



BookShare

TEAM-14

TEAM MEMBERS:

SWETHA CHANDRA KARROTI-21

PALLAVI RAMINENI-49

RAJARAMYA JANAGAMA-17

LATHA MUDDU-29

Table of contents:

Project Deployment.....	3
Introduction.....	3
Motivation.....	3
A section on how to use the system.....	3
Error recognition and handling.....	4
A list of known bugs and deficiencies.....	4
Project management.....	5
Project management report.....	5
Final project evaluation.....	5
Project Proposal.....	6
First Increment Report.....	10
Second Increment Report.....	20
Third Increment Report.....	32
Fourth Increment Report.....	45

PROJECT DEPLOYMENT:

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.

MOTIVATION:

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.

A SECTION ON HOW TO USE THE SYSTEM:

The application can be installed on Android and iOS devices. An APK file is made available for Android devices, whereas iOS devices can use the app using Ionic View app by entering the App ID: C303DCF6. The user of the application can register their account or use their Facebook account to start using the application.

ERROR RECOGNITION AND HANDLING:

Pagination of Books Central, so that 10 books can be displayed per page. This has been handled using jQuery's lazy loading technique which loads items on scrolling to the end of the page.

A LIST OF KNOWN BUGS AND DEFICIENCIES:

1. Text to Speech Plugin is not available for iOS devices.
2. Logged in user's cache is not handled in an efficient way.
3. Requests sent to backend service are not secured by an authentication mechanism.
4. Data sent to backend services are not sent over HTTPS.

PROJECT MANAGEMENT:

PROJECT MANAGEMENT REPORT:

Pallavi Ramineni- 25%

Swetha Chandra Karroti- 25%

Rajaramya Janagama- 25%

Latha Muddu- 25%

FINAL PROJECT EVALUATION:

Agile process model was very useful. Though there were some backlogs in completing the planned features, we successfully managed to complete them in the next increments. Yes we did stick up to the project plan schedule.

Github URL: https://github.com/SCE-UMKC/ASESP16_BookShare_14.git

Youtube URL: <https://www.youtube.com/watch?v=3YTdOQ--KqM>

ADVANCED SOFTWARE ENGINEERING

PROJECT PROPOSAL

Dr.Yugyung Lee

TEAM NUMBER: 14

TEAM MEMBERS: SWETHA CHANDRA KARROTI-21

PALLAVI RAMINENI-49

MUDDU LATHA-29

JANAGAMA RAJARAMYA-17

PROJECT TITLE: BOOK PEERS

PROJECT GOALS AND OBJECTIVES:

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.

MOTIVATION:

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.

SIGNIFICANCE:

Book Sharing is an effective way for reducing and simplifying the searching time of the users. It provides more convenience for users and reduces their travelling issues. Also this increases convenience for users and provides them with many options.

SYSTEM FEATURES:

1. First each user must register in the login page of the application by providing the following fields:

- User Name
- Password
- First Name
- Last Name
- Gender
- School

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
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.

PROPOSED DEVELOPMENT:

Ionic Framework, Android SDK, Angular JS, MySQL Database instance on Amazon RDS.

BACKUP PROJECT:

PROJECT TITLE: CAR POOLING

Car Pool is mainly used to share the journey, which reduces cost of fuel and tolls. Car Pooling is also Eco-friendly. First the person with the car registers himself in the application by providing his personal, car, route of the trip and bank details. Now, another registered user requests for a ride if he wants to go in the same route. Fare is computed depending on Type of car, Distance, Duration of the trip.

BIBILOGRAPHY:

<http://nevonprojects.com/>

<http://www.elprocus.com/top-software-engineering-projects-for-it-and-cse-students-in-2014/>

BOOK SHARE

TEAM-14

TEAM MEMBERS:

SWETHA CHANDRA KARROTI-21

PALLAVI RAMINENI-49

RAJARAMYA JANAGAMA-17

LATHA MUDDU-29

PROJECT GOAL AND OBJECTIVES

OVERALL GOAL:

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.

OBJECTIVES:

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
- First Name
- Last Name
- Gender
- School

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

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

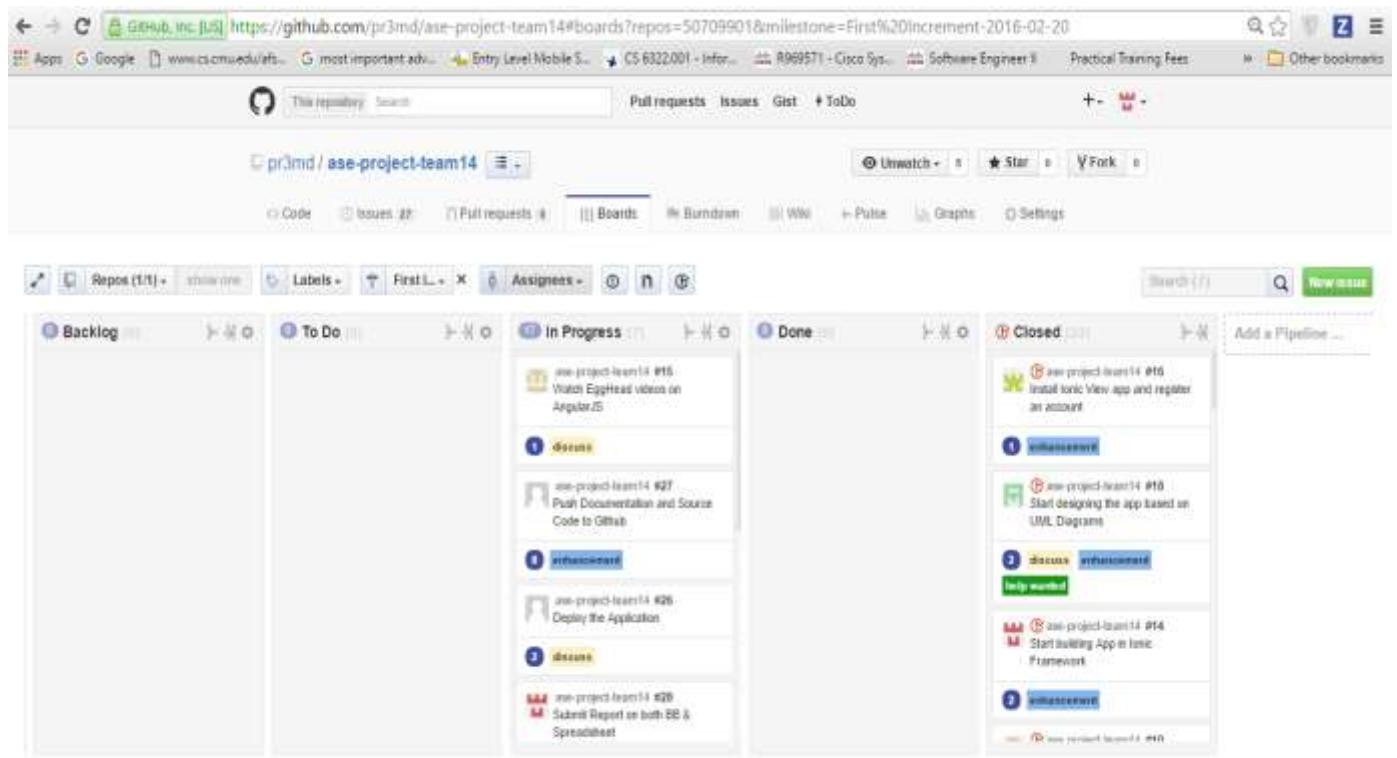
- Price
8. Now the borrower selects his preferences and chooses one among the available users.
 9. Then a notification is sent to the selected user and he can accept or deny the request.
 10. If the request is accepted then the user provides the location to collect the book.

SIGNIFICANCE:

Book Sharing is an effective way for reducing and simplifying the searching time of the users. It provides more convenience for users and reduces their travelling issues. Also this increases convenience for users and provides them with many options.

