeptical to lend a book to another which is issued using your credentials? We will be introducing the feature of library integration, due to which the person who lends the book from the original borrower would be liable to pay late charges, if any, when the book is not returned on time. Also, we are planning to integrate a chat service into the existing wolf-share application that would ease the communication between the borrower and the lender within the application. Also, geolocation tracking of the book would be helpful for the users as this would give them the option of choosing from whom they can borrow the book according to their spatial convenience.**

I. INTRODUCTION

The Library maintains a certain number of books of each kind that can be borrowed. There are certain books that are low in number as compared to the demand for accessing those books. Due to less availability, it becomes difficult for students to get access to such books. As getting access to these books is rare, so there are chances that once a student gets access to one of these, he may wish to retain it for a longer time (by renewing it the allowed number of times). This would hinder the chances of others trying to get access to that book. Although, the borrower may not use the book for the entire span for which he is allocated the book. He will be attending lectures, will have assignments, projects, midterms and quizzes due for other subjects too. And since, the coursework is challenging, students need to allocate a lot of time (usually few days) to a single assignment, project etc. So, in this span, the borrowed book remains with the student, but is not put into use. The student may be aware that he may not be using the book for certain periods, but, still may not check it in back to the Library, as getting access to it again would be a difficult task altogether. This application makes use of the book when it is not in use by the original borrower by allowing the borrower to lend it to someone in need when he is busy with some other task in hand. This lets the book to be more useful and allows more students to access it. The original borrower would get the book back when he needs it. So, this encourages a student to lend his borrowed book to another.

Although a student may help another student by lending him his borrowed book. But, if the book is due and the other student does not return it on time, then the penalty is charged to the original borrowers account. This happens as the library is not aware of who the book is with. The library only knows that the book had been borrowed by the original borrower. There can be such consequences using the existing application. Therefore, we have decided to integrate the Library as a third party for this application. When the original borrower lends his book to another student, a notification will be sent to the Library containing information about the book, the duration of lending, the original borrower and the next borrower. The Library can then update the information regarding the location of the book. This is essential as this would enable easy monitoring of the flow of payments in case of late returns or damaged books. This would also help monitor that students do not take undue advantage of such borrowings.

Another feature that we have decided to add to the existing application is the geolocation feature. With this, the location of a book would be updated whenever it is moved to another students location. This would enable easy tracking of the book. The students/users who are looking for this book can check the locations where this book is available for borrowing. This would give the users more options and they can decide which one to pursue based on their convenience. Also, the feature of chat service within the application would aid the borrower and the lender to exchange the required information within the applications platform, instead of using emails to communicate with each other.

This application aims to bridge the gap between the existing paid book-borrowing services and the lack of availability of books in the Library. Through this application, users can also lend their personal books within the NCSU network. The idea can be scaled up to meets the book borrowing needs of the entire county, state or country.

*This work is a part of Software Engineering project.

[1]Abhinandan Deshpande is a MCS student in North Carolina State University, Raleigh, North Carolina. Email: aadeshp2@ncsu.edu

[2]Deepak Ravindra Patil is a MCS student in North Carolina State University, Raleigh, North Carolina.Email: dpatil@ncsu.edu

[3]Kanchan Bisht is a MCS student in North Carolina State University, Raleigh, North Carolina. Email: kbisht@ncsu.edu

[4]Prashant Nagdeve is a MCS student in North Carolina State University, Raleigh, North Carolina. Email: psnagdev@ncsu.edu

II. CURRENT SYSTEM

A. Technology Stack:

For Wolf-Share application, current technology stack includes PHP at backend, MySQL as database and AngularJS, JQuery and JavaScript at the front end. There are many advantages of using this technology stack, some of them are as follows:

- PHP is similar to C/C++, and programmers familiar with these languages are able to manipulate and learn it quickly. PHP5 is object oriented which helps us build large and complex applications.
- PHP embeds well in HTML and helps create dynamic websites. It runs on most of servers and it also works well on cross-platforms.
- PHP along with MySQL are excellent choices as, they can be interfaced with each other without any effort.
- Apart from easy interfacing with PHP, MySQL offers unmatched scalability to facilitate management of deeply embedded applications. It also features a distinct storage-engine framework that allows system admin to configure MySQL server for a flawless performance.
- Usage of AngularJS saves time as it binds MVC together automatically. Also, it uses HTML as a declarative language. It does not require any plugins or framework for development.

