Wolf-Share: Book Sharing Application

Kavi Pandya
kdpandya@ncsu.edu

Nilay Kapadia
nkapadi@ncsu.edu

Priyance Mandlewala
pjmandle@ncsu.edu

Vishrut Patel
vnpatel@ncsu.edu

ABSTRACT

One of the major problems faced by students in libraries is the long waiting list at most libraries when a new book arrives or a new edition is printed. This limited availability of books causes a lot of students to miss out on the latest novels or more importantly the crucial last week of finals. The libraries face a similar dilemma, it is impossible for them to keep a copy for every student in the class, let alone the faculty and other researchers at the university mainly due to lack of funding. Although there exist many applications which allow its users to lend books in their user-base, these applications face two major problems: First, they charge a fee on behalf of the borrowers, this discourages many people from subscribing to such applications. Secondly, they do not have a varied book base, i.e. the users of these lending applications do not possess rare books or research papers which are crucial for college students and professors alike. The solution to this problem is to integrate the services provided by the library and that of such applications into one application where the library can act as a central body for the entire community, from college students to residents around the area. This integration would not only expand the resources at the libraries' disposal (due to contribution by other users), but also solve the problem of shortage of books that the libraries face.

1 INTRODUCTION

Lack of availability of multiple copies of a single book is one of the major problems which libraries face. Due to lack of funding and feasibility, it is not possible for libraries to keep multiple copies of the same book and it becomes difficult for students to get book during the crucial finals week.

There have been many book lending applications which help users to get books for a short amount of time but they have several problems. One of the major problems these applications face is the lack of availability, it is rare to find field specific books among the user-base of such applications and it usually comprises of novels and storybooks, this also affects the variety of the app making the user-base and book-base very acute. To counter these problems, the applications charge a fee, mainly a shipping fee from a far-off location. This fee discourages many users from using the application. Moreover, the lack of development in this sector has produced half-enthusiastic projects which have a lot of software as well as management problems.

The application proposed in this report is a web-application which aims to bridge the gap between these applications as well as the lack of books in libraries. It will allow users of a specific area, say a zip code or a county (currently limited to the Wolf-Pack community), to lend books among themselves through a central body (in most cases, the library). This would allow users, including college students to share books which they have borrowed, among themselves in a secure manner, as well as allow the people of the community to share books among themselves and the library. This would expand the book-base of the library as it would include potential lenders from the community and thereby reducing the problem of book shortage. This would also enable the community to share books among themselves, which they borrowed from the library or their own, in a safe and secure manner.

2 PAST WORK

- To gauge the viability and the feasibility of the project idea mentioned above we carried out a User Survey targeted at NCSU community. Around 70 NCSU students responded to survey and helped us obtain following insights.

- Around 80% of students faced problems when the required book was not available in library and was expensive online.

- Around 83% of students preferred to have a secure web application that could help book exchange by connecting lender and borrower.

- Around 67% of the student indicated willingness in lending book and 40% were willing to lend it for free via such portal.

- And 84% of users were willing to be rated on basis of the book transition.

In amalgamation of the user survey we carried out an elaborate literature survey where we studied the existing online systems/application to understand the feasibility of online book sharing system. In the analysis we performed, we found that the existing system at first suffers from unverified user base and also include commission fee, which in all deteriorates user experience. Thus, a free, secure web application catering to authenticated Wolfpack community of 42,824[1] members (33,755 students) can act as a great medium for lending and borrowing book. This application apart from catering to the book needs of Wolfpack Community also enables them to meet other members of community and moreover results in virtual increase of NCSU's book resource.

With this concept of we proposed the usage of Django framework, with Python at backend and PostgreSQL as the database. But as the project proceeded we had to change the technology stack as mentioned in the next section.

2.1 Change of Technology Stack

In the initial project report we laid out the plan of using Django framework with Python in backend and PostgreSQL as database. This technology stack was new for us as none of the team members were deeply familiar with the above stated technology. Thus, along with the aim of learning these technologies we pragmatically adopted the commit point concept of spiral model. Following its principles, we set up a commit point of achieving the initial signup prototype within the estimated time frame. But on reaching the commit point we were not able to complete the said functionality and the team too was struggling on learning the concept of the technology. Thus, we canned the proposed technology stack and moved towards incorporating the technology that we had familiarity with. Hence, we incorporated PHP for backend development, MYSQL at database end and Angular JS, JavaScript and jQuery in the frontend. All the team member was proficient in some of the above stated technologies which enabled us to have parallel development and were thus enabled to make up for the lost time in technology transition.

