



BookShare

TEAM-14

TEAM MEMBERS:

SWETHA CHANDRA KARROTI-21

PALLAVI RAMINENI-49

RAJARAMYA JANAGAMA-17

LATHA MUDDU-29

INTRODUCTION:

This Project allows users to share books among various registered users. This process of sharing books saves time. One need not go to the library and search for the books, rather they can just search for the required book using this app. The book collection centers (Library) are very vast and it is quite difficult to search for the required book in thousands of books available there. Instead of consuming the time, the user could easily get the same book by using this Application. The user could ask for the required book in the group. Then the user gets response regarding the availability of the book and the nearest pickup point.

OBJECTIVE:

Our basic point in selecting this project is to utilize the time effectively and to build up communication between people belonging to the same group. Also the user gets popup messages which reminds to collect the book on a particular date .Information about the return date of the book is also available in the application for which the user will get a popup message regarding the same.

FEATURES:

1. First each user must register in the login page of the application by providing the following fields:
 - User Name
 - Password
 - Full Name
 - Email Address
2. After registering into this application, each user must update all the books he has with him.
3. This directs to another page where he includes the following fields:
 - Book ISBN
 - Title
 - Author
 - Book Category
 - Availability – Default value 1.
4. On the other hand, the Borrower who wants to request for a book will have a Request option, where they can search for the books.
5. This search can be done with the following fields:
 - Search by Title
 - Search by Author
 - Search by Category
6. After the Borrower searches for a book, he gets the list of users having the book he requested.
7. Each user who has the book requested, specifies their Availability options which includes:
 - Date

- Time
- Price

8. Now the borrower selects his preferences and chooses one among the available users.
9. Then a notification is sent to the selected user and he can accept or deny the request.
10. If the request is accepted then the user provides the location to collect the book.

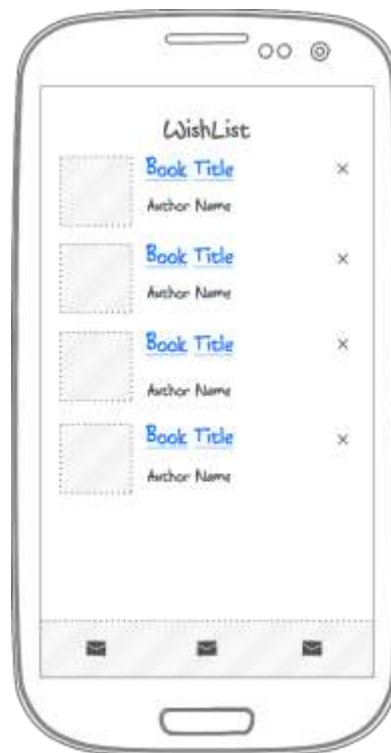
EXISTING SERVICES/API:

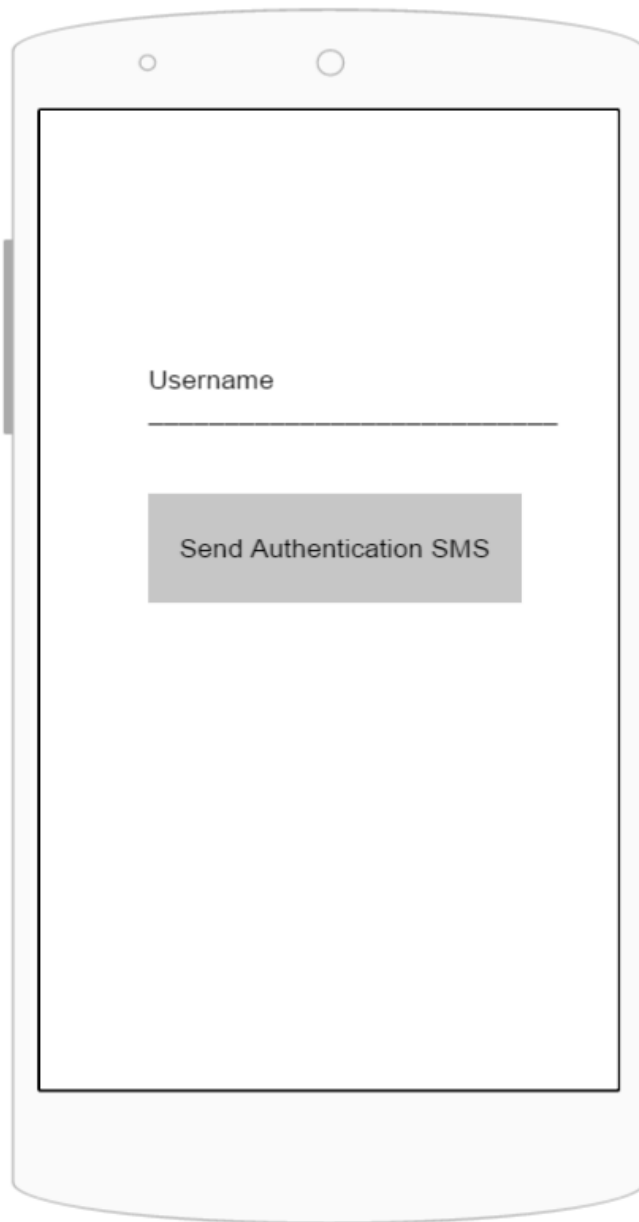
1. Application was developed using Ionic Hybrid Mobile apps Framework on Jet Brains WebStorm IDE.
2. MongoDB is used for storing the data, MongoDB is being hosted on mlab.
3. Backend Services are built using Nodejs, Nodejs is hosted on Heroku platform.
4. Gravatar API is being used to pull the images of the users by computing MD5 hash of their email addresses and the image is being used in Account settings page.
5. Google Books API is being used to retrieve data like Book Description, Author Names, ISBN Verification, Cover Image, Page Count, and Average Rating.
6. Amazon Product Advertising API is being used to retrieve Customer Reviews, Price of the Product Cover Image.
7. Twilio Programmable SMS API is being used to serve SMS notifications to users.
8. Cordova TTS Plugin is being used to convert Text to Speech on Android Devices.
9. Cordova Contacts plugin has been used to Invite People from Phone book to invite users to the application.
10. Cordova Facebook Login Plugin has been used to authenticate user to use the application.

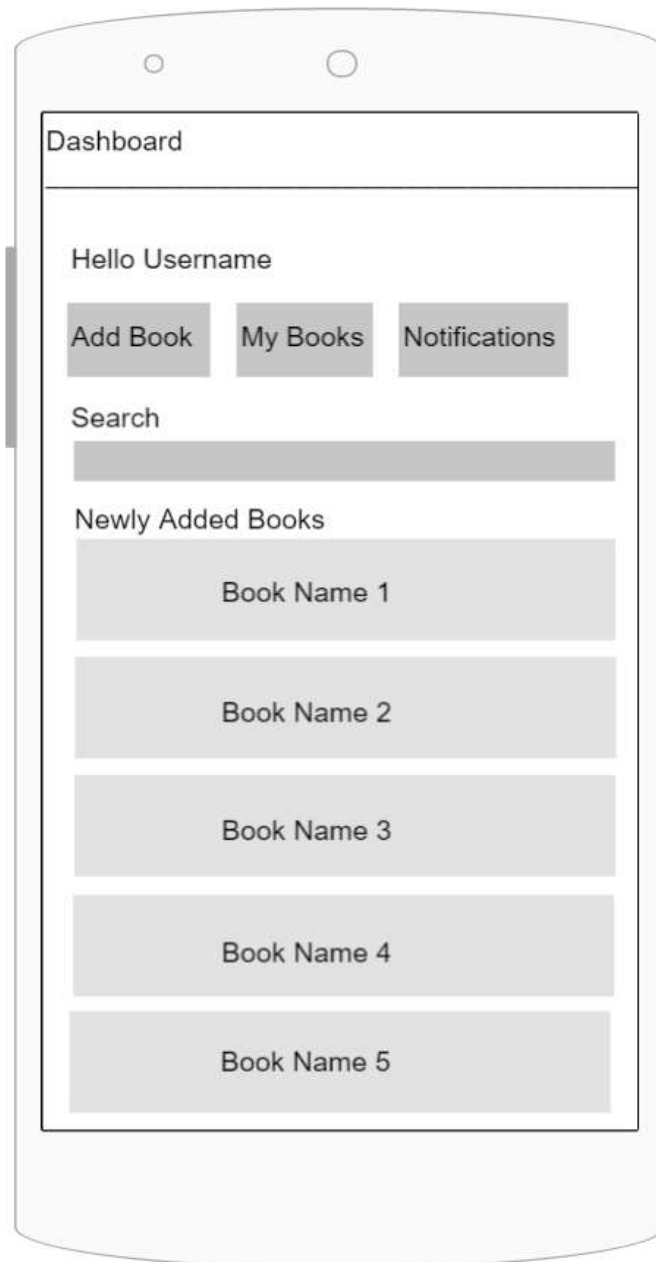
DETAIL DESIGN OF FEATURES (Using Tools):

WIREFRAMES:

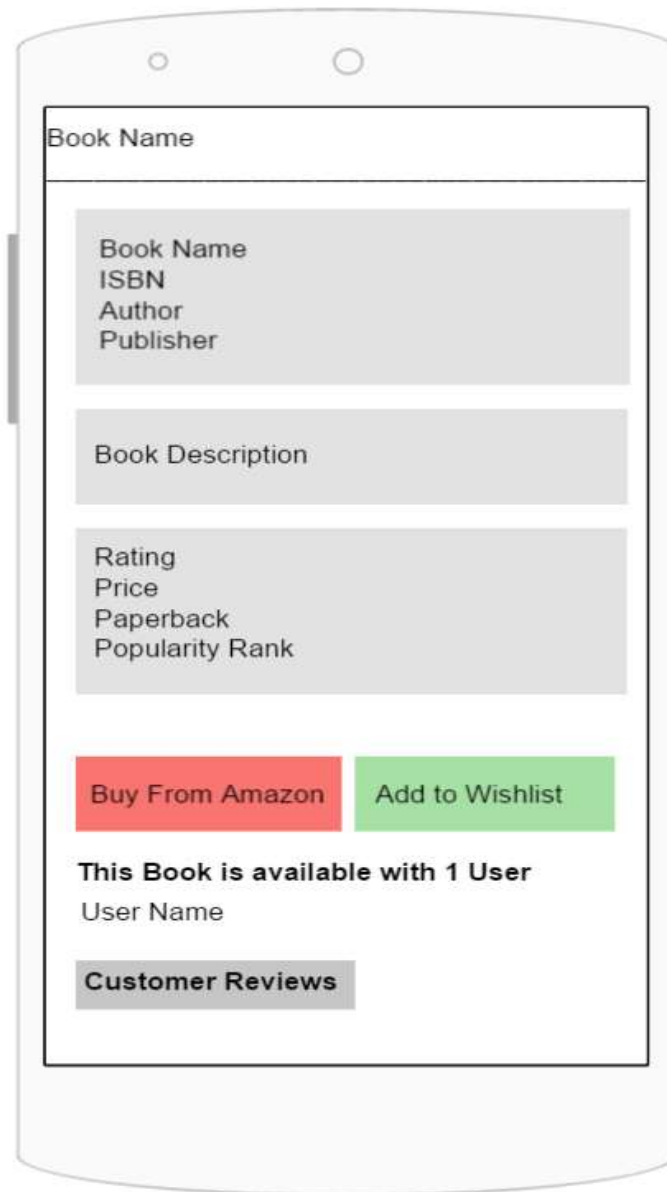


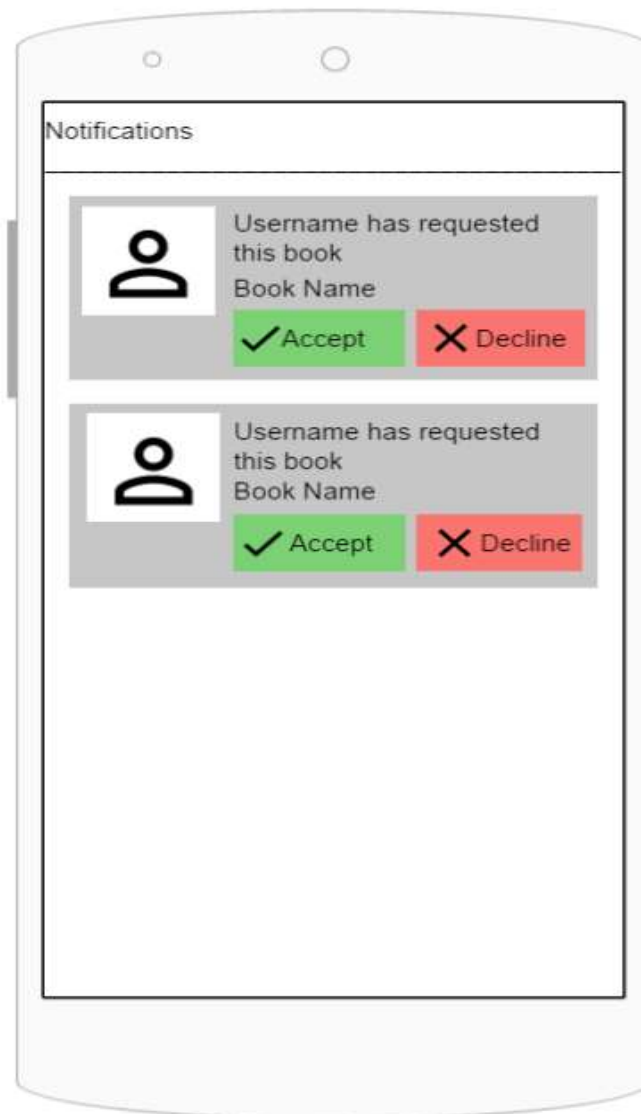


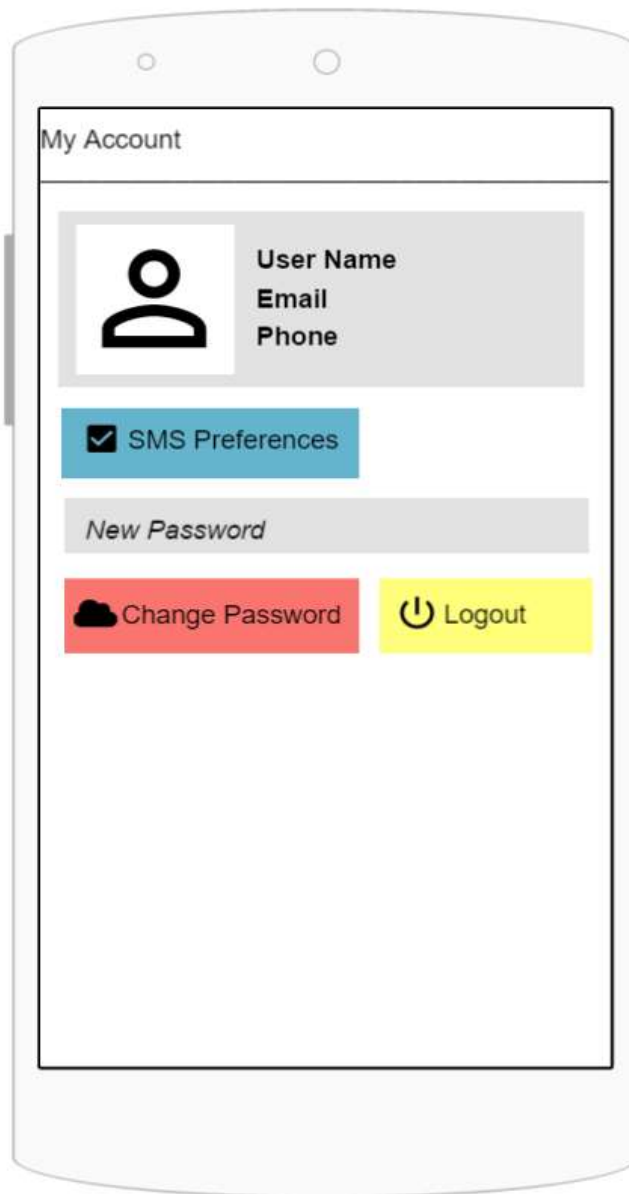


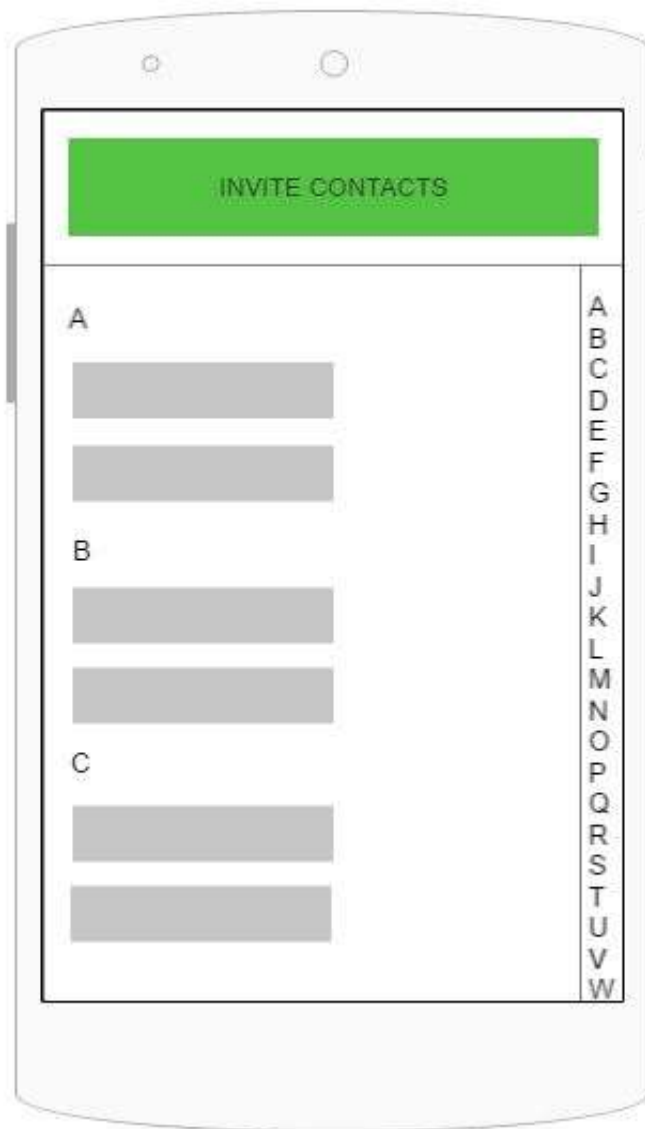


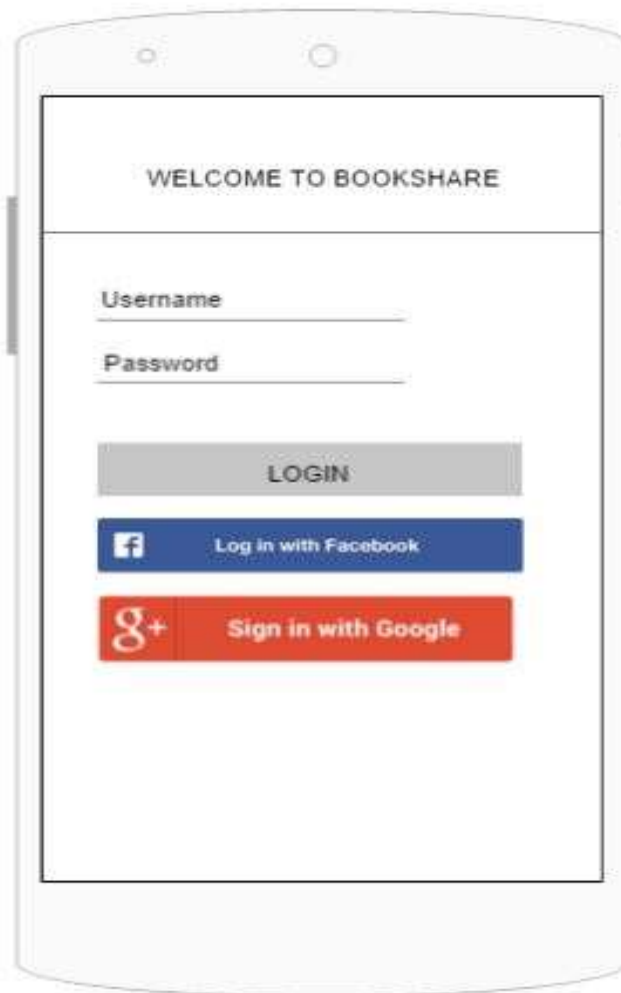