All these features allowed previous application to be built quickly and effectively.

B. System Overview

In the current application developed by our peers, they provided 3 different options for exchanging books. These are as follows:

- Borrow the book by directly going to lender's house.
- Get booked shipped from borrower's house to lender's house.
- Set up a meeting at some location which is convenient for both the parties involved in exchange (i.e. borrower and lender).

In these methods, lenders are given option to choose between exchange methods. When lender wants to enroll/add book into system, it provides with the choice of options of book exchange out of given options which are mentioned above.

When searching for a book in the system, information about exchange method will also be available for borrower. Depending upon the borrower's choice or convenience he/she may able to select out of different options available for same book according to delivery method. After selection of method for the desired book, borrower will receive information about the lender. After that, they can exchange information about the meeting place via email.

Current application scope does not include any monetary transaction. System is not integrated with any payment gateway and nor does it provide any functionality to handle financial frauds. The application is not responsible for any financial security. Due to this reason, concept of penalty for unwanted or unethical conduct is not applied/implemented. This may include situations where borrower does not return book on time or never returns it back, or returns book in deteriorated condition, or borrower's behavior is not good. There might be a situation where lender showed misconduct or provided wrong information about self or the book. For all these cases owners implemented overall user rating, so that borrower as well as lender will always remain informed about any misconducts occurred previously. User rating involves both borrow rating as well as lending rating. When any misconduct happens, system reduces the rating of the user who is responsible for misconduct from his previous rating. Here, instead of charging money, user rating is deducted as a penalty. Thus, while borrowing book, borrower can have a look at rating of lender before borrowing a book and decide whether lender has good rating and is not penalized for misconduct. Similarly, lender can also take a look at borrowers rating to make sure whether his/her book will be placed in safe hands or not. This feature will force users (lenders or borrowers) to keep their rating high in order to utilize the system to full extent.

The application has two parties- lender and borrower. One user can act as both lender and borrower depending on the situation. User can become lender when he uploads one or more book for lending, he remains lender until the book remains lent. Apart from this time, for rest of the time, user remains as a borrower by default.

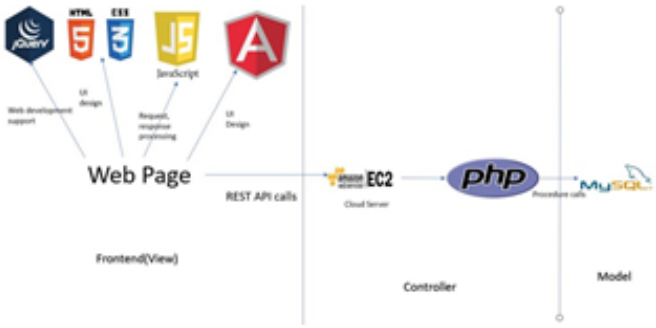

Fig. 1. System Overview

As we can see in Fig 1, the application makes use of AngularJS, which helps build MVC framework automatically. View part of application contains AngularJS, HTML, CSS, JavaScript and jQuery. View receives input from user, then converts it into JSON and sends request using REST API via POST method. The response received from the server is in JSON format and it is then parsed to show desired result. Controller part of system is built using PHP. Controller parses JSON and converts it into structured format which is supported by MySQL. User responses, which are available by querying database are converted to JSON and sent to client as HTTP response. MySQL acts a model for system. It stores required entities and procedures.

C. Application Flow

1) Landing Page, Login and Signup:: Currently, the application only accepts email addresses with ncsu.edu domain names. As we can see in Fig 2, the landing page also gives the facility to search for books even if the user is not registered.