2.2 Boundary Conditions

The project consists of several boundary conditions some of those are addressed in the current application and others have been left to be dealt in the future scope of the project.

2.2.1 Meeting of borrower and lender through the application

The application offers three methods of exchanging the books among lenders and borrowers. Borrow from lender's house, get the book shipped from lender's house to borrower's house, and setup a meeting at a common place to exchange the book. In our application, the lender has an upper hand in selecting which of the three methods he/she prefers for exchanging the book. Thus, at the time of uploading the information on book lending into the system we ask the lender to specific one of the above stated method for book exchange. Thus, at the time of book search borrower along with the information of book would be able to view the options through which this book can be borrowed. On selecting the desired book for borrowing, user gets the email information of the lender and it will be via their email communication that they would be planning about the exchange information of books. As a part of enhancing this communication, in the future scope, one could integrate chat box functionality within the application that enables platform based communication on book exchange.

2.2.2 Penalty Fees

Penalty fees is another boundary condition that our mentor suggested to incorporate in the application. Currently, the application is not supporting any money/card based transaction as it requires payment vendor integration and handling of financial fraud which has the potential to compromise the financial security of the application which goes beyond the scope of the current application. For the same we have not incorporated the concept of penalty fees. But, there may be some cases where some form of the penalty is required to maintain the standard of the application. Many of some cases which might require some penalty are as follows - Borrower did not returned the book back to the lender, borrower returned the book but in damaged condition, borrower/lender behaviour was not good etc. To incorporate such behaviours, we are maintaining overall user rating, borrow rating and lending rating. As a result of this whenever the above stated cases occur for the borrower or lender we deduct their borrow or lending ratings respectively which in turn impacts their overall rating. Thus, instead of having the concept of penalty fees we have incorporated the concept of penalty ratings as stated above. Thus, while borrowing the book borrower can look at the lender rating and while lending the book lender can look at the borrower rating. These features promote borrowers and lenders to maintain their ratings.

2.2.3 Absence of Central Body

As the application is developed for NCSU Community we had initially decided to have NC State Libraries as the central body. This was favourable because NC State libraries have been largest book resource for NCSU community and being a part of this application it can not only help connect borrowers and lenders at its location but it also results in a virtual increase of its book stocks. In spite of all these advantages of having NC State libraries at the central body we could not overlook the administrative overheads it would have resulted in Library's functioning due to increase in book request volume, book storage volume, conflict resolution requests and other administrative overheads. This resulted us from not going forward on having NC State Library as the central body. Thus, as of now there is no central body that overlooks the transaction of the books, but it would be a good task in the future to look into if we could somehow associate NC State Library with the application because then then impact and the reach of the application would be tremendous.

2.2.4 Conflict Resolution

There might be cases that borrower or lender have given one other reviews that they do not agree to. Then there might be cases when the book is not returned by the user or is returned in damaged condition. These are some of the cases that is handled by the application only by downgrading the ratings of the user. There are no steps in addition to that which are being performed to manage any such conflicts because there is no central body as such to review these activities. Hence, borrowers and lenders are encouraged to resolve this situation mutually.

2.2.5 Request Disapproval

A lender may receive multiple borrow requests for the same book for the given lending period. But lender can lend the book only to one of the borrower. Thus, all the borrowers who have their request pending will be notified of their request disapproval by lender. But, in case the lender does not respond to the request then there are no actions taken by the system and the request remains in the system as long as lender does not take any action on them.

3. METHODOLOGY

3.1 Project Software Development Model

Agile development was used in developing this application. After reviewing various development methodologies, agile was chosen because of the flexibility it offers, usable modules it offers after each stage and as the development time was less we could at any point of time deploy the last developed feature. So, we could kind of be ready at any point of time. The agile features would help a lot because of short period of delivery time. The ease of implementation of each solution would be known in the early stages of development in this method, and suitable methods of implementation can be chosen quickly. At each stage work was divided among all the team members based on their interest and skills, some functionalities were also developed through pair programming to ensure good code quality and faster development. Hence there was need for collaboration forum and task management for which Github and Trello were used. Github was helpful in setting milestones, raising issues when one of the team members is stuck on something, and also keep track of entire project. Trello was helpful to assign functional task to individuals and set deadlines for the same, so that everyone is aware of what stage of development a functionality is in and if any help is required for the same.