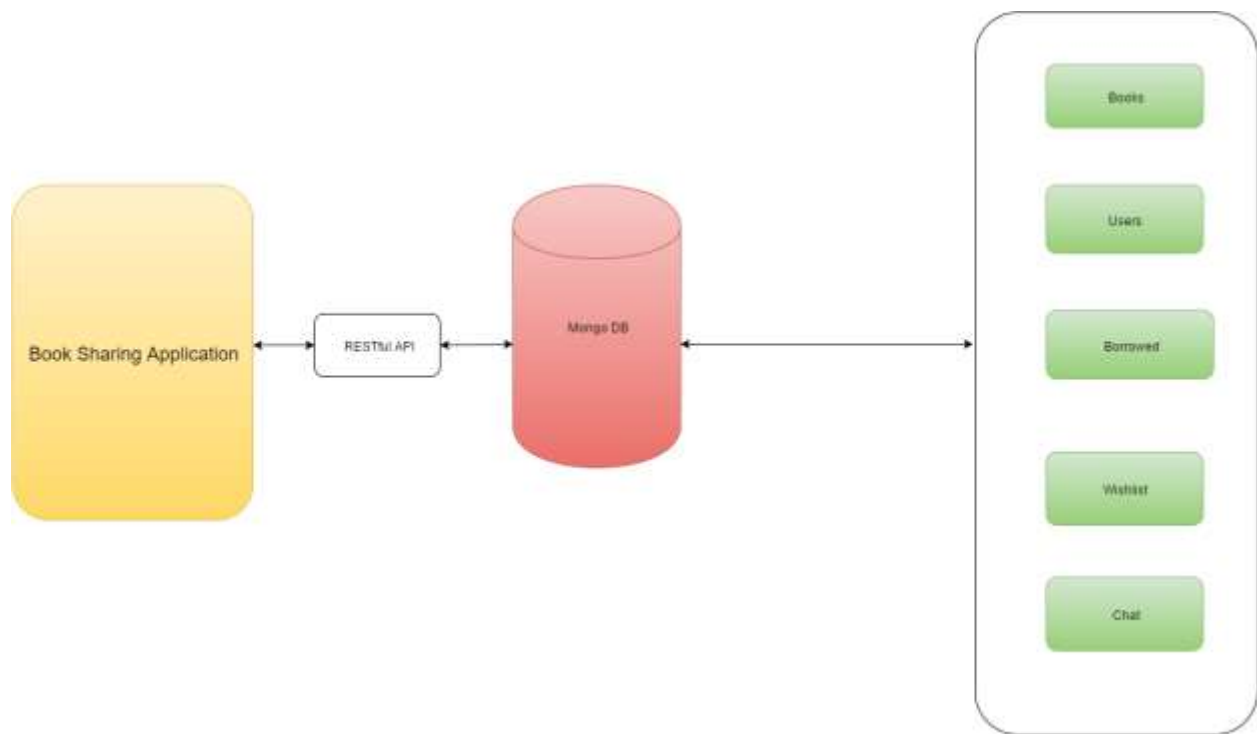




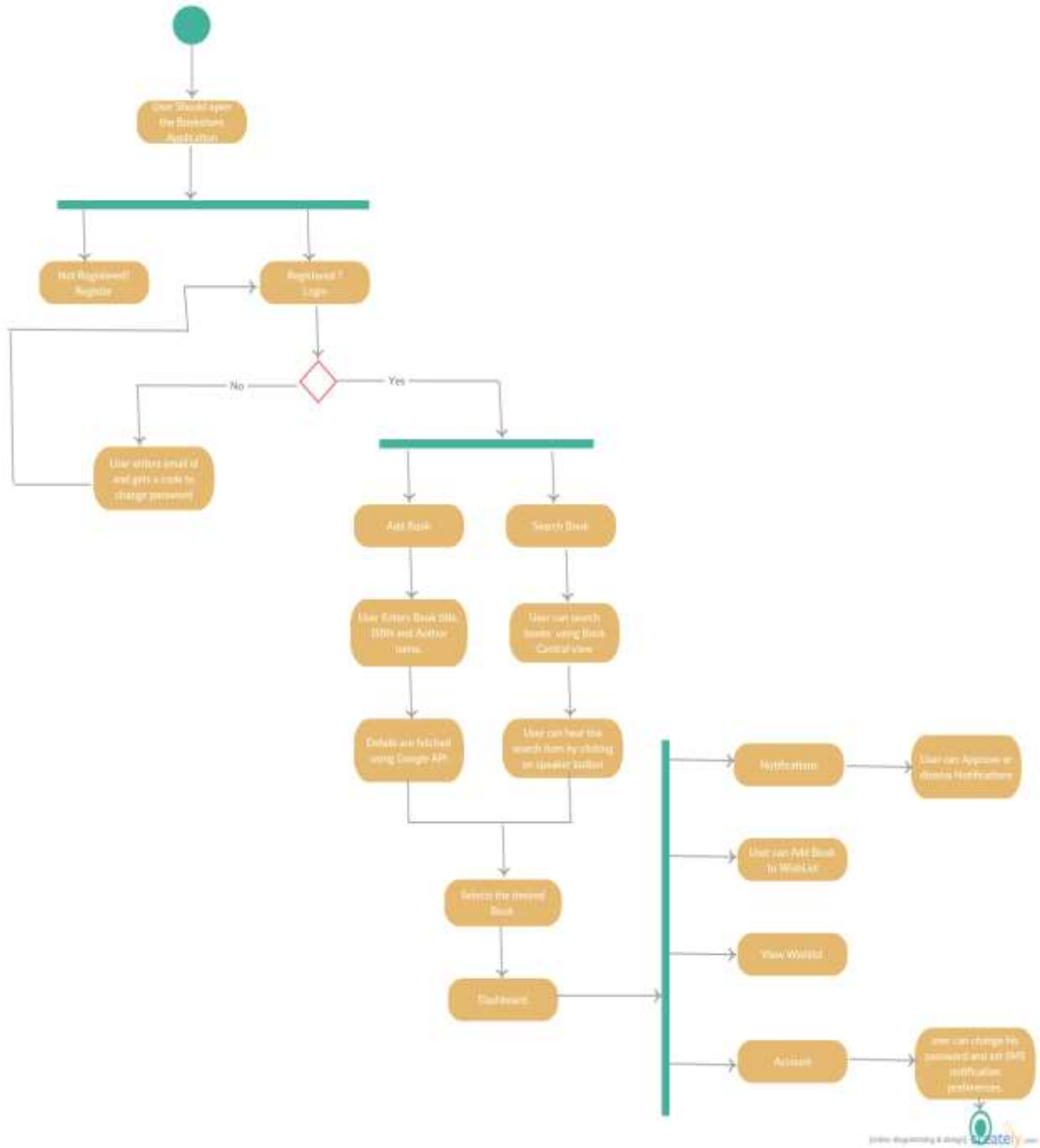




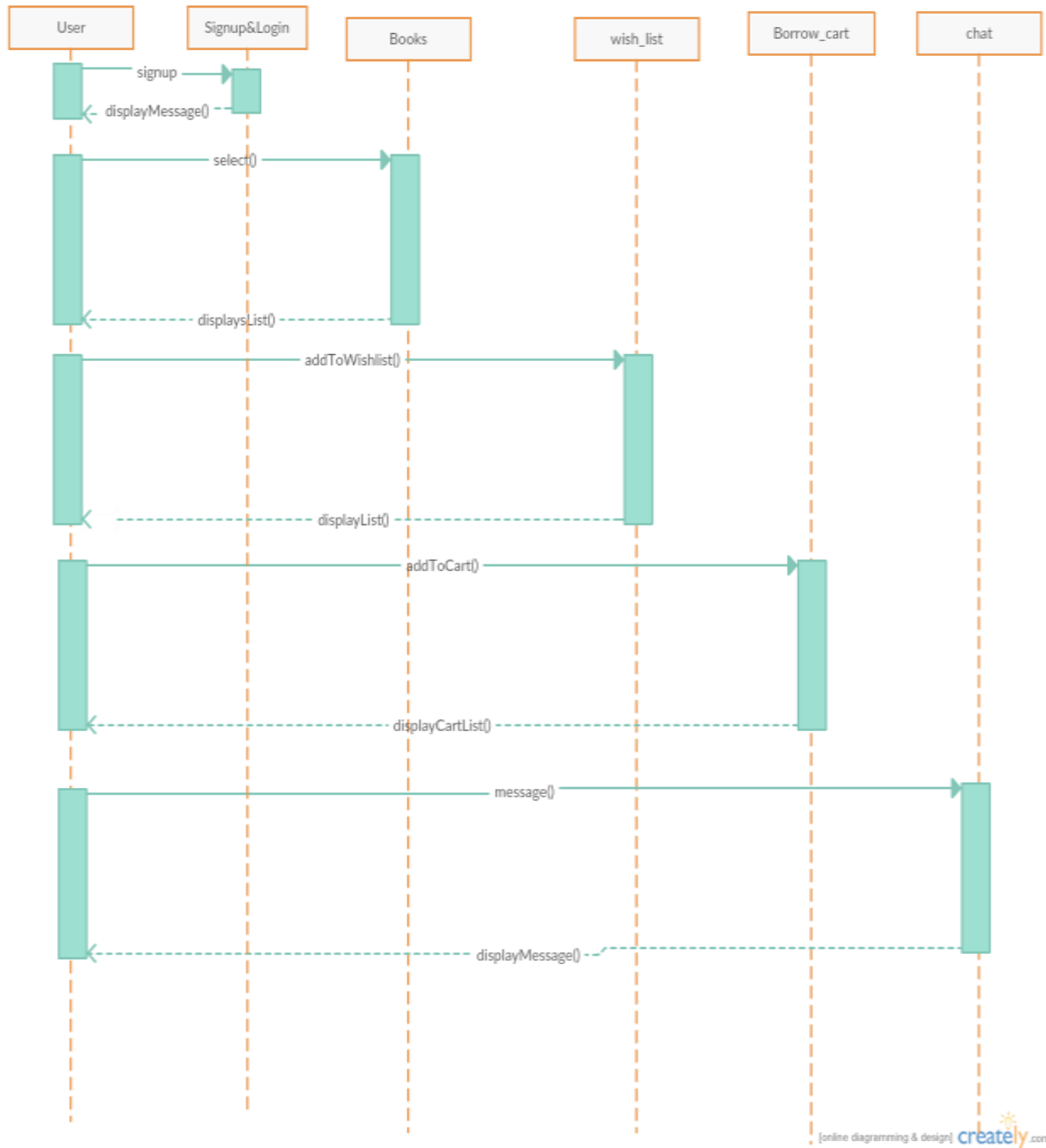
ARCHITECTURE DIAGRAM:



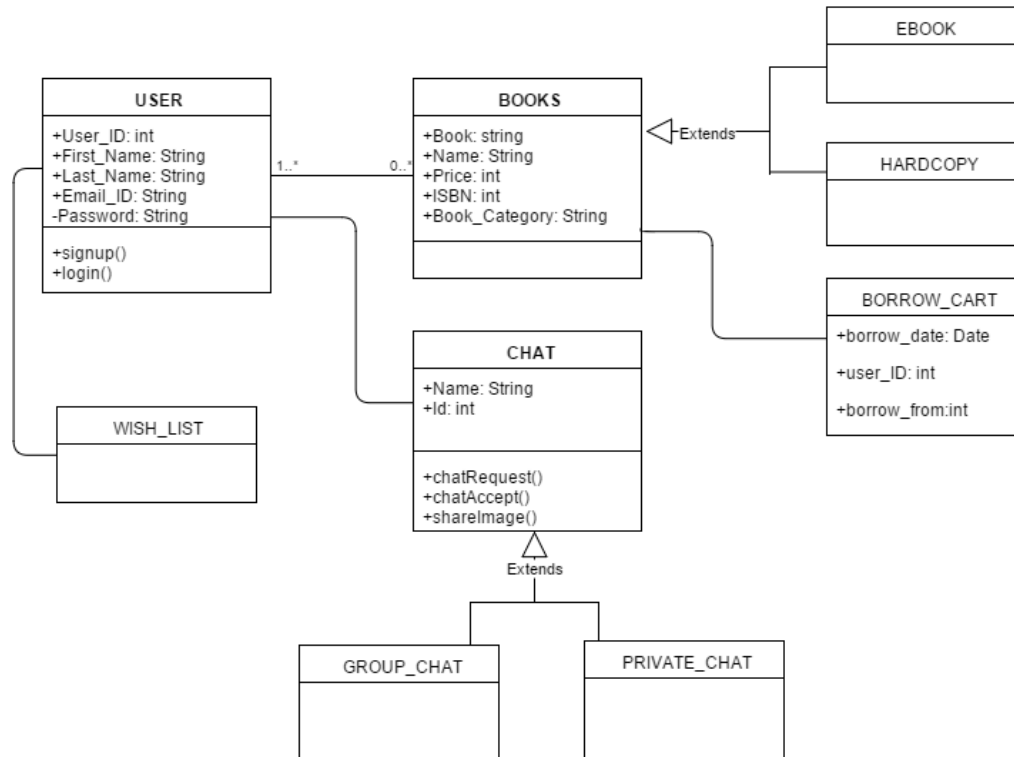
ACTIVITY DIAGRAM:



SEQUENCE DIAGRAM:



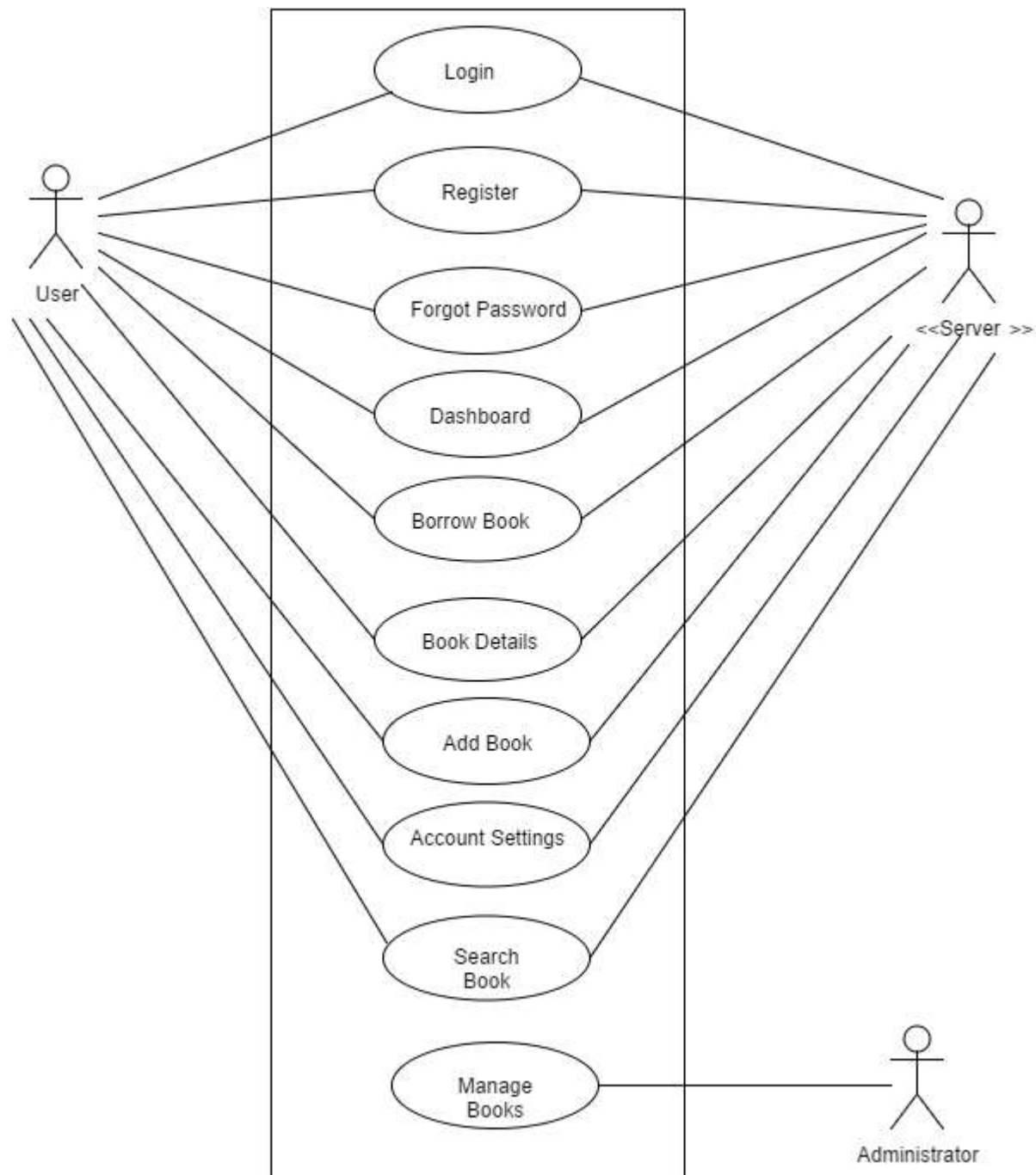
CLASS DIAGRAM:



USER STORIES:

The user as of now will login with hard-coded username and password (admin & admin), upon login the user can search for the list of books available, upon tapping on a book title book details will be shown and the user can borrow the book or add the book to his/her wishlist.

USE CASE:



SERVICE DESCRIPTION:

This service helps students of a University to borrow books from fellow students at their convenience. Create alerts when books gets available or add books to wishlist and Text to speech API that reads out the description of the book.

TESTING:

UNIT TESTING:

Unit testing has been performed using Jasmine, a framework for testing JavaScript code. Karma, Karma – Command Line Interface and Angular Mocks have been installed. PhantomJS is being used for running automated tests. Test cases for Login and Registration are written.

```
1 describe('LoginCtrl', function() {
2
3     var controller,
4         deferredLogin,
5         stateMock,
6         ionicPopupMock;
7
8     beforeEach(module('starter.controllers'));
9
10    describe('#validateLogin', function() {
11
12
13        it('should call login on userDashboard', function() {
14            expect(userDashboard.login).toHaveBeenCalled('pr3md@mail.umkc.edu', 'password1');
15        });
16
17        describe('when the login is executed,', function() {
18            it('if successful, should change state to dashboard', function() {
19
20                expect(stateMock.go).toHaveBeenCalled('user-dash.home');
21            });
22
23            it('if login is unsuccessful, should show a popup', function() {
24
25                expect(ionicPopupMock.alert).toHaveBeenCalled();
26            });
27        });
28    });
29 });
```

```
1 describe('RegistrationCtrl', function() {
2
3     var controller,
4         deferredLogin,
5         stateMock,
6         ionicPopupMock;
7
8     beforeEach(module('starter.controllers'));
9
10    describe('#register', function() {
11
12
13        describe('when the register is executed', function() {
14            it('if successful, should change state to login', function() {
15
16                expect(stateMock.go).toHaveBeenCalledWith('tab-login');
17            });
18
19            it('if registration is unsuccessful, should show a popup', function() {
20
21                expect(ionicPopupMock.alert).toHaveBeenCalled();
22            });
23        });
24    });
25 });
```

PERFORMANCE TESTING:

The image shows a web application dashboard and a Chrome DevTools Performance tab. The dashboard, located at `localhost:8100/#/dashboard/home`, has a header "Dashboard" and a greeting "Hello Swetha Chandra". It features three main buttons: "Add New Book" (red), "My Books" (blue), and "Notifications 0" (green). The Chrome DevTools Performance tab is open, showing a "Grade B" overall performance score (76) for the URL `http://localhost:8100/#/dashboard/home`. The "Make fewer HTTP requests" rule is highlighted, indicating 8 external JavaScript scripts. A list of suggestions on the left includes: "Use a Content Delivery Network (CDN)", "Avoid empty src or href", "Add Expires headers", "Compress components with gzip", "Put CSS at top", and "Put JavaScript at bottom".

Dashboard

Hello Swetha Chandra

Add New Book

My Books

Notifications 0

chrome-devtools://view/extension/winegchidppngppilnkgilnkd/yslow.html#f1

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help

Grade **B** Overall performance score 76 Ruleset applied: YSlow(V2) URL: http://localhost:8100/#/dashboard/home

ALL (23) FILTER BY: CONTENT (6) COOKIE (2) CSS (6) IMAGES (2) JAVASCRIPT (4) SERVER (6)

B Make fewer HTTP requests

F Use a Content Delivery Network (CDN)

A Avoid empty src or href

F Add Expires headers

F Compress components with gzip

A Put CSS at top

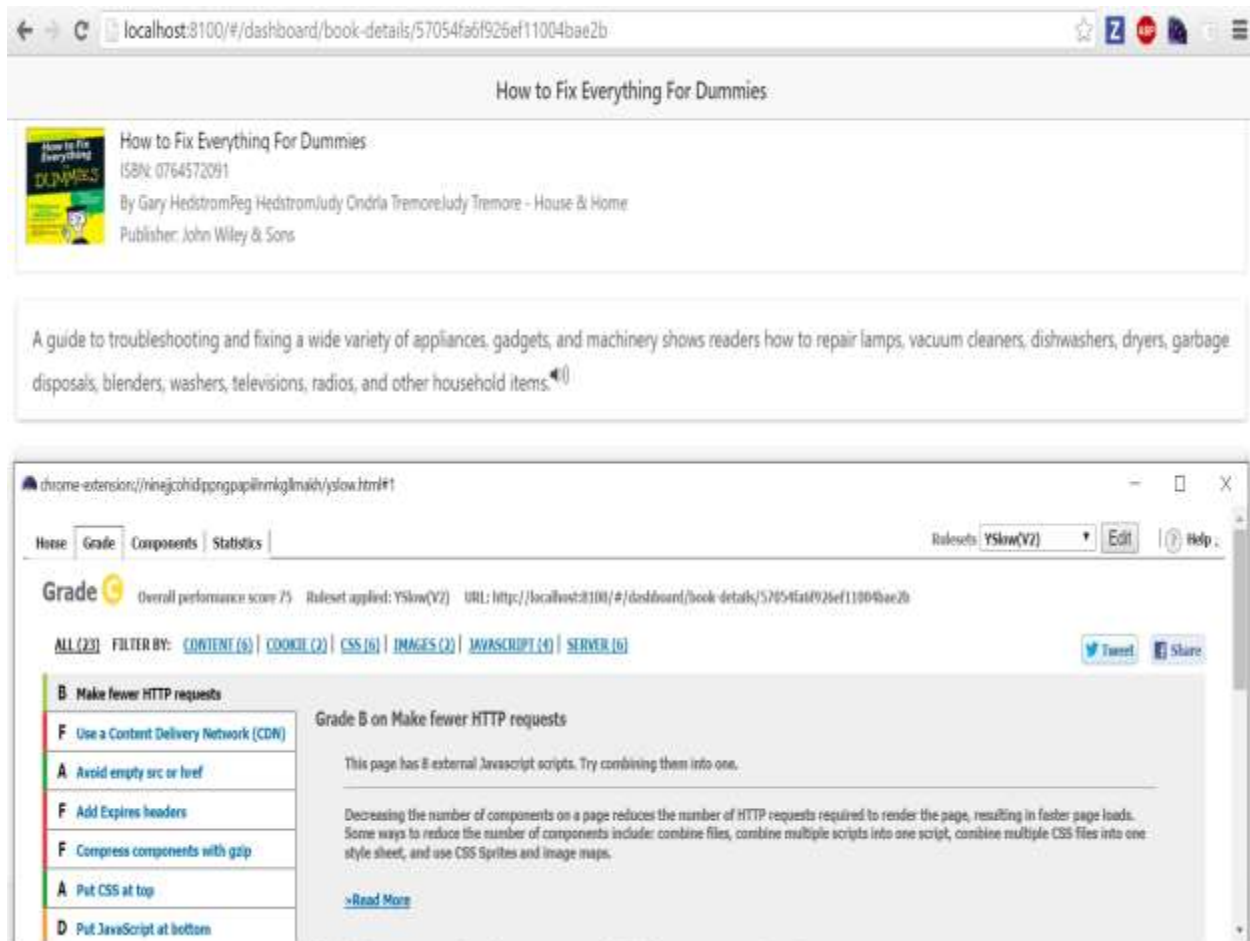
D Put JavaScript at bottom

Grade B on Make fewer HTTP requests

This page has 8 external Javascript scripts. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[Read More](#)



IMPLEMENTATION:

MOBILE CLIENT IMPLEMENTATION:

- The app is being developed using Ionic Framework. Functionality has been added to the controllers in this increment. Login, Registration and Adding a new book to the Collection features have been introduced. All these controller's functions make a **\$http** call to the backend services whenever they are triggered by an event associated with them. In turn the backend services respond to the type of call and communicate with MongoDB to fetch the information requested and send the response back to the user.
- User session information is now stored locally within the user browser. So even when the browser is refreshed, the data can be retrieved. On logout all the app data will be cleared.
- Added Phone Number Field in Registration so that SMS notifications can be sent to the user. **User can set notification preferences.**

- Redesigned the way user adds a book to the inventory. User first enters the title of the book, ISBN and then the Author and taps on Verify. Then a GET call is made to Google Books API to fetch the details of the book based on the input provided by the user. If there are matching books, then all the data like Book's author, publisher, year published, description, page count etc., are populated from the API. If the data is not found, user will need to enter all the data manually.
- **My Books View:** Books added by a user are being shown along with their Title, Author, Category and ISBN. User can remove books.
- **Show book details View:** Details are now being served from MongoDB. As soon as a book title is clicked, Book is looked up using the MongoDB unique Document Object ID and the details of the book like: Book Title, Author, ISBN, Category, Cover image (if available), Description of the book, Page Count, who is this book available with etc., are shown to the user.
- If the book is added manually, cover image of the book will be served from Amazon Product Advertising API. Amazon Product Advertising API is being used to serve the content like Price of the book, Sales Popularity Rank, Customer Reviews, Buy it from Amazon.
- User can listen to the Title of the Book and Description of the book using the Text to Speech feature. User can add a book to wishlist by clicking on Add to Wishlist button on the Book's details page. If the book is already added to the wishlist, the Wishlist button is disabled.
- **Place a Request:** Users can now place requests for books. A notification will be sent to the owner of the book if the order has been placed successfully. A SMS notification will be sent to the user about the request using Twilio SMS API.
- **Dashboard Overview:** Dashboard has been revamped so that user can view books added recently, search the books, add new books, New Notifications with Counter.
- **Forgot password feature:** When a user enter Forgot password view, they will enter their email address that is registered on the file, then a find() lookup is made for the document which contains the supplied email address, if the email is found then Phone number of the user is used and a 5 digit random number is generated and it is stored along with the email address of the user in the One Time Password (OTP) collection and an SMS is sent to the registered phone number of the user.
- Upon receiving the SMS with authentication code, the user will enter the 5 digit code in the app interface, if the authentication code supplied by the user matches the authentication code in OTP collection then the user is granted the control of the account so that they can reset the password and login.