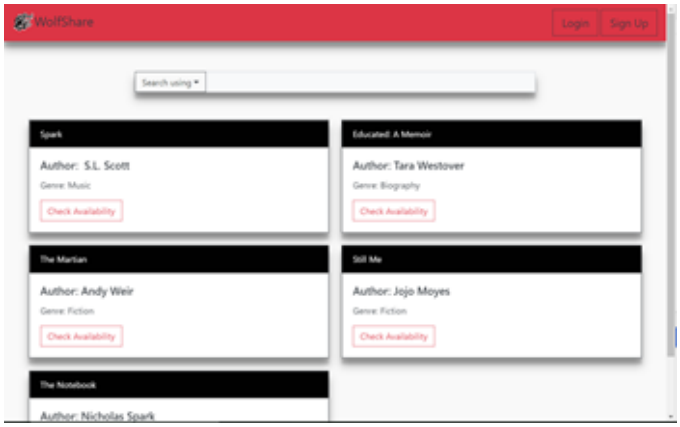

Fig. 2. Pre-login landing page

However, an unregistered user cannot check the availability of a book. When he/she clicks it and is not logged-in a signup/login prompt pops up. A verification code is sent to the email id which needs to be entered by the user while signing up to ensure security. We can see the sign-up and verification pages in Fig 3.

Fig. 3. Sign-up and verification pages

2) Post-login landing page:: The main home page that the user sees after logging in can be seen in Fig 4. It contains a navigation bar on the top which has 5 main tabs:

- Search Tab: This tab redirects the user to the search bar for searching books.
- Add a book Tab: This tab gives the user the ability to add a book for lending.
- Lent Books Tab: This tab is used to display all the books which are currently lent by the user. It also displays the due dates of each book.
- Borrowed Books Tab: This tab is used to display all the books which are borrowed by the user. It also displays the due date of each book.
- Notifications Tab: This tab provides the user with notifications for 3 cases: when a book he has lent is due, when a book he has borrowed is due and when the borrow request has been accepted/denied by the lender.

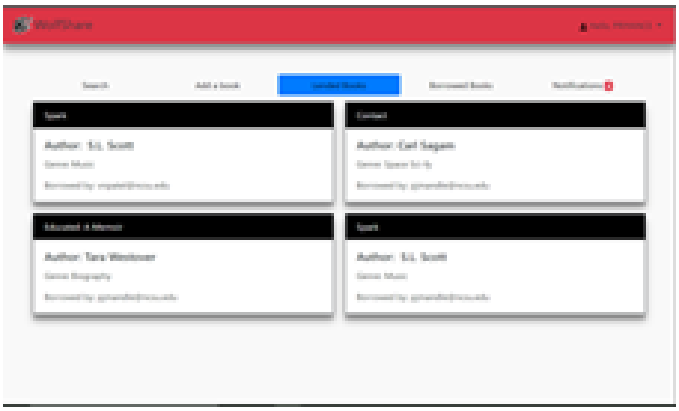

Fig. 4. Post-login landing page

3) Add a book Tab:: This tab redirects the user to a form where he/she can add a book which he/she wishes to lend out. The form contains basic book identification information such as title, author, genre, etc. It also contains a start and due date set by the lender and a mode of delivery. The mode of delivery, currently has 3 options: Meet at commonplace, shipping by lender to borrower and meet at lender's house.

4) Approval Card:: This card-based notification, as we can see in Fig 5, is displayed in the notification tab where a lender gets a request by a borrower and is given the option to approve/disprove a borrower based on the rating of the borrower. This allows a borrower to make an informed decision, whether or not to borrow a book.

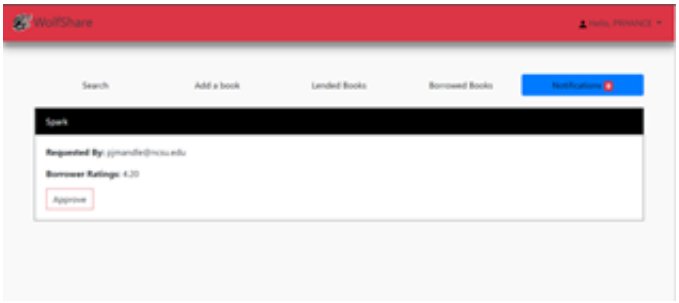

Fig. 5. Approval/Disapproval card

5) Book Card:: This card as seen in Fig 6 displays information about a book which a user has currently searched for and wishes to check its availability. It contains information in two parts. The first part on the left shows information about the book and for what period it is available for borrowing. The second part on the right shows information about the lender, for example his/her name, and ratings/review. This allows a borrower to make an informed decision, whether or not to borrow a book.