3.2 Architecture

The application has only one type of user which may act as lender or borrower as and when required, the reason being majority of users will act as borrower and lender,hence there is no exclusivity in either of the roles. A user is a lender only when he uploads a book for lending until the book is lended, rest of the time he is a borrower when s/he places borrow request. The graphical user interfaces have a uniformity throughout the application to make it easy for users to use.

The application was developed based on MVC architecture.

View: The view part was developed using HTML, CSS, AngularJS (For design), JavaScript (API calls), JQuery (Web development support). The view is designed keeping in mind ease of use. The functionalities receive input from user, convert it into JSON and call the REST API using POST method passing JSON as parameter. Receiving JSON response from server and display appropriate screen after JSON parsing.

Controller: The controller part is developed in PHP. It parses JSON and passes appropriate output to the SQL Procedures. It also gets user responses of database query which it then converts to JSON and send as HTTP response to client.

Model: The model is developed in MySQL. It includes relational tables for required entities and procedure to interact with the database which are called by the controller.

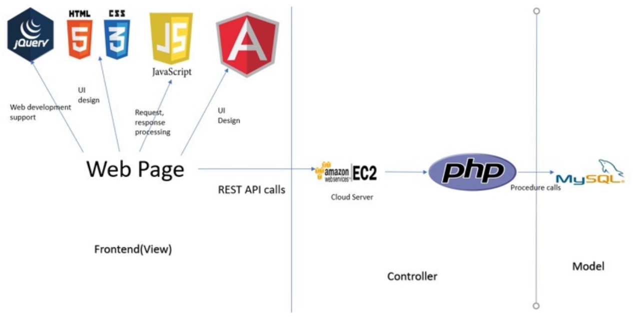


Figure 1: Architecture Diagram.

4 IMPLEMENTATION

The application starts with a pre-login landing page. This page allows anyone landing at the page to search for books currently available, although to check the availability of the book, the user must login. The lender and the borrower both begin at the same login page and are redirected accordingly after login verification. In case of a new user, a signup option is provided. The signup constraints are mentioned below. After logging in, the user is redirected to a landing page whose details are mentioned in section 3.2.

4.1 Signup

New users are redirected to the signup page. Only users with an NC state email-id are allowed to use Wolfshare as it is restricted to the WolfPack community, although this system, in the future can be expanded to other universities as well. The new incoming user must enter basic signup details such as email-id, name, etc. The user is sent a verification code via email for verification. The user must enter the verification code immediately without refreshing the page. In case the user refreshes the page or decides to sign up at a later stage, they must re-enter all information during signup again as well as verify their email. This measure has been primarily implemented for security purposes.

4.2 Landing Page (Post login)

The landing page (page after logging in) consists of 4 main parts. The first is the navigation tabs where the user can check the books he lent or borrowed and a button to lend a book in case he/she wishes to do so. The second part is the search bar. This allows the user to search for books that he wishes to borrow from other users. The user can search books based on 3 main filters: author, genre and name. The user gets a dynamic list of all available books. Each available book is represented using a "card". The information displayed on the card is described in section 3.3. The third part is the Notifications tab which displays all borrow requests in case a lender wishes to lend his book.

4.3 Database

We have seven tables in the database and ten procedures which are described in sub-sections below:

4.3.1 Book

This contains information of the book which has been uploaded by the lender for the purpose of lending. It contains 'id'(int, size of 6) an auto incremented unique identifier for the book, 'email'(varchar, size of 50); contains NCSU Email i.d. of the lender, 'title'(varchar, size of 50); it is the title of the book, 'author'(varchar, size of 50); contains the name of the author of the book,'genre'(varchar, size of 50); specifies the genre of book such as fiction, mystery etc., 'start_Date_Time'(timestamp) contains the start time and date when this book would be available for lending and 'end_Date_Time'(timestamp) contains the end time and date when this book would be returned by borrower.

4.3.2 Borrower

This contains information of the borrower. It contains 'id'(int, size of 6) an auto incremented unique identifier for the borrower [this is necessary because a single borrower can place multiple borrow request for multiple books and in such case email is no longer a unique identifier to access individual borrow request, hence we use auto incremented id to identify each borrow], 'email'(varchar, size of 50); contains NCSU Email i.d. of the borrower, 'ratings'(float - 3,2); contains aggregate ratings(0 to 5) for the borrowers as rated by all the lender s/he borrowed book from. 'No_of_reviews'(int, size 11) are the total number of reviews submitted by lender for this borrower.