- Gravatar is being used to display the Profile picture of the user. The email address supplied by the user during registration is retrieved and a MD5 hash is computed for the same and a GET call is sent to Gravatar API to pull the image corresponding to the hash.
- Cordova Find Contacts plugin has been used to pull contacts from the user's phone to Invite People from their contacts list. Twilio Programmable API has been used to send out SMS Invitations to try the application.
- Cordova Facebook Login plugin has been used to authenticate user to use the application by logging in with their Facebook credentials. Scope of user's information is First Name, Last Name, Profile Picture and General Information.

SERVER IMPLEMENTATION:

1. **Database:** MongoDB is being used for storing information about users, books and wish list. The database is currently hosted on mlab servers. As of now there are 2 collections defined. They are users and books. In Users collection, the document will have the fields: email, password, full name with the index as email. In books collection, the document will have the fields: Book Title, Author, ISBN and availability (which is automatically set to 1 – stating available).

2. **Service composition:** Backend services are developed with Nodejs. As of now three services are developed, they are: Processing Login, Registration and Adding new Book into the collection. The Nodejs application has been deployed to Heroku platform.

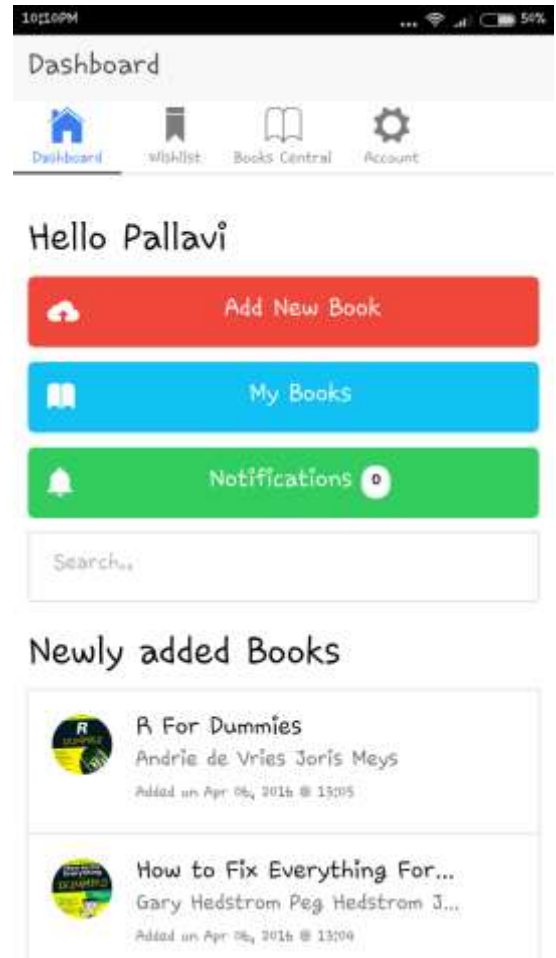
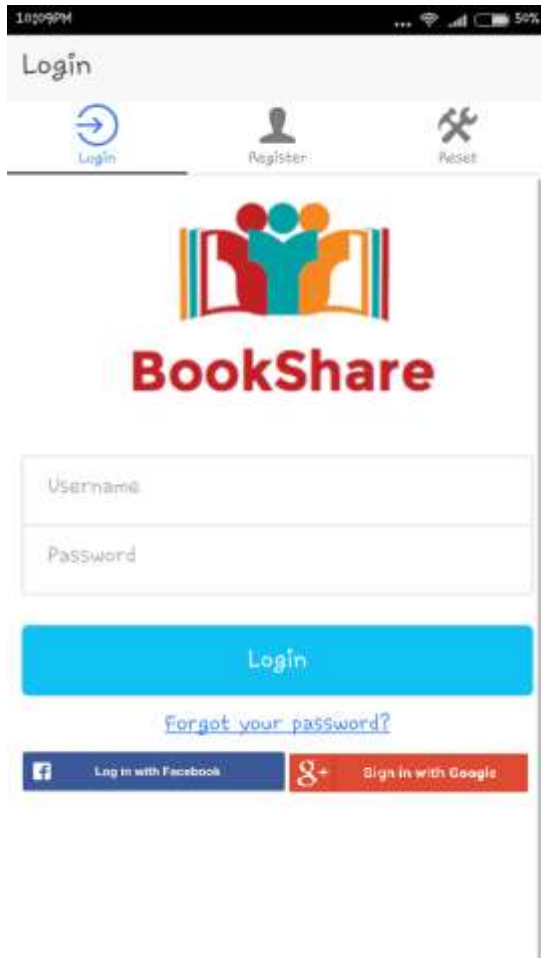
Service	Method	Service End Point	Request Payload	Response
Login	GET	https://floating-plateau-55000.herokuapp.com/bookshare/api/auth/login	Username, Password	Success – Fullname, Email
Register	POST	https://floating-plateau-55000.herokuapp.com/bookshare/api/auth/register	Email, Password, Full Name	Success – Document ID, Username, Fullname
Add Book	POST	https://floating-plateau-55000.herokuapp.com/bookshare/api/book/new	Book Title, Author, ISBN	Success – Document ID
Users	GET	https://floating-plateau-55000.herokuapp.com/bookshare/api/getUsers	-	Email, Fullname

Books	GET	https://floating-plateau-55000.herokuapp.com/bookshare/api/allbooks	-	Book Title, Author, Owner

DEPLOYMENT:

1. The Mobile application has been deployed to Ionic Framework servers and the working application can be viewed using Ionic View mobile app or an Android Device using APK file.
2. Web Portal for user administration has been deployed to DigitalOcean droplet.
3. Backend services are deployed to Amazon AWS Nodejs platform.
4. MongoDB has been hosted on mlab.
5. All the source code has been pushed to Github.

SCREENSHOTS:



10:11PM

Add New Book

Dashboard Wishlist Books Central Account

Book Title:

ISBN

Author(s)

Verify

10:12PM

Add New Book

Dashboard Wishlist Books Central Account

understanding

3642041000

christof

Verify

10:12PM
47%

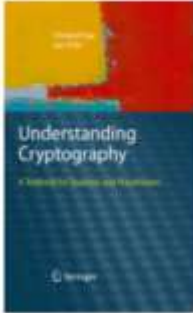
Add New Book

Dashboard
Wishlist
Books Central
Account

understanding
3642041000
christof

Verify

Understanding Cryptography
by Christof Paar



Cryptography is now ubiquitous – moving beyond the traditional environments, such as government communications and banking systems, we see cryptographic techniques realized in Web browsers, e-mail programs, cell phones.

10:12PM
47%

Add New Book

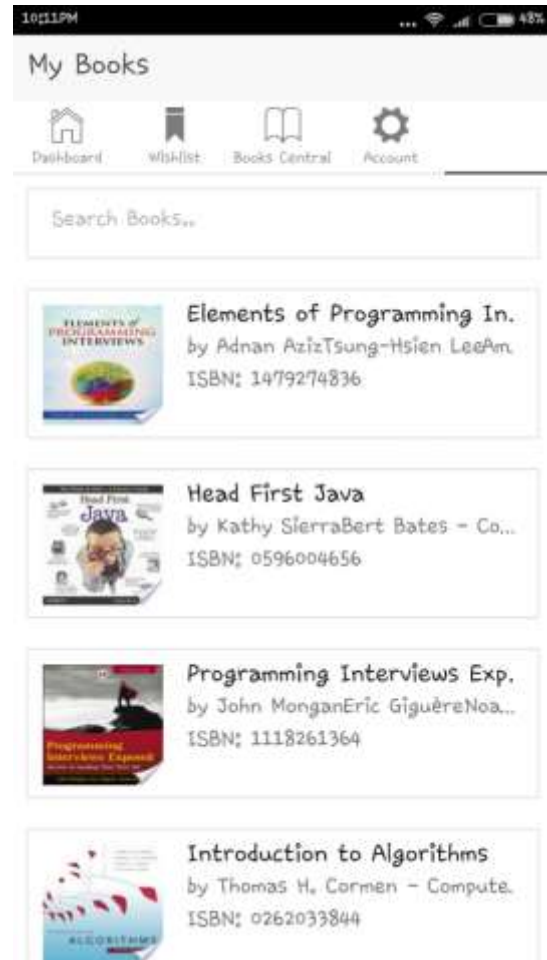
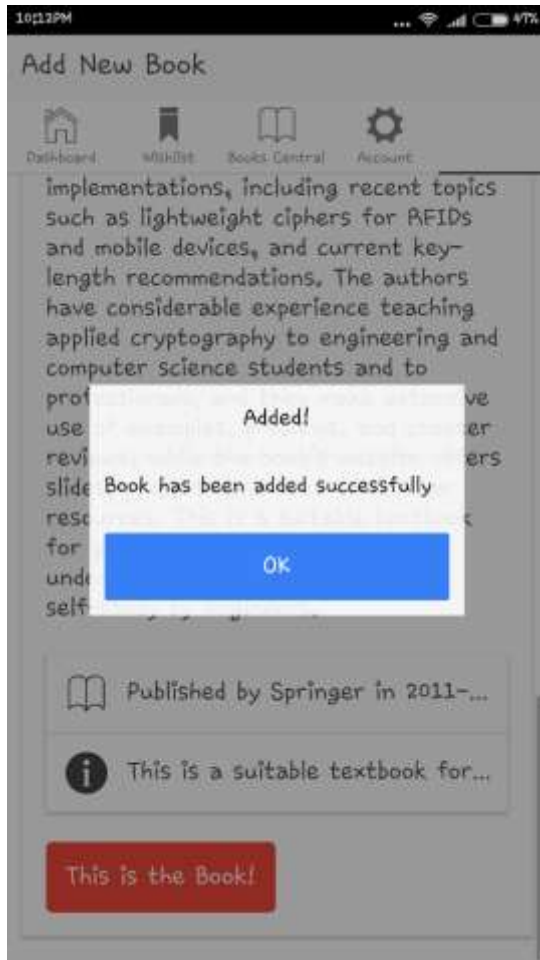
Dashboard
Wishlist
Books Central
Account

implementations, including recent topics such as lightweight ciphers for RFIDs and mobile devices, and current key-length recommendations. The authors have considerable experience teaching applied cryptography to engineering and computer science students and to professionals, and they make extensive use of examples, problems, and chapter reviews, while the book's website offers slides, projects and links to further resources. This is a suitable textbook for graduate and advanced undergraduate courses and also for self-study by engineers.

Published by Springer in 2011-...

This is a suitable textbook for...

This is the Book!