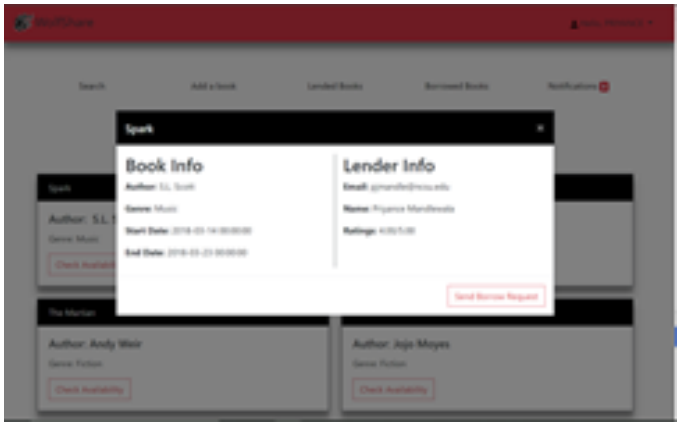

Fig. 6. Book card

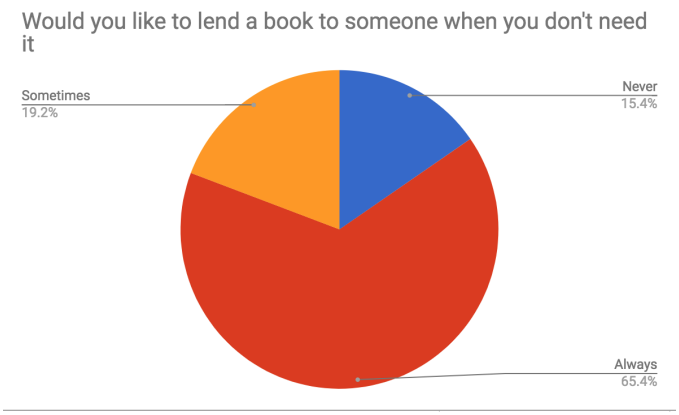

Fig. 7.

III. Survey

In order to learn more about the thinking process of our target audience(university students), we have conducted a small survey to build on the already available product. Before going to the survey we conducted, let us first discuss the survey conducted by the previous group.

- Around 80% people who took the survey were facing difficulties while issuing book from the library since, it was not available.
- About 83% of the students preferred to have a secure web application that could facilitate temporary book lending/borrowing.
- Around 40% of the students were willing to lend it for free. This means that the other 60% were expecting some kind of reward for the service.
- About 84% of the students were comfortable with the rating based system of authenticity.

Let us discuss what we can take from the above found data. Firstly, the students are facing a problem and there are some solutions for this problem which are waiting to be accepted by the community. Secondly, the students did not mind lending a book temporarily. And lastly, people tend to rely on some assuring parameter that indicates the genuineness of the other user(rating of the user in this case).

And now let us discuss the new survey we organized, to see how people responded when we suggested some solutions to them for the above problem.

A. Would you like to lend a book to someone when you don't need it

With this question, we wanted to emphasize on the part when the user doesn't need that book. This will reduce the 'shelf-time' of each book. As we can see from Fig. 7, the people are not sadist in nature. They don't mind lending a book. The 35% of the students who said 'No' for lending must have their reasons. But the most important one of those reasons being, no confidence in the unknown person's ability to return the book on time. Hence we asked the user if they would mind sharing the book which they have issued from the library.

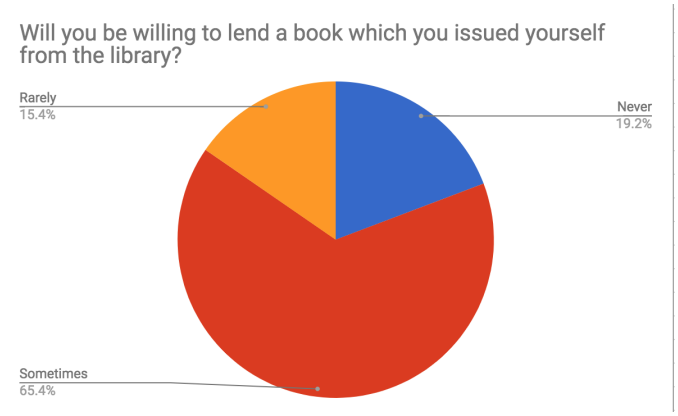

Fig. 8.