4.3.3 Borrow Request

This table contains the information of the book borrow request by the the borrower. It contains 'book_id'(int, size of 6) an auto incremented unique identifier for the book, id'(int, size of 6) an auto incremented unique identifier for the borrower, 'borrower_email'(varchar, size of 50); contains NCSU Email i.d. of the borrower, ''start_Date_Time'(timestamp) contains the start time and date when by which the borrower wants the book, 'end_Date_Time'(timestamp) contains the end time and date by which book would be returned by borrower.

4.3.4 Lender

This contains information of the lenders. It contains 'id'(int, size of 6) an auto incremented unique identifier for the lender [this is necessary because a single lender can lend multiple books and in such case email is no longer a unique identifier to access individual lending, hence we use auto incremented id to identify each lender], 'email'(varchar, size of 50); contains NCSU Email i.d. of the lender, 'ratings'(float - 3,2); contains aggregate ratings(0 to 5) for the lenders as rated by all the borrowers s/he borrowed book from. 'No_of_reviews'(int, size 11) are the total number of reviews submitted by borrower for this lender.

4.3.5 User

This table contains all the required information of user such as; 'firstname'(varchar - size 30), 'lastname'(varchar - size 30), 'addr1'(varchar - size 100) which is a street address, 'addr2'(varchar - size 100) which is apartment number, 'city' (varchar - size 100), 'state' (varchar - size 2) containing the abbreviations of the state such as 'NC' for North Carolina, 'zipcode' (varchar - size 5) such as 27606, 'email' (varchar - size 50) that is NCSU email of the user, 'reg_date' (timestamp) is the date when User registers into the application, 'sec_code' (int - size 5) which is the security code user is emailed to authenticate their sign up process, 'verification_status' (tinyint - size 1) which indicates whether user has been verified or not, 'password' (varchar - size 30) containing the password of the user.

4.3.6 Review

This table contains data related to reviews for a book transaction Id (integer primary key),transaction_id (integer, refers to transaction table), borrow_book_condition_review (integer),borrow_review_text(text review for condition of a book),borrow_exp_review (integer value indicating if experience review is given or not) borrow_exp_text (experience review text),lender_review_approve (review approval status), return_book_review (return book review in 0-5),return_review_text(text review for return varchar),return_exp_review (return star review),return_exp_text (textual return experience review) , borrower_review_approve (review status).

4.3.7 Transaction

This table contains required information of transaction such as: id (integer of length 6, auto increment), lender_email(varchar 50), borrower_email (varchar 50), book_Id(integer of length 6), start_Date_Time (datetime format), end_Date_Time (date time),Location (varchar of length 50).

4.4 Database Procedures

This project incorporates the concept of reusability and modularity wherever feasible. For the same, procedures have been defined for the purpose of accessing/adding/deleting/modifying the information in the MySQL database. Procedures are subroutines which contains optional input and output parameters with statements accessing the database. They can be invoked as and when required just like functions but provide more system level access. These features make procedures a viable option in module based development. This section covers the procedures developed in the project and the tasks it performs.

4.4.1 addSignUpInformation

This procedure is called when user has filled up the SignUp form on the SignUp page of the application. This procedure adds following information of the user into the MySQL database; First Name, Last Name, Address 1, Address 2, City, State, ZipCode, Email, SignUp date, Security Code, Verfification Status and the Password.

4.4.2 addBookForLending

This procedure is called upon when user uploads his book on the application for the purpose of lending it. To lend a book this procedure takes following data from the lender; Email, Title of book, Author of book, Genre of book, starting and ending date of the book lending period. The purpose of recording author and genre of the book along with the title is be

4.4.3 addBorrowRequest

This procedure is called upon when user wants to borrow a book on the application from the lender. To place a borrow request this procedure takes following data from the borrower; Email, bookid [mapped to the title of the book selected by the borrower], starting and ending date of the book borrowing period.

4.4.4 displayBookSearchResult

This procedure is called upon when the user wants to search the availability of the books in the application. This functionality would mainly be used by the borrowers to search the books catalogue. This search functionality takes following data to display the result; Title or genre or author of the book, start and end date for borrowing it and in the output the application will provide the information of the lenders lending the requested book and the lending period matching to that of the requested borrow period.