Understanding Cryptography



Understanding Cryptography
 ISBN: 3642041000
 By Christof PaarJan Pelzl - Com.
 Publisher: Springer

Cryptography is now ubiquitous – moving beyond the traditional environments, such as government communications and banking systems, we see cryptographic techniques realized in Web browsers, e-mail programs, cell phones, manufacturing systems, embedded software, smart buildings, cars, and even medical implants. Today's designers need a comprehensive understanding of applied cryptography. After an introduction to cryptography and data security, the authors explain the main techniques in modern cryptography, with chapters addressing stream ciphers, the Data Encryption Standard (DES) and 3DES, the Advanced Encryption Standard (AES), block ciphers, the RSA cryptosystem, public-key cryptosystems

Understanding Cryptography



Rating: 4 / 5
 Price: \$49.95 [a](#)
 Paperback: 372 pages
 Popularity Rank: 28110



Buy from Amazon



Add to Wishlist

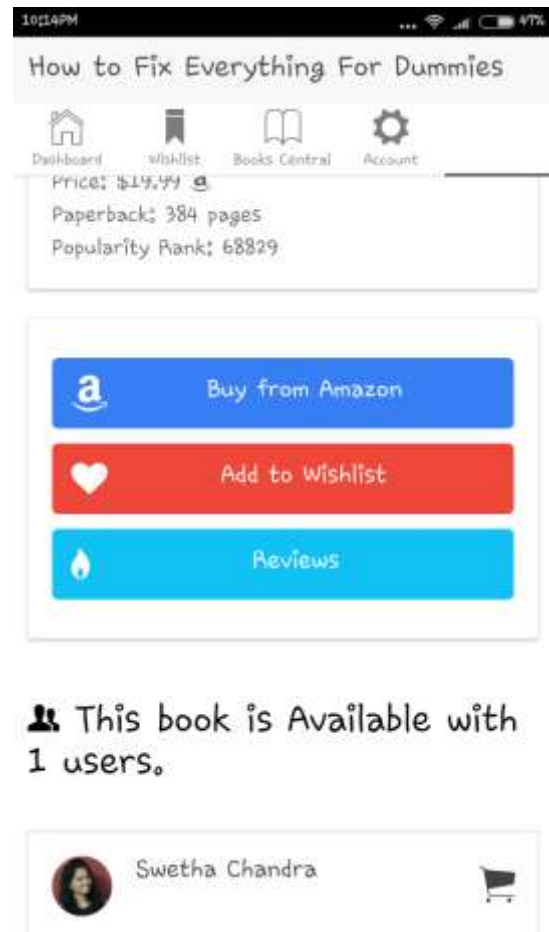
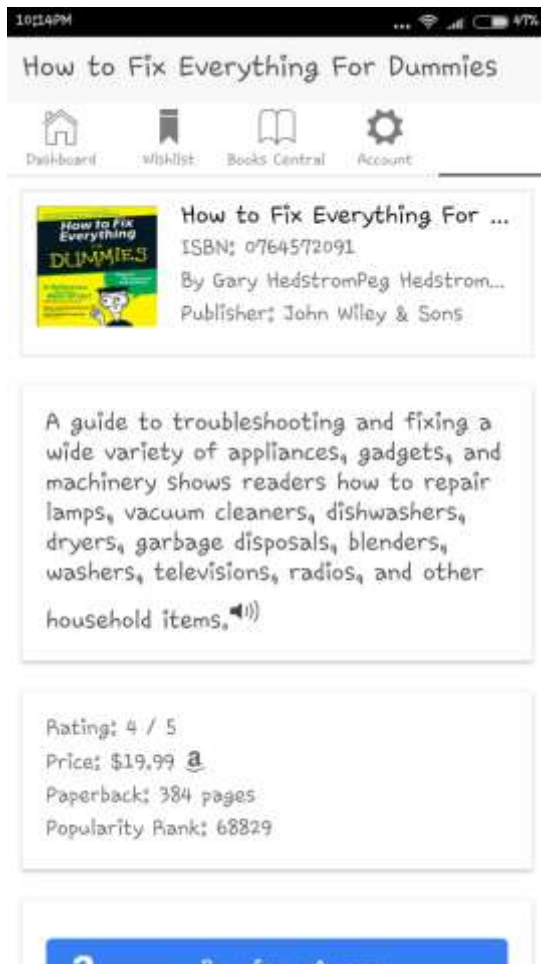


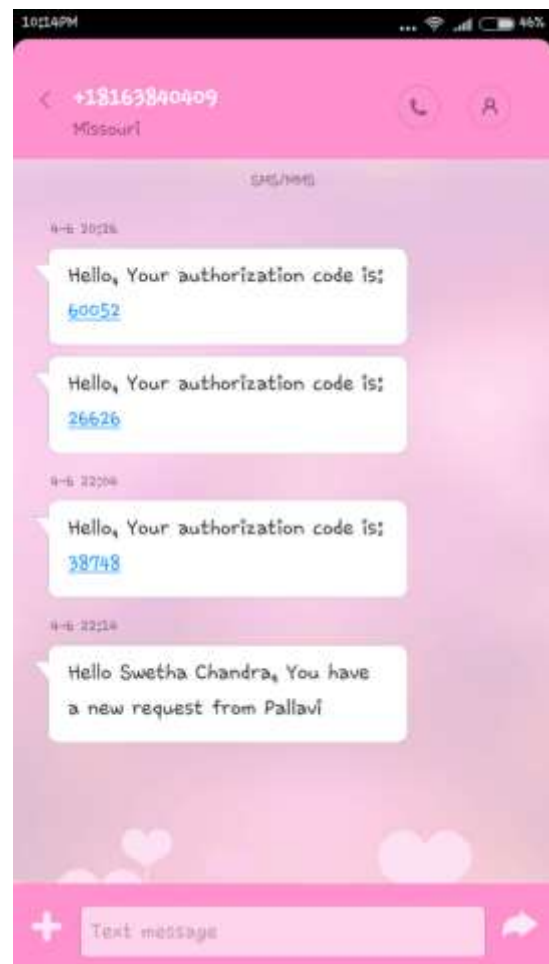
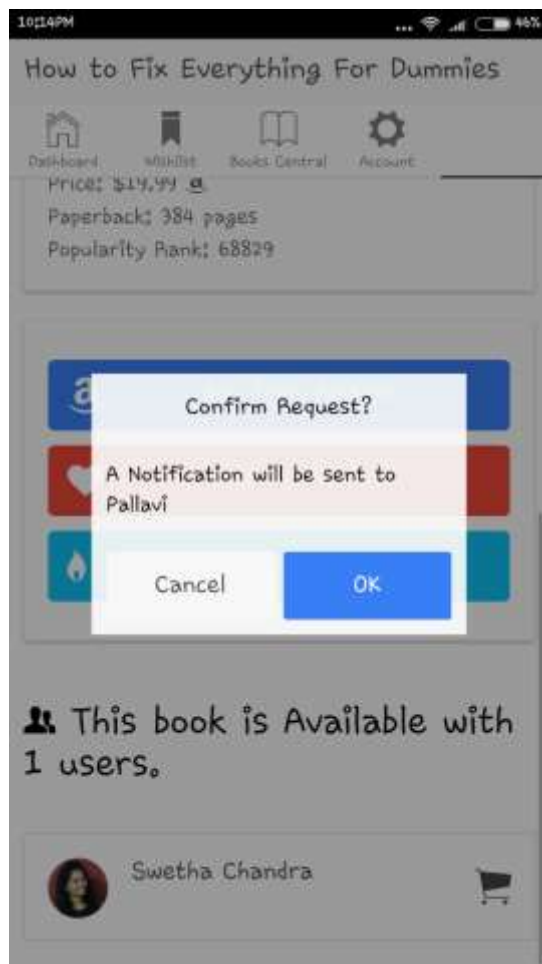
Reviews

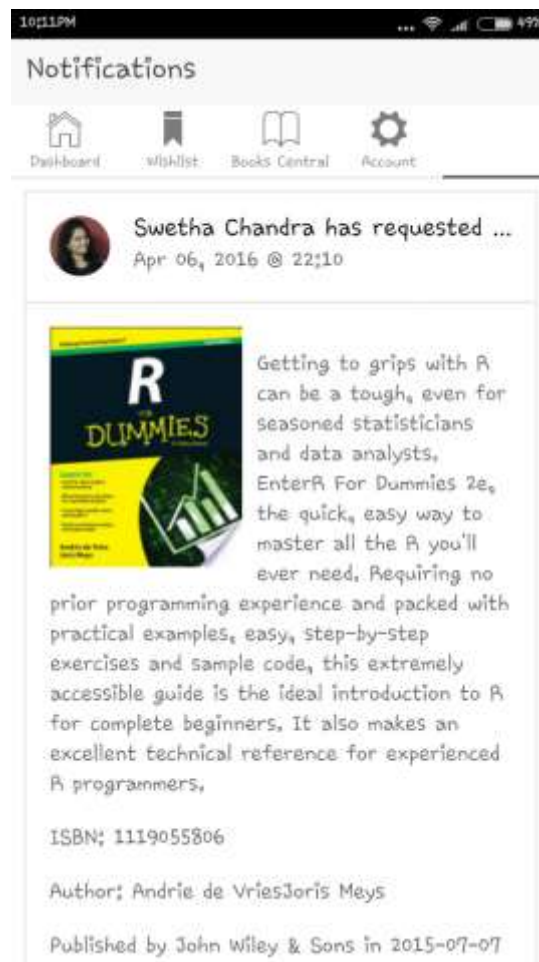
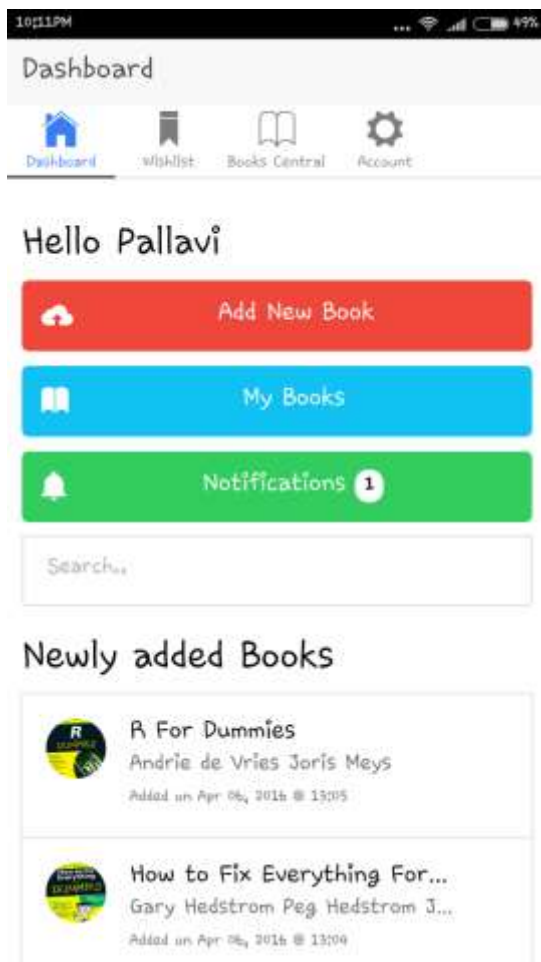
This book is Available with 1 users.

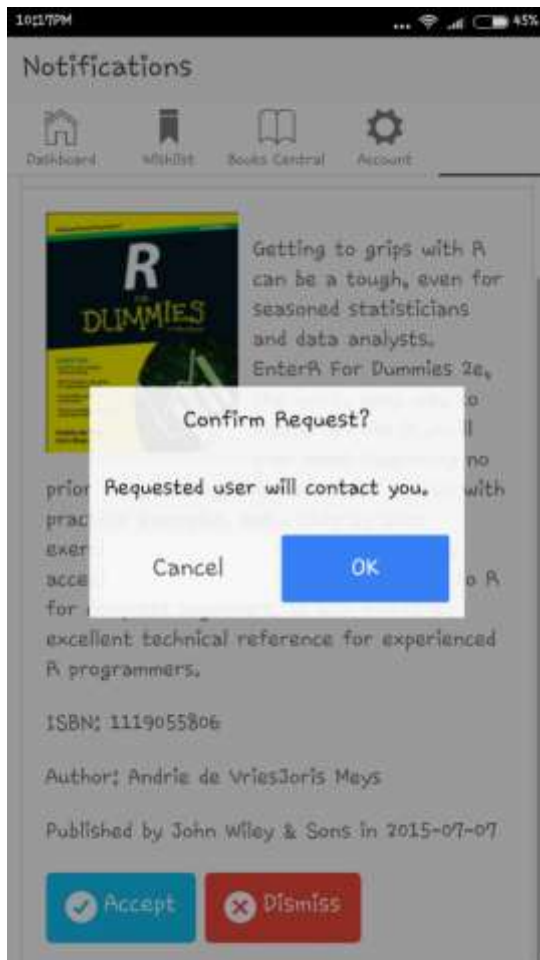


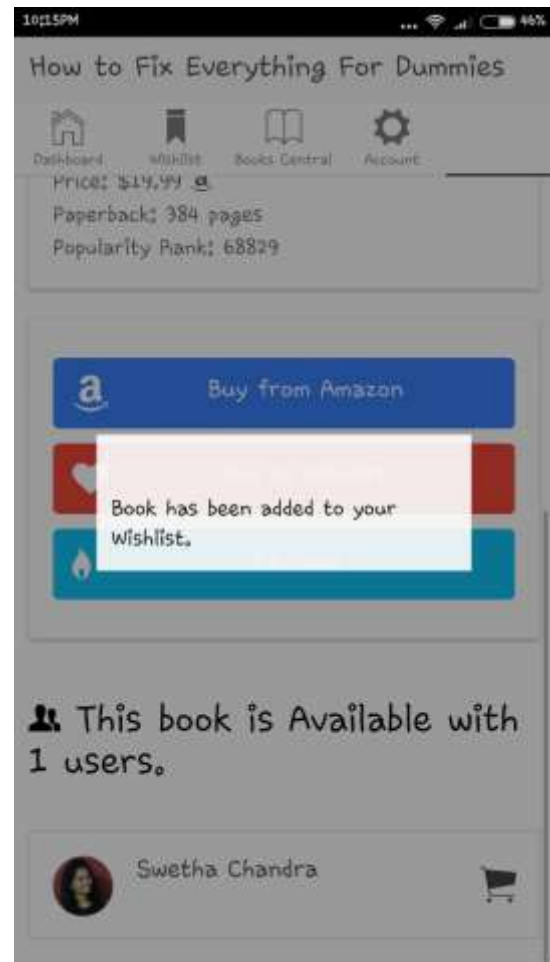
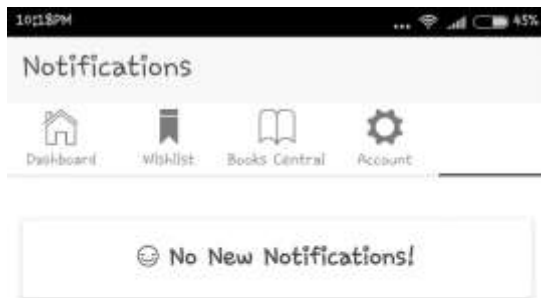
Pallavi