PROJECT PLAN:

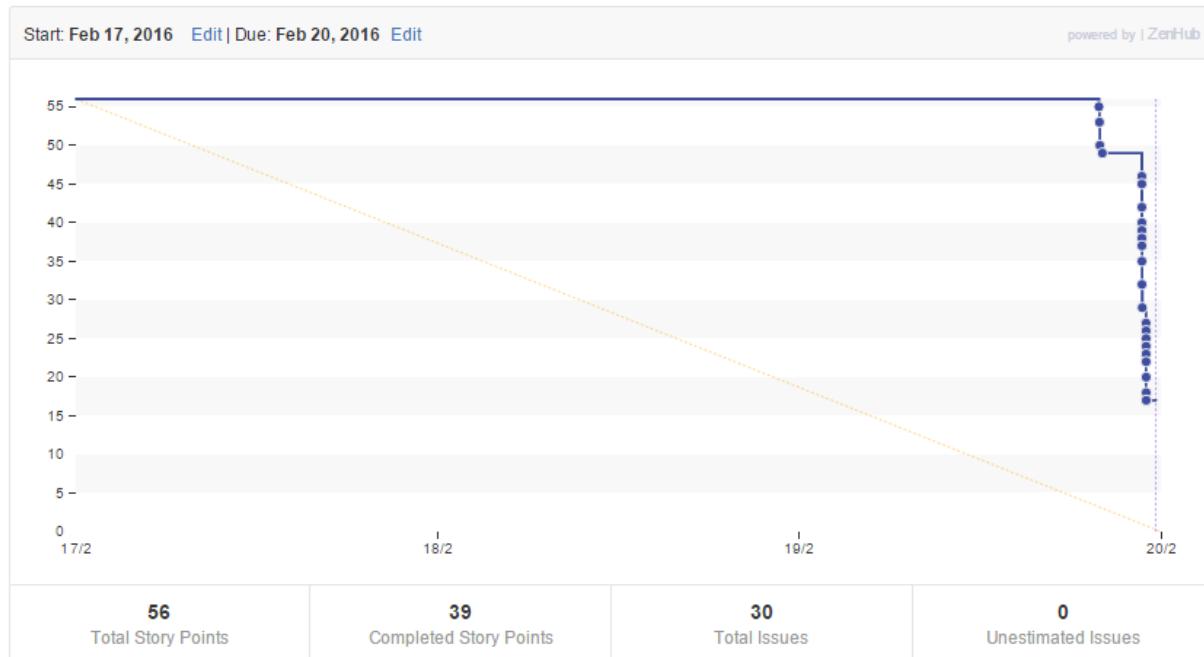
Zenhub Screenshot:



Burn Down Chart:

First Increment

 Edit Milestone  Change Milestone ▾



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 and Account Settings are developed.

Service Design: The base design of this application concentrates on User Interface and Application architecture. All the user screens are designed and backend functionality will be built in further increments.

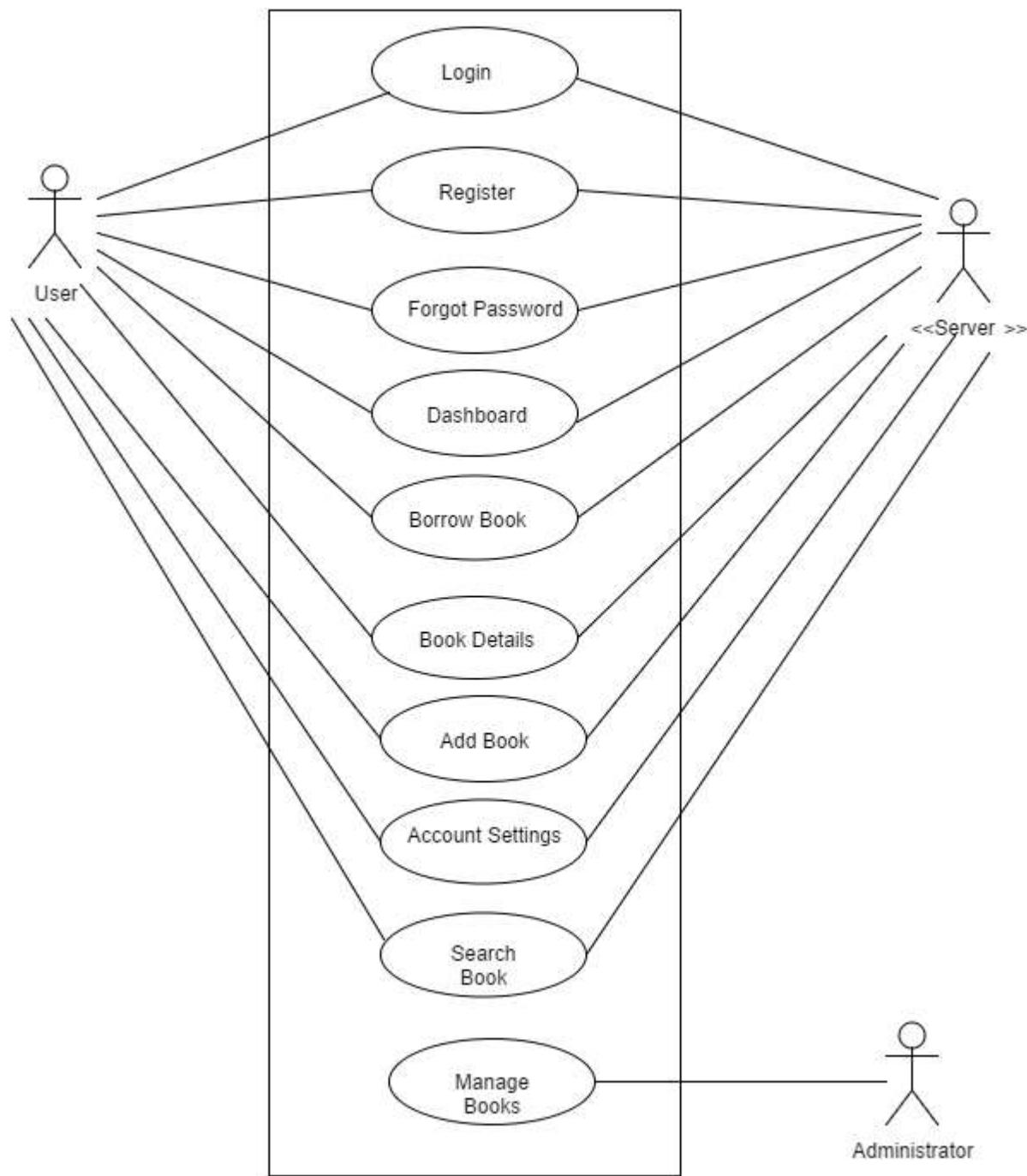
Service Implementation: At this stage, fake data is being used in **services.js** file to fill the user interface, once backend services are established data will be retrieved from the Database.

Project Timelines/Members/Task Responsibility:

<https://github.com/pr3md/ase-project-team14#boards?repos=50709901&milestone=First%20Increment-2016-02-20>

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.