B. Will you be willing to lend a book which you issued yourself from the library?

The motive behind asking this question was, to know if people are willing to share a book which they are responsible for. These books (issued from the library) are different from the personally owned books, since the user is responsible for the issued book and would damage his history with the library if he doesn't return it. Hence, as expected, the students are reluctant to lend a book.As seen in Fig. 8, 0% people have said that they would always be available to lend a book issued from the library. Now to counter this problem we suggested them a solution and asked them if they would prefer it.

C. What if the library changes the ownership of that book temporarily (So that late fees will be charged to his account). Then will you lend the book to someone else?

This question was framed in a way that it suggests a solution to the previous problem, and asks about the acceptance possibility of the same. We can see the drastic difference in the number of people saying 'Always' for the same question in Fig.8 and Fig.9, 62% people are willing to lend the book if they are not held responsible for any damages. Now, this

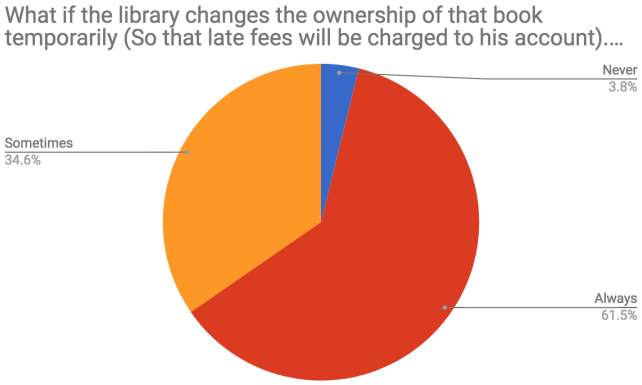

Fig. 9.

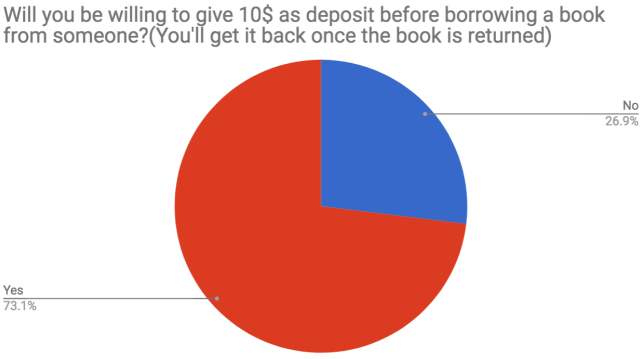

Fig. 10.

can be the basis of our hypothesis that 'People don't mind helping out each other'. Having said that, now we wanted to know, how much money the people are willing to keep as a deposit, in order to borrow a book from an unknown person.

D. Will you be willing to give 10$ as deposit before borrowing a book from someone?(You'll get it back once the book is returned)

Even though we could see around 75% people were fine with the 10$ deposit, but that much amount might not be enough for some cases where the books are expensive. But the Fig.10, demonstrates people's commitment and understanding towards proving their genuineness. Hence, we think that just connecting the user's Paypal account is enough. And if any damage to the book is done, we can charge the user for that amount later.

Hence, we can conclude that adding a payment option is necessary for the users to trust this application.

IV. Proposed plan of Implementation

Currently the system works very well under different scenarios and users can easily borrow books from their fellow students. But there are some areas of improvement. We will be implementing the following features in the application to make it better for users.

A. Payment Integration

Motivation: Once the third party borrower (2nd Borrower) acquires the book from original borrower (1st Borrower) for certain time period, he can use that book for that particular time period. But what if 2nd borrower does not return the book? Or what if he has lost the book? In this case, the original borrower has the liability of book and if he is not able to return the book to the library, he will be charged some amount of penalty based on the duration of returning the book. The user who lent his books will incur a loss which is very unfair situation in real life.