4.4.5 forgotPassword

This procedure is called upon when the user has forgotten their password. This procedure takes the new password from the user and updates the old password with the new one.

4.4.6 retrieveBorrowRequests

This procedure will fetch the number of borrow requests, borrower name and his/her ratings for a book for notifying the book lender.

4.4.7 signUpValidated

This procedure is called upon when the User has successfully signed up using the security code. The task of this procedure is to update the user record as a verified user.

4.4.8 validateLogin

This procedure would be called all the time the user logs into the system. This procedure confirms the users email and password and then allows access to application.

4.4.9 validateSignUp

This procedure would be called upon when we need to check whether the user is in the system or not.

4.4.10 validatedOrNot

This procedure is called to check the verification status of the User.

4.5 Installation Instructions

1. Install XAMPP or WAMPP (link provided) XAMPP:https://www.apachefriends.org/download.html WAMPP:http://www.wampserver.com/en/

2. Clone the Git Repository in the htdocs folder or www folder for xampp and wampp respectivley.

3. Setup ports in the XAMPP/WAMPP for MySQL (using PHPmyAdmin) and localhost.

4. To setup the Database, open PHPmyAdmin in your browser and create a database called "booksharingapplication" (without quotes).

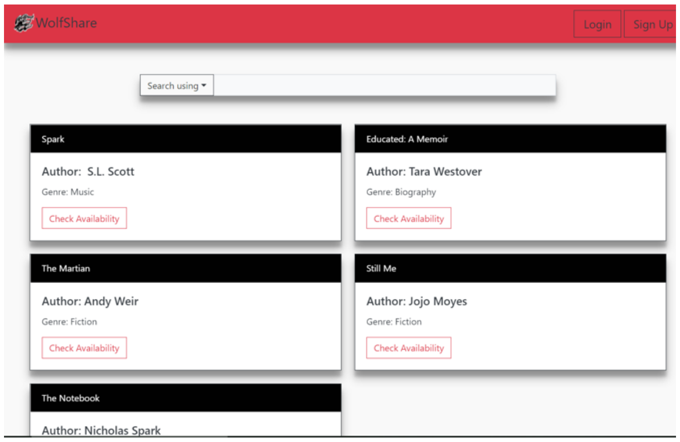
5. Then click on the import button in PHPmyAdmin and provide the file "booksharingapplication.sql" inside the procedures folder to import the database, tables and procedures.

6. Run localhost:portno/BookSharingApplication/

5. APPLICATION INTERFACE AND FLOW

5.1 Landing Page, Login and Signup

The screenshot below shows the landing page(pre-login) as well as options to sign up (for new users) and Login (for returning users). Currently, the application only accepts email addresses with ncsu.edu domain names. The landing page also gives the facility to search for books even if the user is not registered. This makes the application more open and does not force the user to signup initially. Moreover, 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. Additionally, for security reasons, a verification code is sent to the email id which needs to be entered by the user while signing up.



Spark
Author: S.L. Scott
Genre: Music

Educated: A Memoir
Author: Tara Westover
Genre: Biography

The Martian
Author: Andy Weir
Genre: Fiction

Still Me
Author: Jojo Moyes
Genre: Fiction

The Notebook
Author: Nicholas Spark

Figure 2: Pre-login landing page

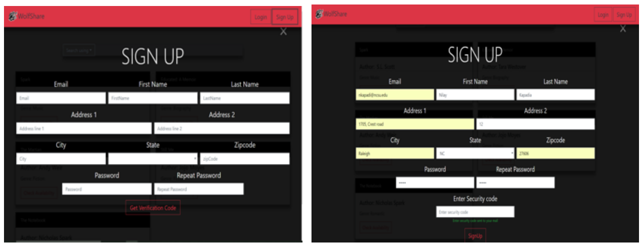
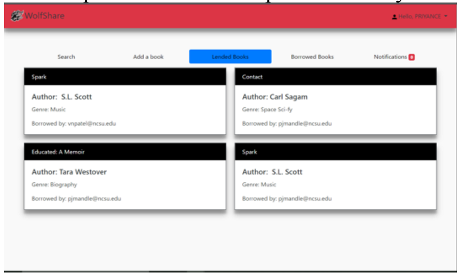


Figure 3: The signup and verification pages.