Project First Increment Report

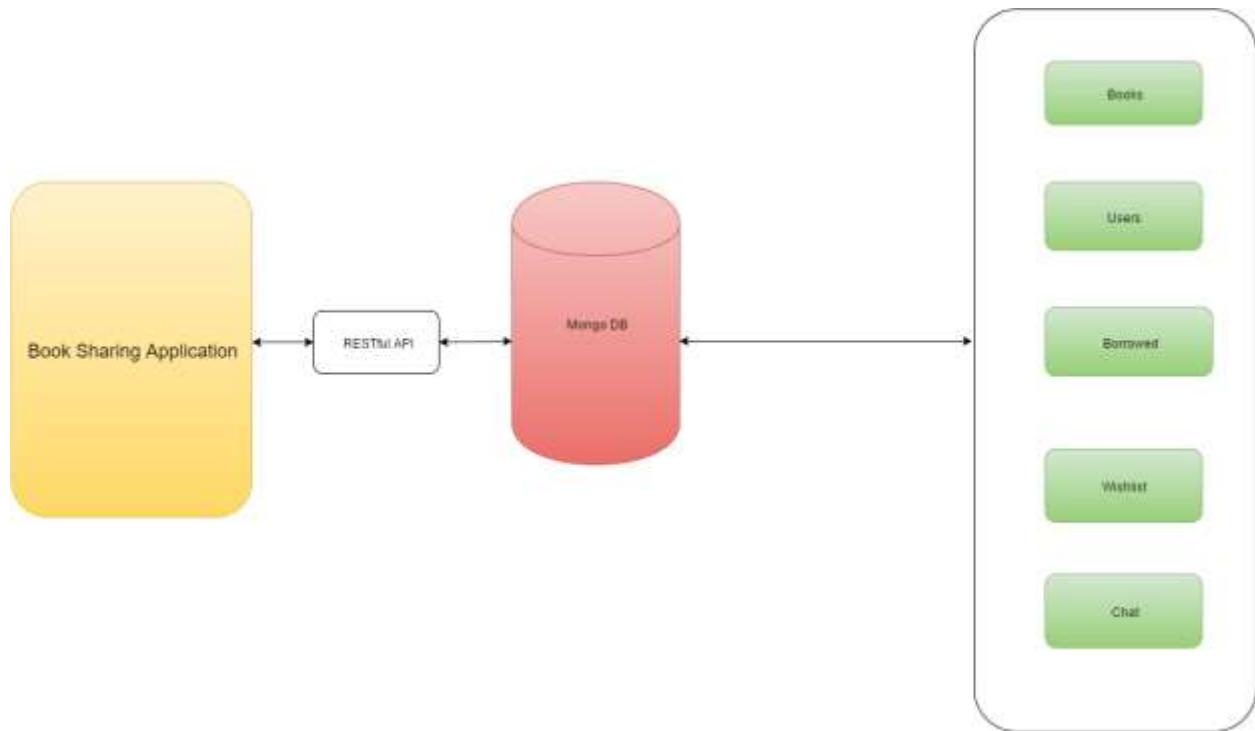
Detail Design:Wireframes



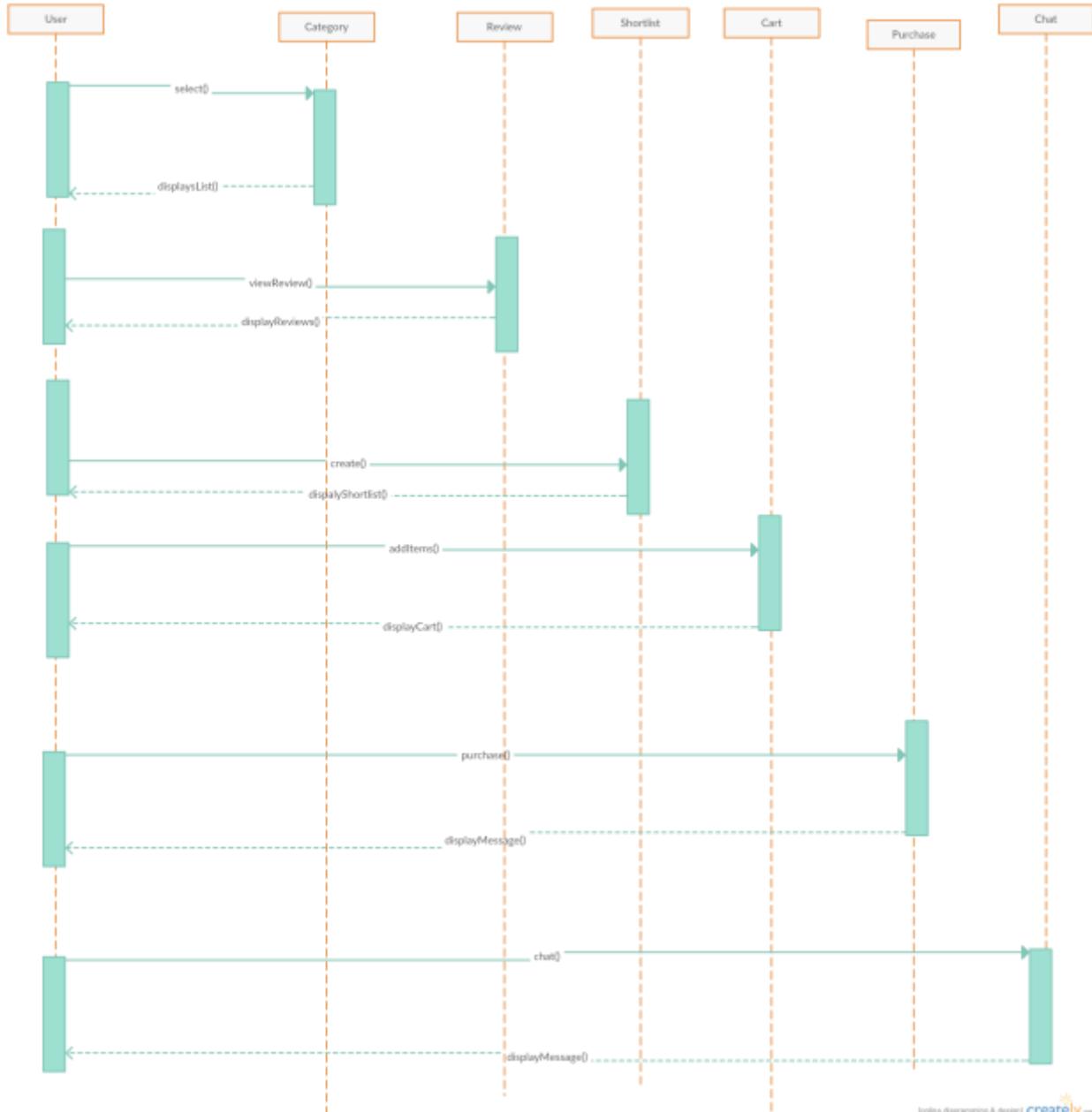




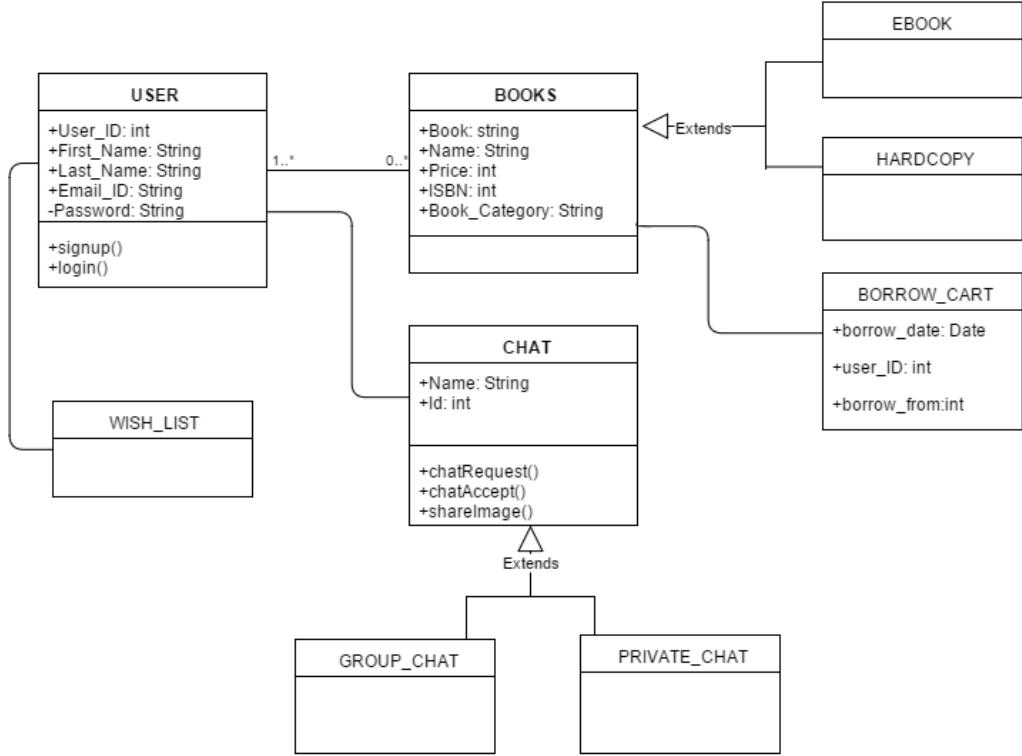
ARCHITECHTURE DIAGRAM



SEQUENCE DIAGRAM



CLASS DIAGRAM



Used existing services / API:

Application was developed using Ionic Framework on Jet Brains WebStorm IDE. No existing API's are being used in the development of the application, REST services are being developed and deployed on our own servers.

Implementation and Deployment:

The first increment of the Application was deployed to Ionic servers which can be viewed on Ionic View application. The application backend is being built with **MEAN** (MongoDB, ExpressJS, AngularJS and Nodejs) stack and it is deployed at 104.236.229.136:3000. The application makes RESTful calls with this server.

The image shows a mobile application interface with two main sections: a sidebar on the left and a main content area on the right.

Left Sidebar (My apps):

- Header: "My apps" with three vertical dots for more options.
- Item: "bookShare" with a cloud icon.
- Details:
 - ID: C303DCFG
 - SIZE: 2.3 MB
 - MODIFIED: FEB 19, 2016 2:49:02 PM

Right Main Area (Login Screen):

- Header: "Login" with three icons: a blue arrow, a person, and a wrench.
- Text: "Welcome to BookShare".
- Form fields:
 - "Username" (empty input field)
 - "Password" (empty input field)
- Buttons:
 - A large blue button labeled "Login".
 - A link below it: "Forgot your username/password?".

8:53PM

8:53PM

Login

 Login

 Register

 Reset

Welcome to BookShare

admin

...