Solution: To address this problem, we have come up with the idea of payment gateway integration. We will be using a payment gateway service of Paypal to achieve this task. If 2nd borrower requests for a book from 1st borrower, he/she will be asked to pay some fixed percentage amount of the price of the book and this fixed amount will be received via Paypal payment gateway. Once he pays the amount, he will be able to acquire the book. After using this acquired book, he can return the book in specified time. Once the 1st borrower confirms the return, 2nd borrower will get back the amount he has paid for the book.

In case, if the 2nd borrower is not able to return the acquired book in given deadline, he will be charged some amount and that amount will be deducted from his/her Paypal account. Penalty charges depend on the duration of book return as per rules defined by NCSU libraries. If 2nd borrower loses the book, same scenario will be applied in this case.

B. Integration with NCSU Library

Motivation: We will try to integrate our application with NCSU library to transfer the liability of borrowed items to 2nd borrower. When 2nd borrower is not able to return the book before deadline, he/she will be charged some amount. This amount goes to original borrowers Paypal account.

After this, paying a penalty by himself/herself (original borrower) becomes his/her responsibility. Due to multiple penalties, original borrowers track record in the Library may get damaged. If penalty is there because of 2nd borrower, only he should pay his/her penalty directly without involving lender (1st borrower) in the transaction.

Solution: To avoid this, we have decided to integrate the application with the NCSU Library. Whenever any transaction of borrowing books happens, library will have a record of it. Consider that user A has borrowed book from the Library and Library has a record of it. Now another user B requests user A for the book. As user A approves the request of user B and user B gets the book successfully, Library will make a note of it. Based on this record, Library can track which user has missed the returning deadline and charge that particular borrower for this transaction. In case, if user loses the book, Library knows that particular user had the book and he/she lost it. In this system, user who misses

the deadline or loses a book will suffer from penalty, not the one who lends the books.

C. Chatbox

Motivation: When 1st borrower approves a request for a book from some 2nd borrower, both the parties need to discuss about the place from where 2nd borrower can get a book from user. Current way of communication is email. But email is private to lender and he may not want to share his personal email. Even if both the users agree to communicate via email, they need to go to website other than that of Book Sharing application.

Solution: In order to facilitate the easy way of communication with other users, we will be implementing a chatbox where users can discuss about the preferred way of exchanging books. With the help of this chatbox, user can easily communicate with other lenders/borrowers without navigating away from Book sharing Web application.

D. Geolocation

Motivation: As of now, the book searching criteria is based on title, genre and author. There might be the possibility that user searches for a book and requests it but lender cannot give the book because of geographic limitations. This results in wasting time of borrower as he was not able to receive the book. Hence it would be nice if we are able to search only those books which can be delivered to us.

Solution: Implementing a Geolocation functionality will help user to avoid false request and he can utilize his time in making request to only those borrowers who can deliver books. Next time when user searches for a book, lenders who are near his location will be displayed to borrower. Borrower also can enter the specific area and can search for users in specific areas.

E. Book Recommendation

Motivation: Most of the times, students struggle with textbooks required for their curriculum as they dont know which book in particular they should refer to, for their subjects. Students navigate on web and find something which may not be good. If user wants to read books related to specific topic, he has to search for best books before he requests for any book on our application.

Solution: We will provide a feature where user will get a suggestion about which book he should read for particular topic and subjects. User also can request other users to recommend books. This will result in more active involvement of users. As a user would know that he will easily find which book he will need for a particular topic, users will use our system very often.

V. Development Life-cycle and Implementation

We will stick to spiral model of development for this project too. In our previous project we used spiral model to develop prototype of functionality and then iteratively we developed our main application. Similarly, we will proceed with spiral model this time too as team members are familiar with this development model and found it comfortable working with this model.

Advantage of using spiral model is that it incorporates iterative prototyping process with controlled and systematic aspects of waterfall model. And also it helps build more complete software with each iteration.

Below is the original diagram of spiral model.