5.2 Post-login landing page

This is the main home page that the user sees after logging in. It contains a navigation bar on the top which has 5 main tabs:

1. Search Tab: This tab redirects the user to the search bar for searching books.
2. Add a book Tab: This tab gives the user the ability to add a book for lending. Further description about the contents of this tab are in section 5.2.1.
3. 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.
4. 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.
5. Notifications Tab: This tab provides the user with notifications for 3 cases: In case a book he has lent is due, in case a book he has borrowed is due and in case a borrow request has been accepted/denied by the lender.



Spark
Author: S.L. Scott
Genre: Music
Borrowed by: vnpatel@ncsu.edu

Contact
Author: Carl Sagam
Genre: Space Sci-fy
Borrowed by: pjmandle@ncsu.edu

Educated: A Memoir
Author: Tara Westover
Genre: Biography
Borrowed by: pjmandle@ncsu.edu

Spark
Author: S.L. Scott
Genre: Music
Borrowed by: pjmandle@ncsu.edu

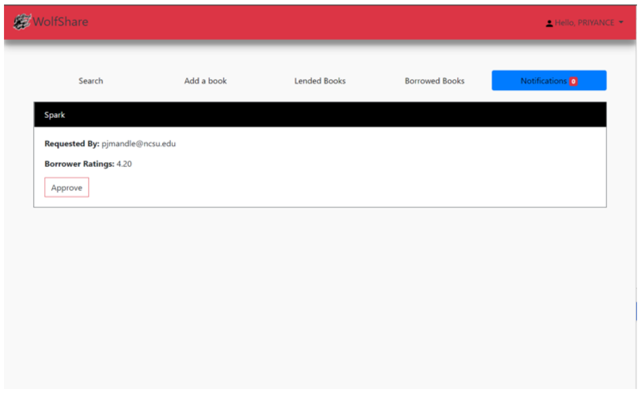
Figure 4: The post-login landing page.

5.2.1 The 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.

5.3 Approval Card

This is a card-based notification which 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.

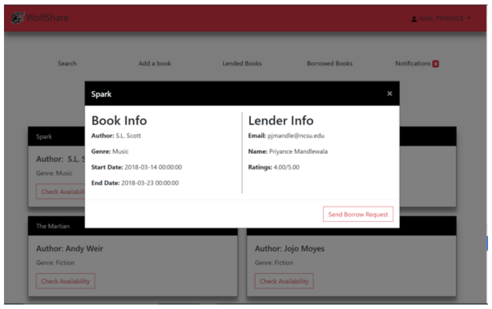


Spark
Requested By: pjmandle@ncsu.edu
Borrower Ratings: 4.20

Figure 5: Approval/Disproval Card

5.4 Book Card

This card 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.



Spark

Book Info
Author: S.L. Scott
Genre: Music
Start Date: 2018-03-14 00:00:00
End Date: 2018-03-23 00:00:00

Lender Info
Email: pjmandle@ncsu.edu
Name: Priyance Mandlewala
Ratings: 4.00/5.00

Spark
Author: S.L. S
Genre: Music

The Martian
Author: Andy Weir
Genre: Fiction

Author: Jojo Moyes
Genre: Fiction

Figure 6: A Book Card

6 USER EVALUATION

We evaluated our software with the help of 12 students manually experimenting and testing our client-side app by hosting the app on AWS. We felt that it would be very difficult for our reviewers to subjectively review our application as it would require a comprehensive long-term review. So, we kept our evaluation metrics to the point and easier to review.
We primarily included rating based reviews for the reviewers to rate the application on factors such as ease of use, the general user experience and places to improve in the future and functionalities to be added.
The two main issues we found with our current version of the website are the slow email-verification service and lack of "history of books" maintenance, i.e. the ability to view previously lent/borrowed books.
The reason for a slow email-verification service is due to the use of SMTP service (by Yahoo!) for sending emails to the user. This can be easily resolved by opting for a paid, more comprehensive service. The second problem which we overlooked while developing the application is the ability to look at previously lent/borrowed books. This can be easily resolved.
The reviews also mentioned many places to improve, most of which we have incorporated in our future work. The main suggestions included the ability to give a review of a user which we have mentioned in section 7.1 and the ability to chat with the lender/borrower post request processing which we have mentioned in section 7.2 of the report.

7 RESULTS

We based our results on 5 main questions we asked to the users which are mentioned in the subsections below. Each subsection shows a detailed graph of the entire review process for each question. These questions are primarily objective for ease of review but we have incorporated a few subjective questions primarily in the form of suggestions which we have incorporated in the above section.