 Login

[Forgot your username/password?](#)

Invalid login

Login

 Register

 Reset

Password

you@somedomain.com

Full Name

UMKC

 Signup

8:53PM 8:54PM

Reset Account Dashboard

Login Register Reset Dashboard Wishlist Account

Enter your Email Address

Email

Reset Account

Search...

Available Books

	Headfirst Java Kathy Sierra & Bert Bates
	Cracking the Coding Interview Gayle Laakmann McDowell
	Steve Jobs Walter Isaacson
	Headfirst JavaScript Eric T. Freeman & Elisabeth Robins
	Beginning Software Engineering Rod Stevens

8:54PM 80% 10:35PM 66%

Dashboard Headfirst JavaScript

Dashboard Wishlist Account Dashboard Wishlist Account

Crack

Available Books

 Cracking the Coding Interview..
Gayle Laakmann McDowell

 Author: Eric T. Freeman & Elisabeth Robson
ISBN: 144934013X

Add to Wishlist

This book is available for borrowing from Latha.

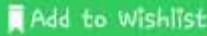
Borrow from Latha

10:34PM ... WiFi 66% 8:54PM ... WiFi 80%

← Beginning Software Engineering Your Wishlist

 Dashboard  Wishlist  Account  Dashboard  Wishlist  Account

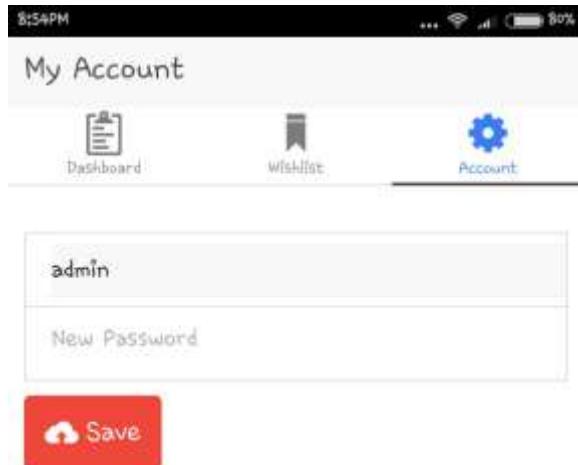

Author: Rod Stevens
ISBN: 1118969146



This book is available for borrowing from Pallavi.



Book name 1
Book name 2
Book name 3
Book name 4
Book name 5
Book name 6
Book name 7
Book name 8



Project Management:

Work Completed: Base user interface of the application has been developed and deployed on Ionic Servers which can be viewed using Ionic View application.

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

Second Increment: In the next increment, Application backend will be developed.

<https://github.com/pr3md/ase-project-team14#boards?repos=50709901&milestone=Second%20Increment-2016-03-11>

BIBILOGRAPHY:

<http://nevonprojects.com/>

<http://www.elprocus.com/top-software-engineering-projects-for-it-and-cse-students-in-2014/>



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

EXISTING SERVICES/API:

- Application was developed using Ionic Hybrid Mobile apps Framework on Jet Brains WebStorm IDE.
- MongoDB is used for storing the data, MongoDB is being hosted on mlab.
- Backend Services are built using Nodejs, Nodejs is hosted on Heroku platform.
- 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.

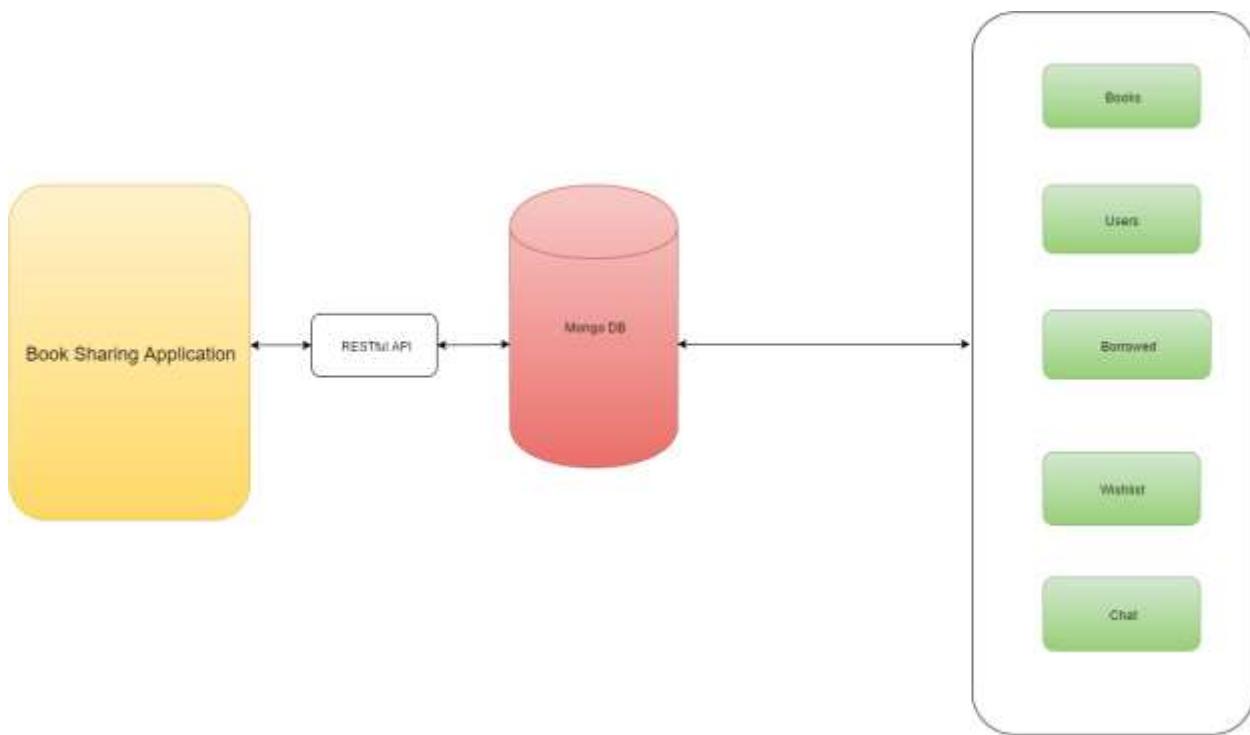
DETAIL DESIGN OF FEATURES (Using Tools):

WIREFRAMES:

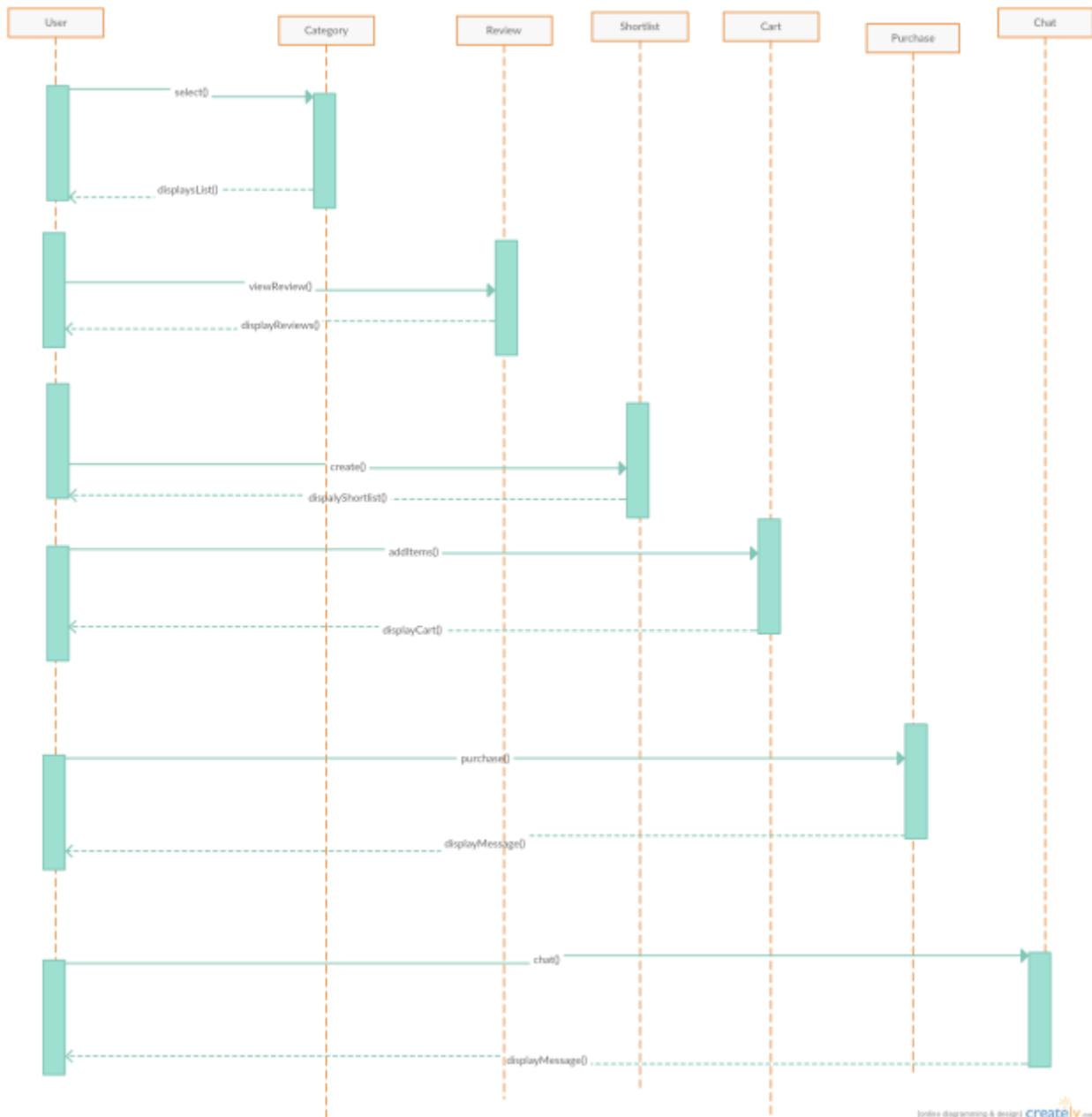




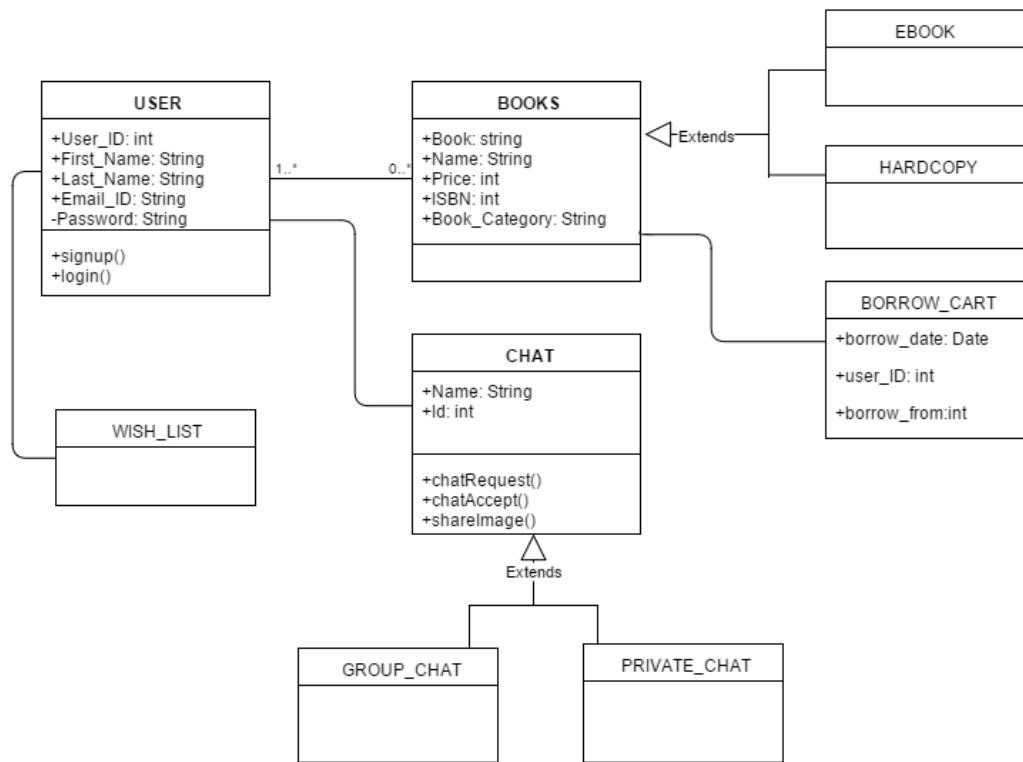
ARCHITECTURE DIAGRAM:

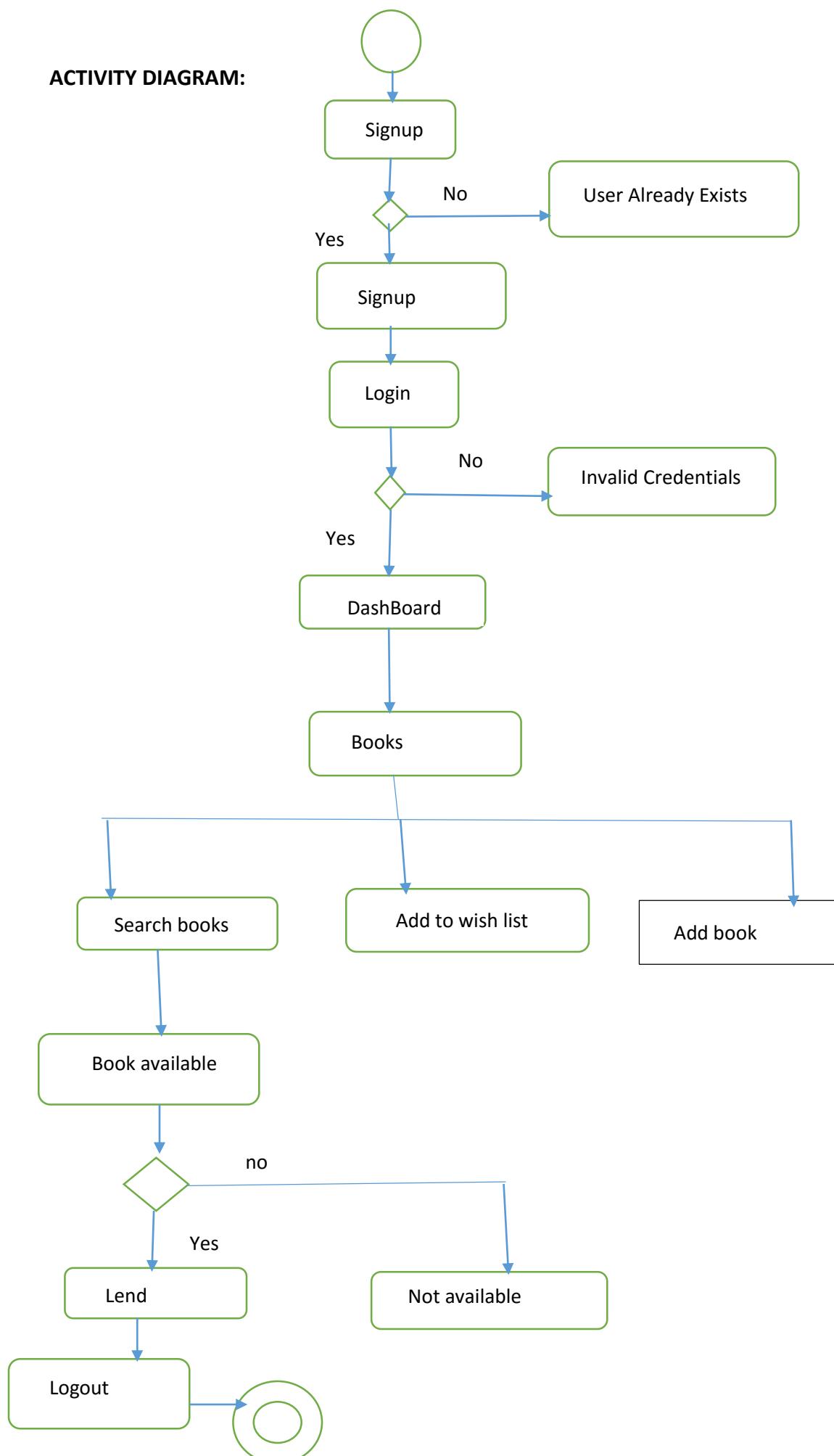


SEQUENCE DIAGRAM:



CLASS DIAGRAM:

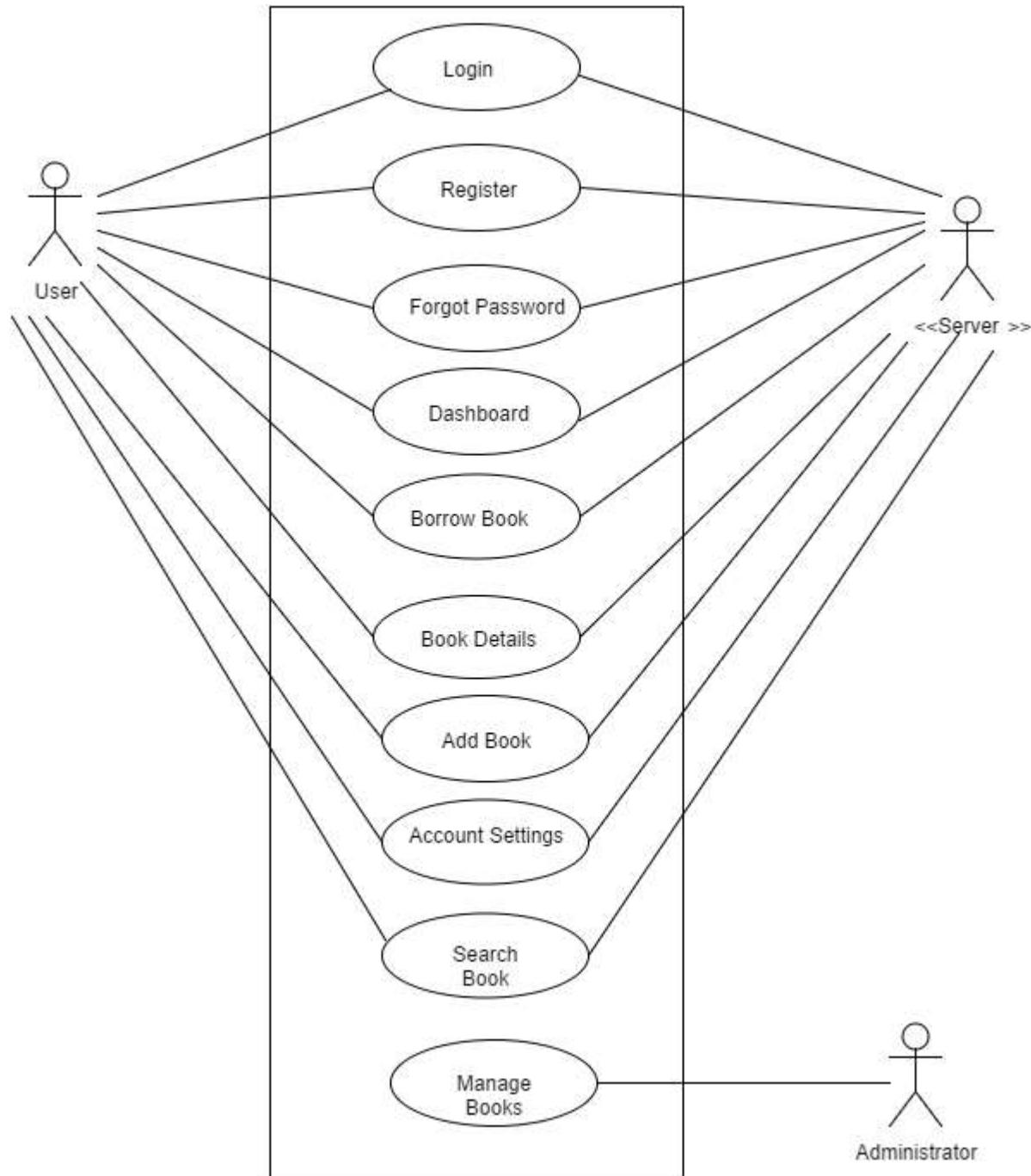


ACTIVITY DIAGRAM:

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         it('should call login on userDashboard', function() {
13             expect(userDashboard.login).toHaveBeenCalledWith('pr3md@mail.umkc.edu', 'password1');
14         });
15
16         describe('when the login is executed,', function() {
17             it('if successful, should change state to dashboard', function() {
18                 expect(stateMock.go).toHaveBeenCalledWith('user-dash.home');
19             });
20
21             it('if login is unsuccessful, should show a popup', function() {
22                 expect(ionicPopupMock.alert).toHaveBeenCalled();
23             });
24         });
25     });
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         describe('when the register is executed,', function() {
13             it('if successful, should change state to login', function() {
14                 expect(stateMock.go).toHaveBeenCalledWith('tab-login');
15             });
16
17             it('if registration is unsuccessful, should show a popup', function() {
18                 expect(ionicPopupMock.alert).toHaveBeenCalled();
19             });
20         });
21     });
22 });
23 });
24 });
25 });

```

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.

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.

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.
2. Web Portal for user administration has been deployed to DigitalOcean droplet.
3. Backend services are deployed to Heroku Nodejs platform.
4. MongoDB has been hosted on mlab.
5. All the source code has been pushed to Github.

SCREENSHOTS:

Login and Registration pages:



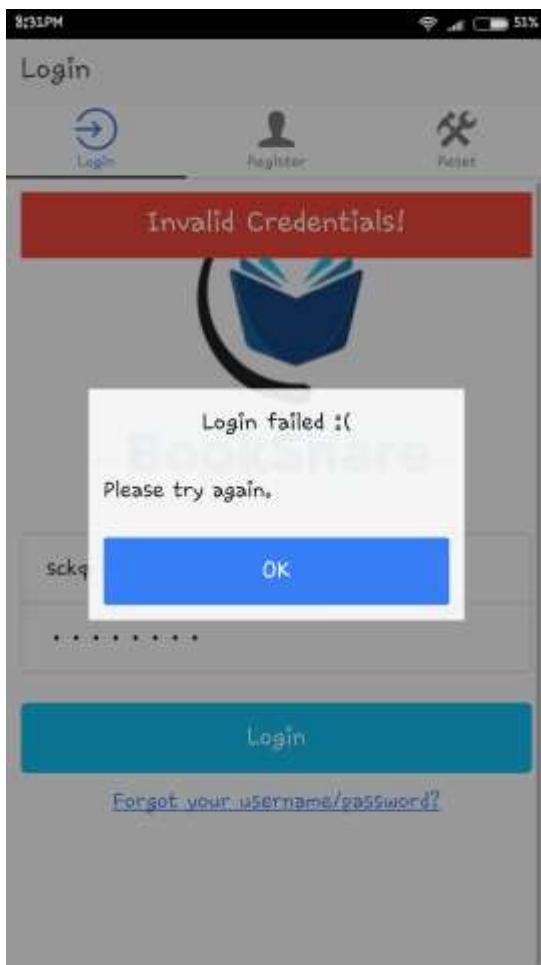
Registration of Existing Users:



Registration of New Users:



Invalid Credential and Successful Login:



Forgot Password:



User's Dashboard after Successful Login and Adding Books:



Hello Swetha Chandra

Book Title

Author Name

ISBN

Add my Book

Search..

Available Books

Headfirst Java

Kathy Sierra & Bert Bates

Cracking the Coding Intervi..

Gayle Laakmann McDowell



Hello Swetha Chandra

Software Engineering-A Practitioner's App

Roger S. Pressman, Bruce R. Maxim

9780072853186

Add my Book

Search..

Available Books

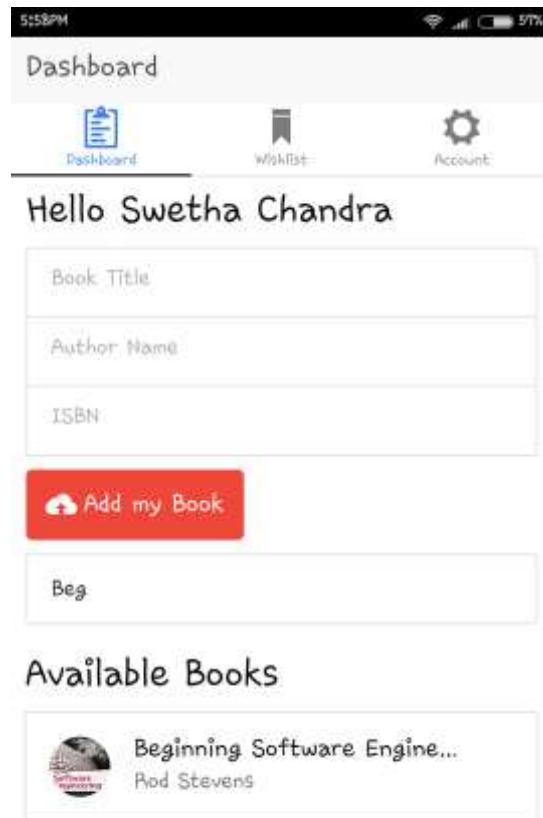
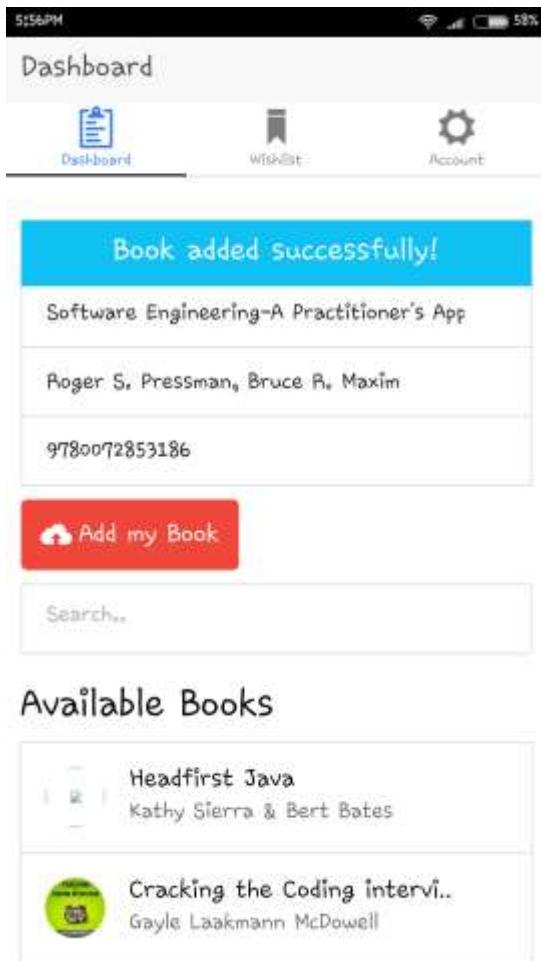
Headfirst Java

Kathy Sierra & Bert Bates

Cracking the Coding Intervi..

Gayle Laakmann McDowell

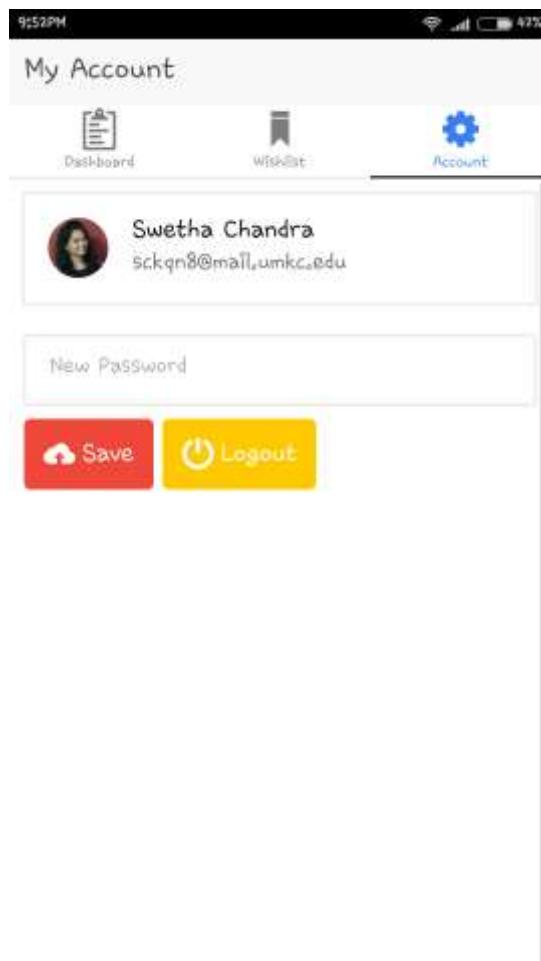
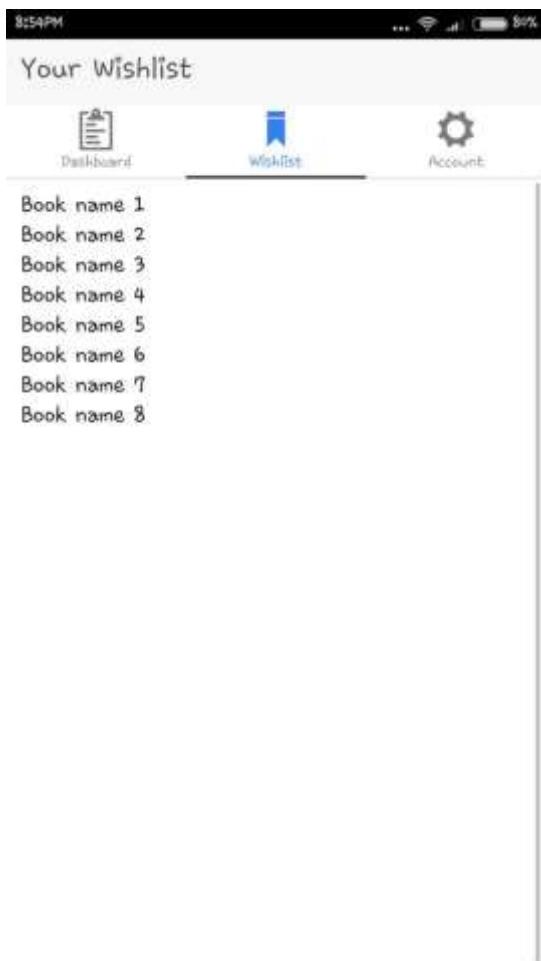
Book Added Successfully and Search for a Book:



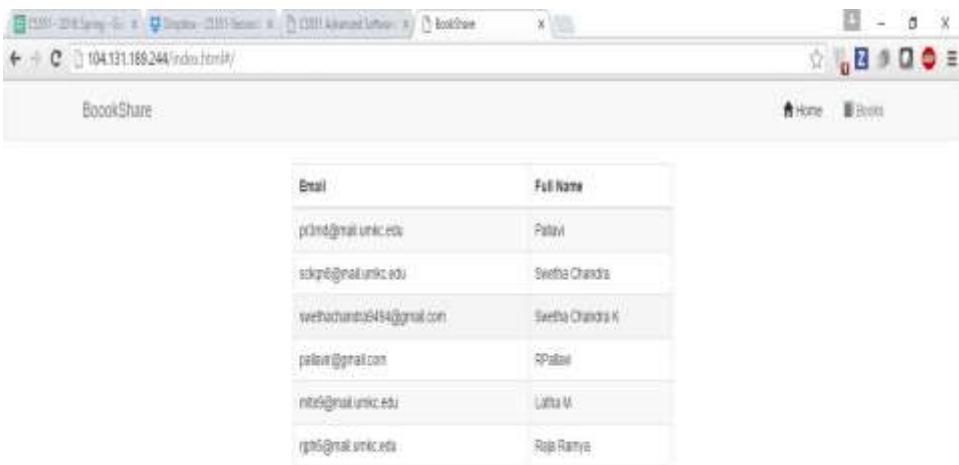
Selected Books can be added to Wishlist:



Wishlist and Logout Option:

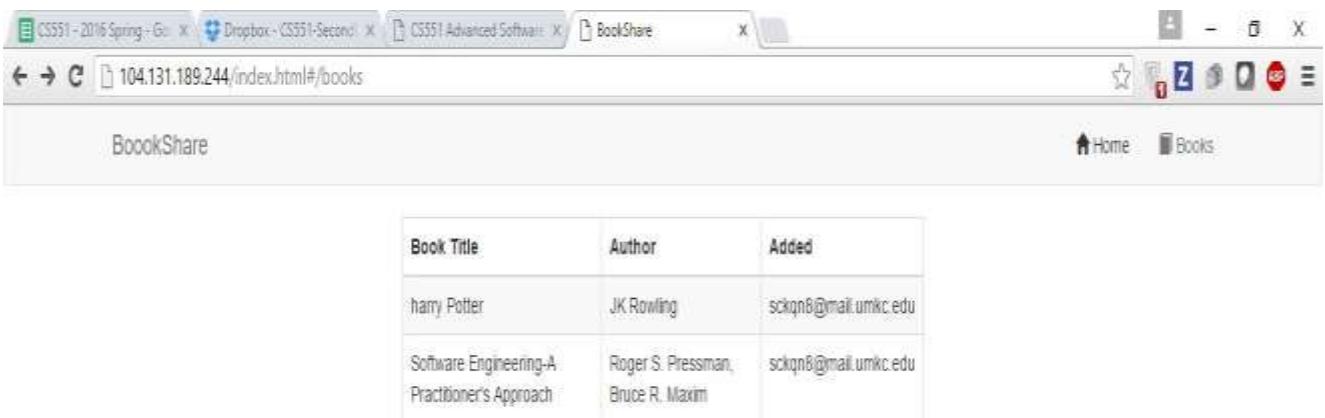


DEPLOYING WEB APPLICATION:



A screenshot of a web browser window titled "BookShare". The address bar shows the URL "104.131.189.244/index.html#". The page content displays a table with two columns: "Email" and "Full Name". The data rows are:

Email	Full Name
p03n6@mail.umkc.edu	Patry
sckqn8@mail.umkc.edu	Svetla Chandra
wethachanta6454@gmail.com	Svetla Chandra K
palav@rediff.com	RPalav
nita@mail.umkc.edu	Nita V
rpt6@mail.umkc.edu	Raja Ramya

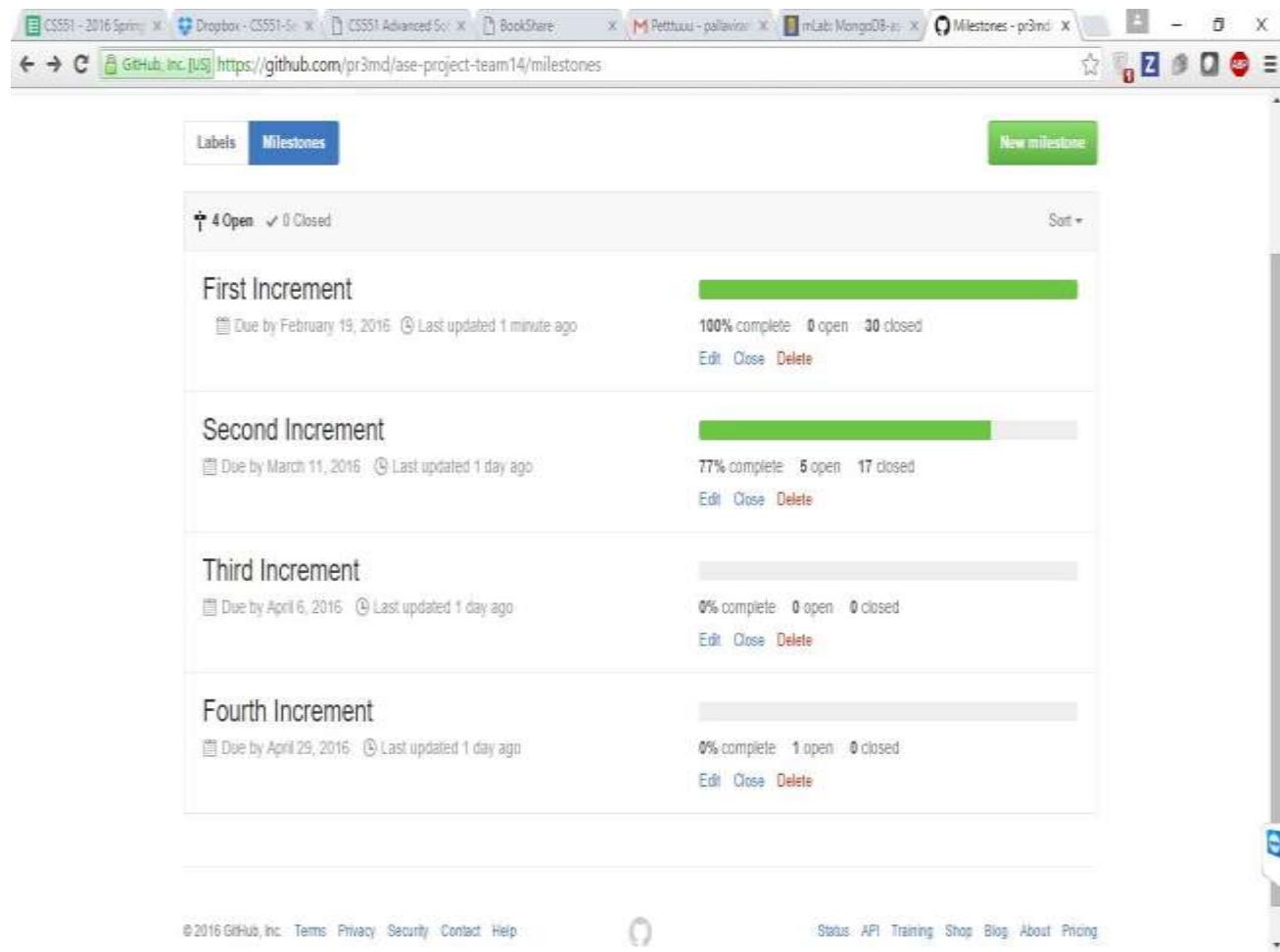


A screenshot of a web browser window titled "BookShare". The address bar shows the URL "104.131.189.244/index.html#books". The page content displays a table with three columns: "Book Title", "Author", and "Added". The data rows are:

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:



← → C GitHub, Inc [US] https://github.com/pr3md/ase-project-team14#boards?repos=50709901

The repository Search Pull requests Issues Gist + ToDo

pr3md / ase-project-team14

Unwatch ⚡ Star ⚡ Fork ⚡

Code Issues Full requests Boards Burndown Wiki Pulse Graphs Settings

Repos (11) Labels Milestones Assignees

New issue

Search

New Issues

#451 Create backend for creating Collections enhancement