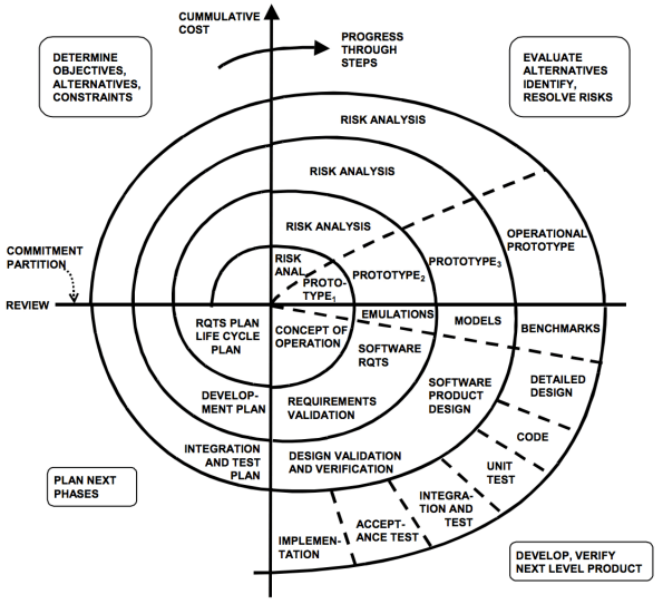

Fig. 11. The original spiral model diagram. Credit Barry Boehm, & Wilfred J. Hansen

A. Requirement Gathering

The first thing we planned to do was to get overall idea of new requirements which user may want or like. In order to achieve this we conducted a survey. Questions were designed to give more insights about users view regarding improvement. Questions were based on whether borrower and lender agree upon transfer of liability or any inclusion of monetary transaction for any misconduct that might rise during lending and borrowing activity.

Apart from the conducted survey, we also talked with library personal about library's role in our application scope. Currently, NCSU library does not allow transfer of liability between users due to various reasons, one of them is high number of hold requests that are made to library for a particular book. And libraries follow first come first serve policy for allocating books.

We are also planning to get more and more suggestions from students and library to improve our product as a whole, as this is what spiral model does, it keeps incorporating the feedback.

B. Design and Build

We are keeping current architecture intact and there will be no change in current technology stack. As mentioned in

proposed improvement in previous section named as 'proposed plan of implementation' we are trying to incorporate payment gateway feature as one of our improvement. There is slight addition in overall implementation architecture.

C. Risk Analysis

Major risk involved in payment gateway integration is security and privacy of user accounts and transactions. If security is not up to the mark, there is a measure of threat to user account details as they can get compromised and altogether entire system may get compromised.

In order to address this, we need to implement encryption and decryption of transactions. There are variety of API's widely available that provides such facility. But before using any API, we need to consider how efficient these APIs are and how difficult it is to break their encryption.

After this analysis is complete, we will be able to proceed further.

VI. CONCLUSIONS

Based on conducted user survey and received feedback, we will try to implement all the necessary functionalities which were suggested by users. In previous system, there was no strong measure to prevent users from any kind of misconduct such as bad handling of book or miss-placing/losing of borrowed book, damaging of book. By implementing such functionality, we can prevent this misconduct and provide better usability. Apart from this, chatbox functionality can also be implemented to provide better communication between users.

ACKNOWLEDGMENT

We express our sincere gratitude towards our Software Engineering professor, Dr. Timothy Menzies, for giving us the opportunity to freely change the assigned project, implement and test it using the tools and technologies of our choice. We also thank our Teaching assistant Amritanshu Agrawal for his valuable guidance and our fellow peers for helping us in requirement gathering by participating in the survey made by us.

REFERENCES

[1] PHP: https://github.com/php
[2] AngularJS: https://github.com/angular
[3] HTML5: https://www.w3.org/html/
[4] PHPMailer:https://github.com/PHPMailer/PHPMailer
[5] CSS3: https://www.w3.org/Style/CSS/
[6] JS:developer.mozilla.org/enUS/docs/Web/JavaScript
[7] MySQL: https://github.com/mysql
[8] Software Engineering: A Practitioner's Approach 8th Edition, Roger S. Pressman, Ph.D., Bruce R. Maxim, Ph.D.
[9] https://github.com/VishrutPatel/BookSharingApplication/blob/master/Report/Wolfshare
[10] https://www.lib.ncsu.edu/borrow/fines