7.1 How appealing is the idea

During evaluation we wanted to know the feasibility of the idea and how appealing it is for NCSU students to actually use it for their book sharing purpose. Of the 12 students who used our application, 7 rated the feasibility of the idea as 5 and 5 rated it as 4 on the rating scale from 1-5. This shows the potential on the usage of the application.

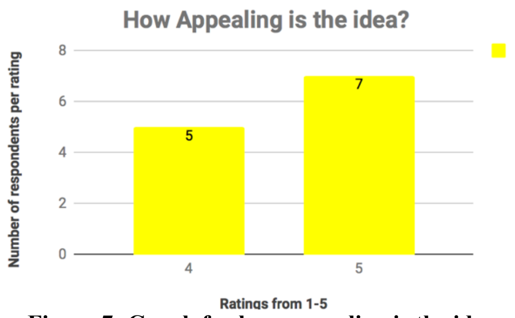


Figure 7: Graph for how appealing is the idea.

7.2 Overall rating of the platform

Next, we asked the reviewers to review entire system as whole, including its functionality, ease of usage, completion of required task of lending/borrowing book, searching of required books on genres, title, author and other such functionalities. Of the 12 students who reviewed our application, 9 rated the overall system as 4 and 3 rated it as 5 on the rating scale from 1-5. These ratings indicate that most of the users were satisfied with the current working of the system and the functionalities it provides.

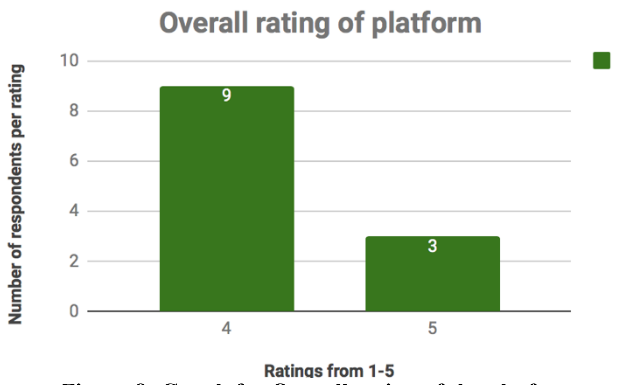


Figure 8: Graph for Overall rating of the platform.

7.4 Would it be easier to procure books through this app

The main aim of the project was to make it easy for the NCSU community to share their books via this application. Thus, we asked the reviewers to review the process of lending/borrowing process of the application and thus review the procurement process of the books via this application. Of the 12 students who reviewed our application, 91.7% of reviewers stated that it was easy to procure book through this application where as 8.3% of reviewers stated that it may be easy. These reviews enabled us to conclude that the application was successful in achieving its aim of making it easy and convenient for NCSU community to share books among themselves.

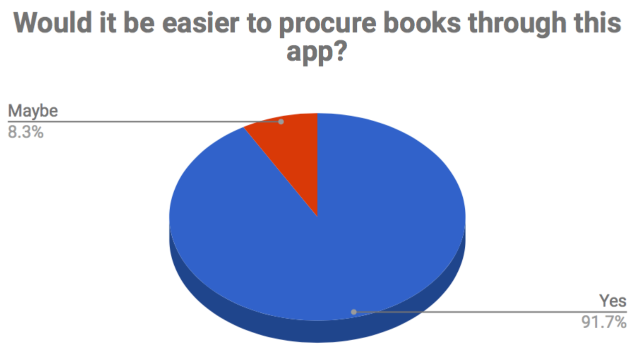


Figure 9: Pie-Chart for ease of use of the application.

7.5 User Interface Rating

To enhance user experience in using the application we have worked extensively on providing responsive User Interface developed in JavaScript, AngularJS, HTML5 and CSS3. Of the 12 students who used our application, 7 rated the interface rating of the idea as 5 and 5 rated it as 4 on the rating scale from 1-5. This shows the potential on the usage of the application.

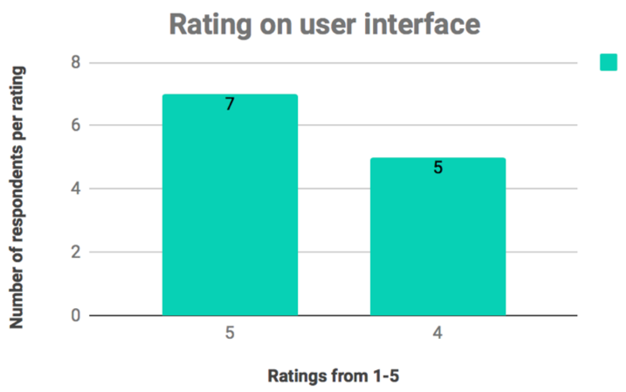


Figure 10: Graph for quality of UI.