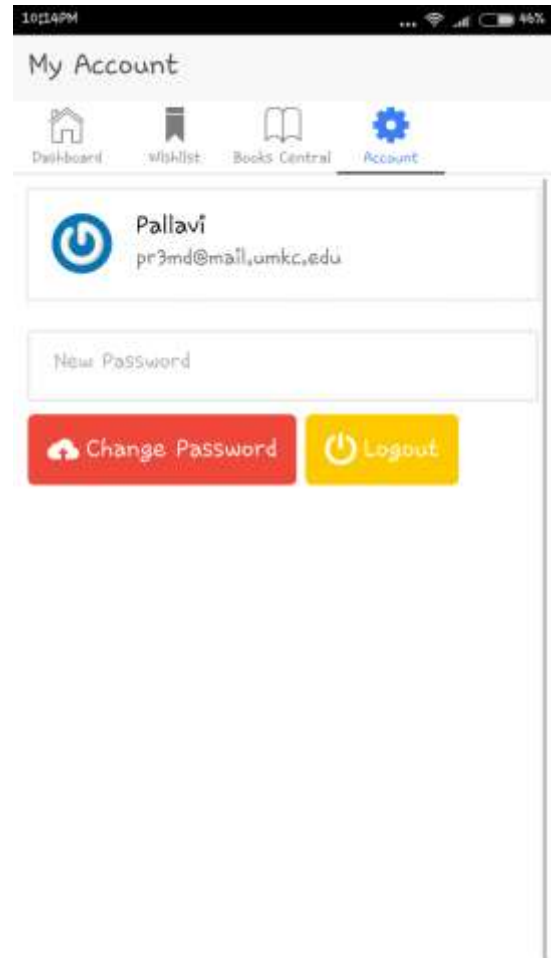
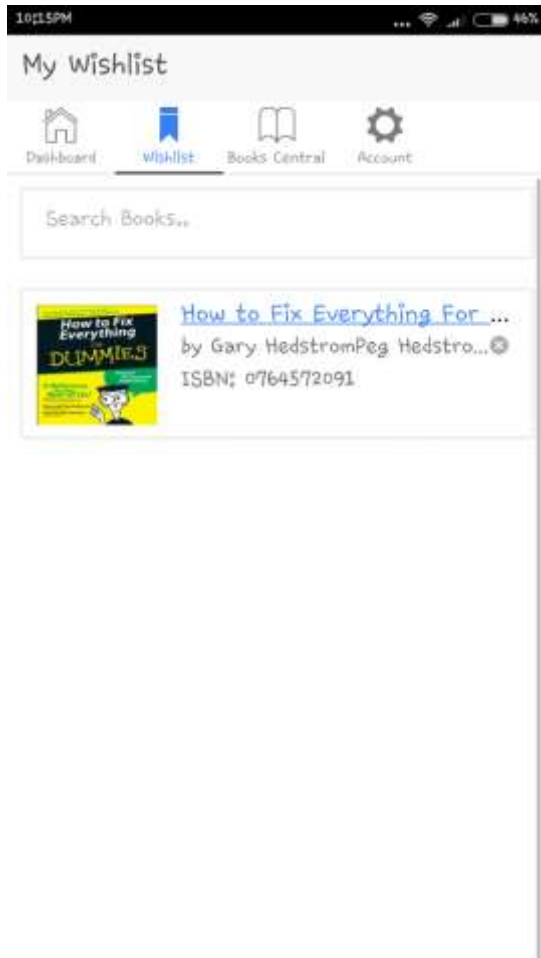




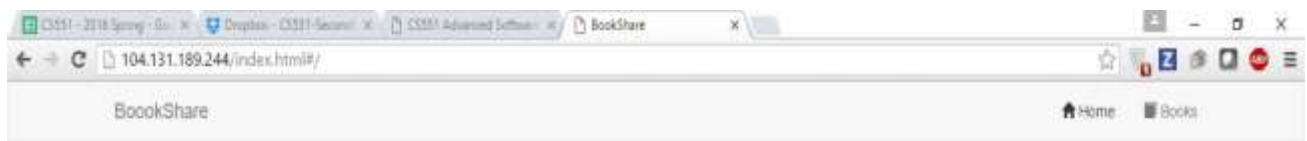




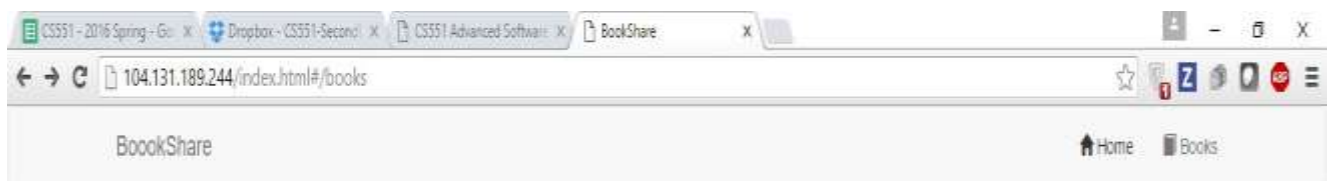




DEPLOYING WEB APPLICATION:



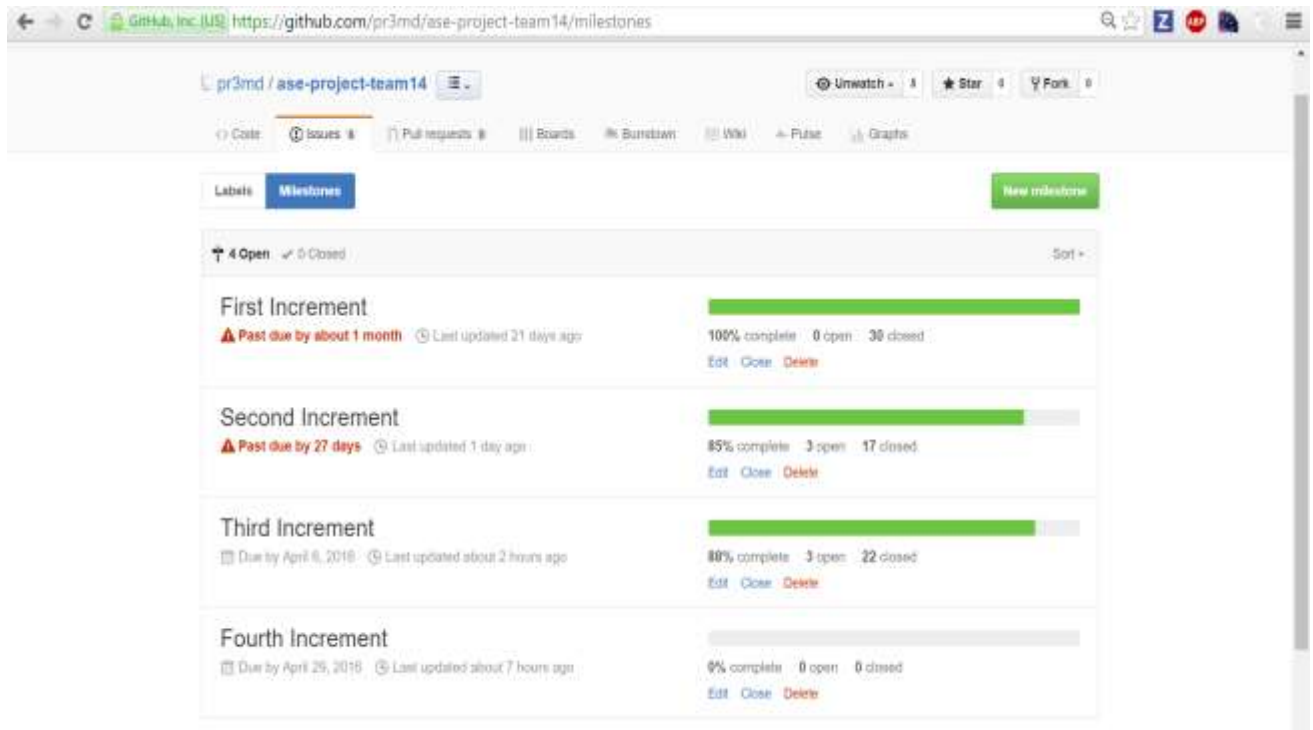
Email	Full Name
pr3md@mail.umkc.edu	Pallavi
sckqn8@mail.umkc.edu	Svettha Chandra
svethachandra9494@gmail.com	Svettha Chandra K
pallavi@gmail.com	RPallavi
mtb9@mail.umkc.edu	Latha M
rph6@mail.umkc.edu	Raja Ramya



Book Title	Author	Added
harry Potter	JK Rowling	sckqn8@mail.umkc.edu
Software Engineering-A Practitioner's Approach	Roger S. Pressman, Bruce R. Maxim	sckqn8@mail.umkc.edu

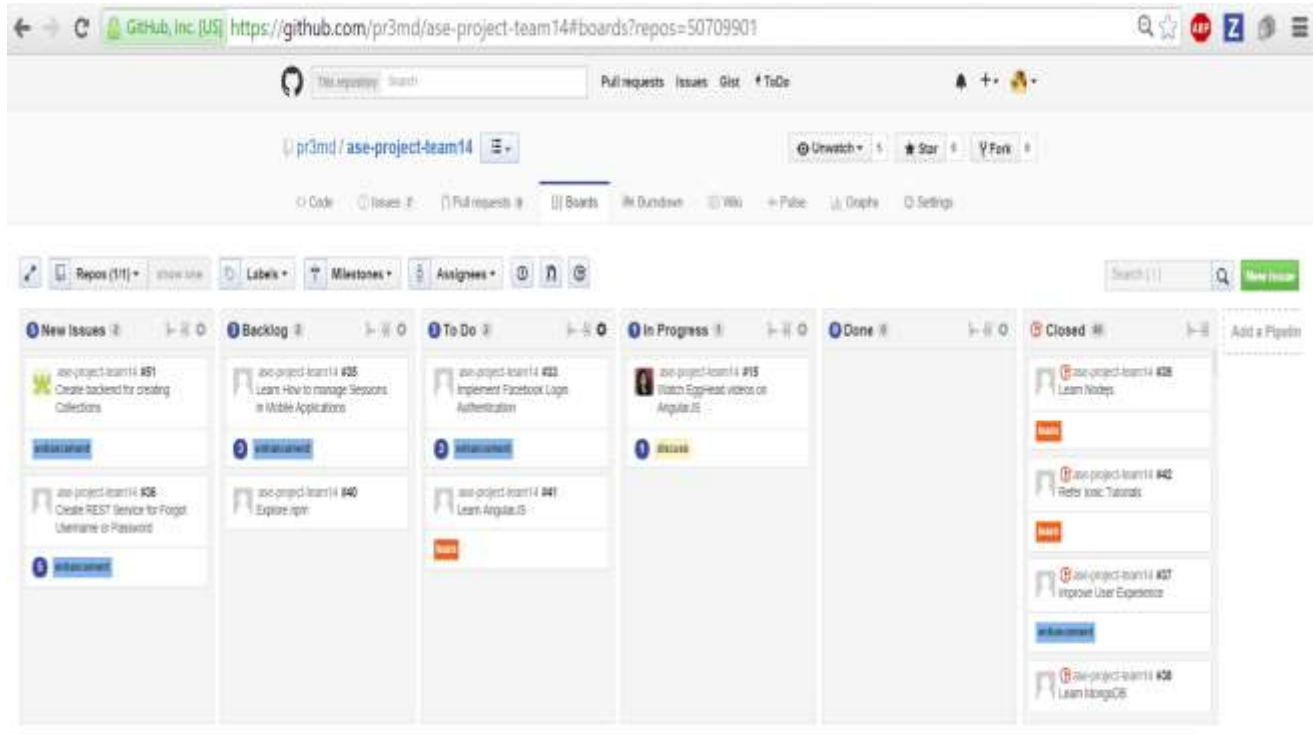
PROJECT MANAGEMENT:

PROJECT TIMELINES/MEMBERS/TASK RESPONSIBILITY:



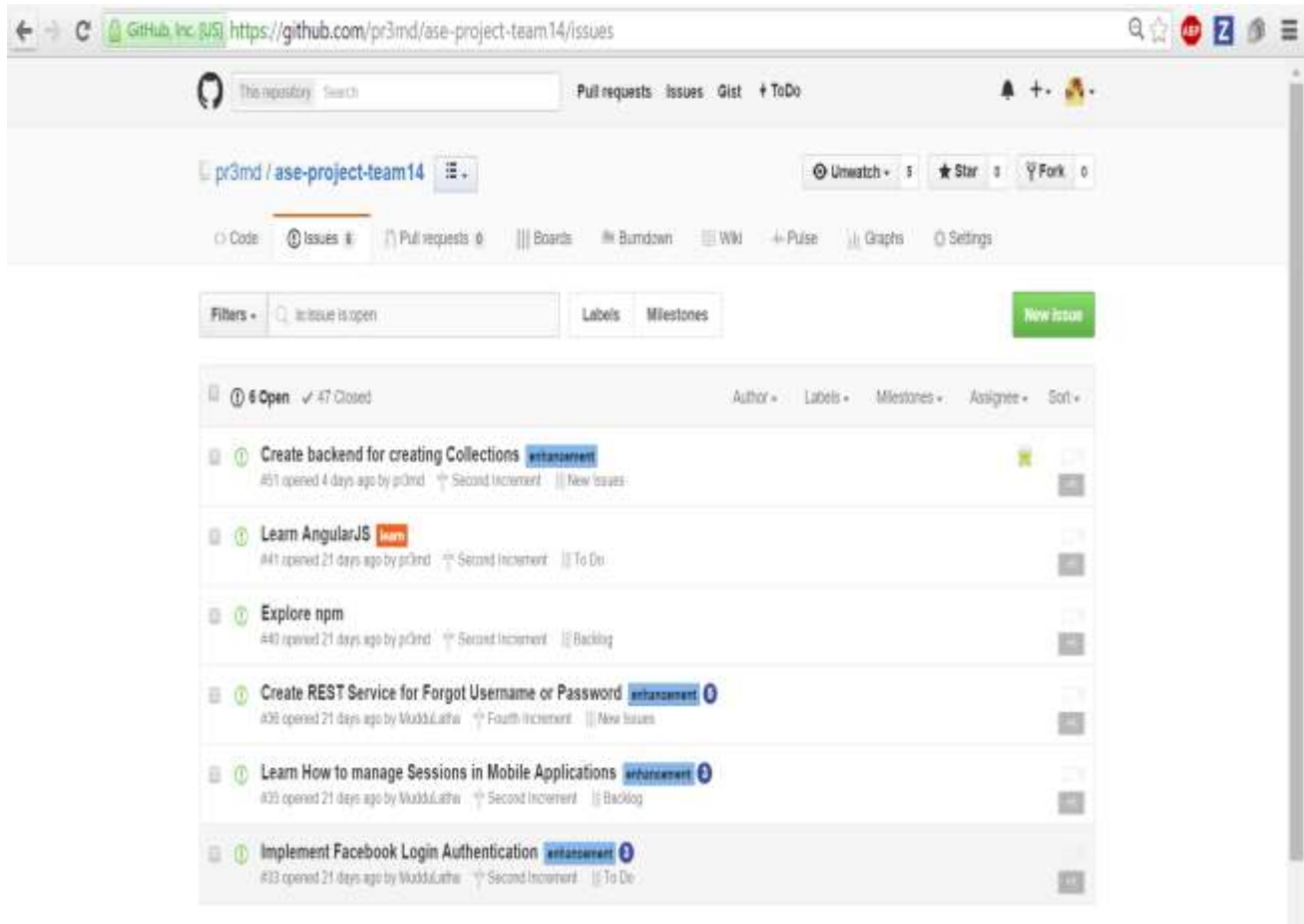
The screenshot shows the GitHub interface for the repository `pr3md/ase-project-team14`. The 'Milestones' tab is selected, displaying a list of four milestones. Each milestone includes a progress bar, completion status, and due date.

Milestone	Progress	Status	Due Date
First Increment	100% complete	0 open, 30 closed	Past due by about 1 month
Second Increment	85% complete	3 open, 17 closed	Past due by 27 days
Third Increment	80% complete	3 open, 22 closed	Due by April 6, 2018
Fourth Increment	0% complete	0 open, 0 closed	Due by April 25, 2018



The screenshot shows the GitHub interface for the repository `pr3md/ase-project-team14`, specifically the 'Boards' tab. The board is organized into columns: New Issues, Backlog, To Do, In Progress, Done, and Closed. Each column contains a list of issues with their titles and status.

Column	Issue	Status
New Issues	ase-project-team14 #31 Create backend for creating Collections	Unassigned
	ase-project-team14 #36 Create REST Service for Forgot Username or Password	Unassigned
Backlog	ase-project-team14 #35 Learn How to manage Sessions in Mobile Applications	Unassigned
	ase-project-team14 #40 Explore sign	Unassigned
To Do	ase-project-team14 #33 Implement Facebook Login Authentication	Unassigned
	ase-project-team14 #41 Learn AngularJS	Unassigned
In Progress	ase-project-team14 #15 Watch Egghead videos on AngularJS	Unassigned
Done		
Closed	ase-project-team14 #38 Learn Nodejs	Completed
	ase-project-team14 #42 Refactor Tabs	Completed
	ase-project-team14 #37 Improve User Experience	Completed
	ase-project-team14 #38 Learn MongoDB	Completed



WORK COMPLETED:

Developed RESTful services using Nodejs for handling registration, login, adding new books, removing books, requesting books, notifications, approve/decline requests, wishlist etc.,. Application communicates with the backend using **\$http** of AngularJS. Backend communicates with mlab to fetch the data requested by the application and forwards the response back to application. The application is developed using Ionic Framework.

STORIES:

In the first increment, the application interface development has started using Ionic Framework. Basic functionality like User Login, Registration, Forgot Password, Dashboard, Book Details, Search Filter, Account Settings, Notifications, Book Details, Wishlist are developed.

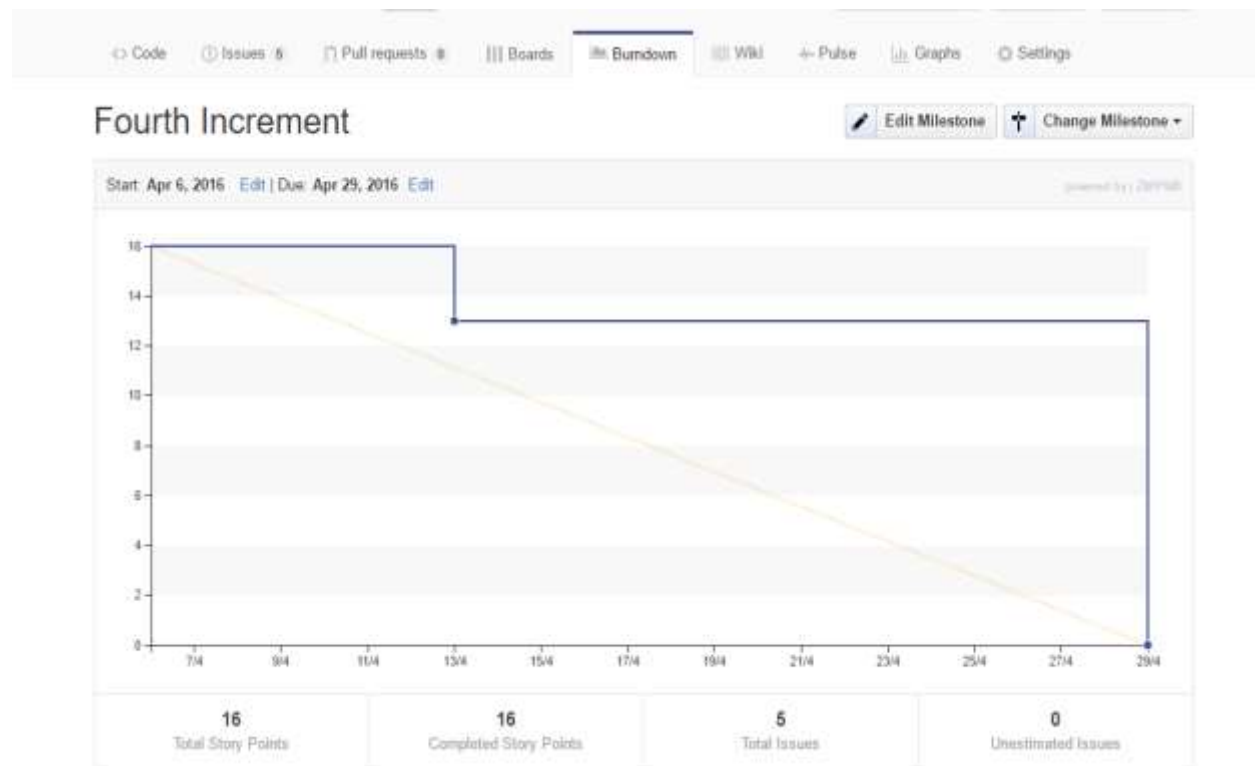
Service Design: All the user functionality has been built and the app is fully functional.

Service Implementation: All the services are being served by the NodeJS server and other APIs which allow the application to function smoothly.

LIMITATION:

The application will be able to handle limited number of users because the application's Database is hosted on mlab's sandbox.

BURNDOWN CHART:



GITHUB URL: <https://github.com/pr3md/ase-project-team14.git>

BIBLIOGRAPHY:

1. Wireframes design: <http://draw.io.com/>
2. Android studio: <http://developer.android.com/sdk/index.html>
3. ZenHub – Agile project management tool for GitHub: <https://www.zenhub.io>
4. Google Books API - <https://developers.google.com/books/docs/v1/using#intro>
5. Twilio Programmable SMS API - <https://www.twilio.com/docs/api>
6. Amazon Product Advertising API - <https://affiliate-program.amazon.com/gp/advertising/api/detail/main.html>
7. Gravatar API - <https://en.gravatar.com/site/implement>
8. Cordova TTS Plugin
9. Cordova Facebook plugin
10. Cordova Invite contacts.