7.6 Performance rating w.r.t. response time

Of the 12 students who used our application, 3 rated the response time of the idea as 5, 8 rated it as 4 and 1 rated it as 1 on the rating scale from 1-5. This shows the potential on the response time of the application.

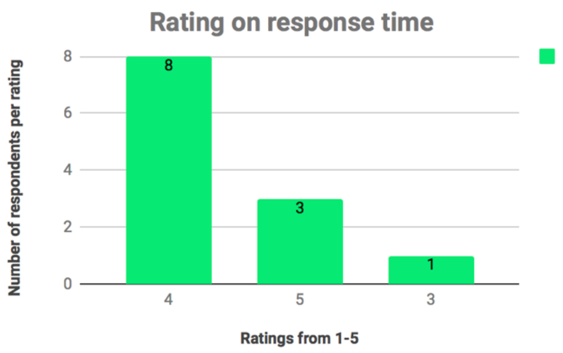


Figure 11: Graph for Response times of the application.

8 CONCLUSION

Based on the survey we conducted at the early stage of the project, we found out that approximately 83% of people would like to get a web application for connecting lenders and borrowers. This shows that there is an existence of a gap between systems justifying the purpose of this application.

From our extensive surveys after the development of the web application we can conclude that about 89% of the students found that it would be easier to procure books through the app. Talking about the technological development most of the surveyors rated an average of about 4.3 on 5 for different aspects like UI, Response time, overall functioning of the platform, user liking of the idea.

However, the existing system requires few enhancements in the functionalities before being available for general use. Some of the most important requirements include a way to maintain the integrity of the system through a central body, Email notifications on approval of a borrow request, proper review systems, If required a form of deposit to ensure proper maintenance of books, chat box for discussing about places to meet.

If the above functionalities are integrated this could be helpful for the wolfpack in terms of knowledge sharing as well as networking among the fellow NC State Students. It could also help students to discuss among fellows with similar taste and help each other achieve greater heights.

9 FUTURE WORK

9.1 Review Mining

The current scope of the project takes into consideration the ratings based on clickable stars. Where user can rate lending/borrowing experience from 0 stars to 5 stars. This methodology of obtaining ratings makes it easy to compute rating but at the same time does not cover the reason for providing such ratings. For the same we have provided a textbox where user can describe the reasons for giving such ratings and state their experience. Currently, we are just adding the text in the database and not performing any sentimental based analysis on it. But going forward we can incorporate review mining using natural language processing such as nltk, natural language toolkit. Through language identification, segmentation, normalization, classification, disambiguation and relationship extraction we can obtain sentimental insights on the reviews which may help us better understand the reviewing by the user and thus we can better rate the user.

9.2 Chatbox

Once the borrower places the book request and the lender approves it they are required to discuss about the book exchange via email. That is, there is no platform based communication channel which could allow them to discuss about book exchange on the web application itself. Currently we are just providing lender and borrower each other email id and they are motivated to carry on their communication via email. This is not much of an convenience because for such discussion they have to go away from the web application for multiple times. Thus, it would be a great to provide the chat box functionality that could allow lender and borrower to sort out their book exchange related communication on the application itself.

9.3 Geolocation

Currently the book searching criteria is on the basis of title, genre and author. In addition to this it would be viable to search book based on geographical region. That is if one book request results in multiple borrow options from multiple lenders then it may be more viable for the borrower to take the book from the lender which is geographically near to him/her. Thus, this functionality would help connect lender and borrower who are nearby to one another. Several geolocation API's are available for PHP and are supported by all the major browsers.

9.4 Book Suggestion

To incorporate more involvement of the users on the website we can provide a notification feature where one user can suggest the book s/he came across the application to their friend. This would result in more active involvement of users and they could get time update on availability of new books in the system via their friends which otherwise the might miss out.

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

Chit Numbers:

1. PMY
2. VBO
3. XRB
4. FII
5. SAA
6. QMV
7. LMA
8. ERH
9. ZHS
10. ZDC
11. ZIQ
12. HTB