#455 Learn How to manage Sessions in Mobile Applications enhancement

#459 Implement Facebook Login Authentication enhancement

Backlog

#456 Create REST Service for Forgot Username or Password enhancement

#460 Explore npm enhancement

#461 Learn AngularJS enhancement

To Do

#462 Learn How to manage Sessions in Mobile Applications enhancement

#463 Implement Facebook Login Authentication enhancement

In Progress

#464 Watch Egghead videos on AngularJS enhancement

Done

Closed

#465 Learn How to manage Sessions in Mobile Applications enhancement

#466 Refactor Ionic Tutorials enhancement

#467 Improve User Experience enhancement

#468 Learn MongoDB enhancement

Add a Pipeline

This screenshot shows a GitHub project board for the repository 'pr3md / ase-project-team14'. The board is organized into six columns: 'New Issues', 'Backlog', 'To Do', 'In Progress', 'Done', and 'Closed'. Each column contains a list of GitHub issues with their titles, descriptions, and labels. The 'New Issues' column has two items: '#451 Create backend for creating Collections' and '#455 Learn How to manage Sessions in Mobile Applications'. The 'Backlog' column has three items: '#456 Create REST Service for Forgot Username or Password', '#460 Explore npm', and '#461 Learn AngularJS'. The 'To Do' column has two items: '#462 Learn How to manage Sessions in Mobile Applications' and '#463 Implement Facebook Login Authentication'. The 'In Progress' column has one item: '#464 Watch Egghead videos on AngularJS'. The 'Done' and 'Closed' columns each have four items, mostly labeled as 'enhancement'.

← → C GitHub, Inc [US] https://github.com/pr3md/ase-project-team14/issues

The repository Search Pull requests Issues Gist + ToDo

pr3md / ase-project-team14

Unwatch ⚡ Star ⚡ Fork ⚡

Code Issues Full requests Boards Burndown Wiki Pulse Graphs Settings

Filters Issue is open Labels Milestones New issue

Open ✓ Closed

Author Labels Milestones Assignees Sort

Create backend for creating Collections enhancement

opened 4 days ago by pr3md Second Increment New Issues

Learn AngularJS enhancement

opened 21 days ago by pr3md Second Increment To Do

Explore npm

opened 21 days ago by pr3md Second Increment Backlog

Create REST Service for Forgot Username or Password enhancement

opened 21 days ago by MudduLatha Fourth Increment New Issues

Learn How to manage Sessions in Mobile Applications enhancement

opened 21 days ago by MudduLatha Second Increment Backlog

Implement Facebook Login Authentication enhancement

opened 21 days ago by MudduLatha Second Increment To Do

This screenshot shows the GitHub issues page for the repository 'pr3md / ase-project-team14'. The page lists several open issues, each with a title, description, author, and labels. The issues are: 'Create backend for creating Collections' (enhancement), 'Learn AngularJS' (enhancement), 'Explore npm' (enhancement), 'Create REST Service for Forgot Username or Password' (enhancement), 'Learn How to manage Sessions in Mobile Applications' (enhancement), and 'Implement Facebook Login Authentication' (enhancement). The page includes filters for 'Issue is open', 'Labels', and 'Milestones', and a 'New issue' button. At the top, there are links for 'Pull requests', 'Issues', 'Gist', and 'ToDo'.

WORK COMPLETED:

Developed RESTful services using Nodejs for handling registration, login and adding new books. 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 and Account Settings are developed.

Service Design: The base design of this application concentrates on User Interface and Application architecture. All the user screens are designed and backend functionality will be built in further increments.

Service Implementation: At this stage, fake data is being used in **services.js** file to fill the user interface, once backend services are established data will be retrieved from the Database.

WORK TO BE COMPLETED:

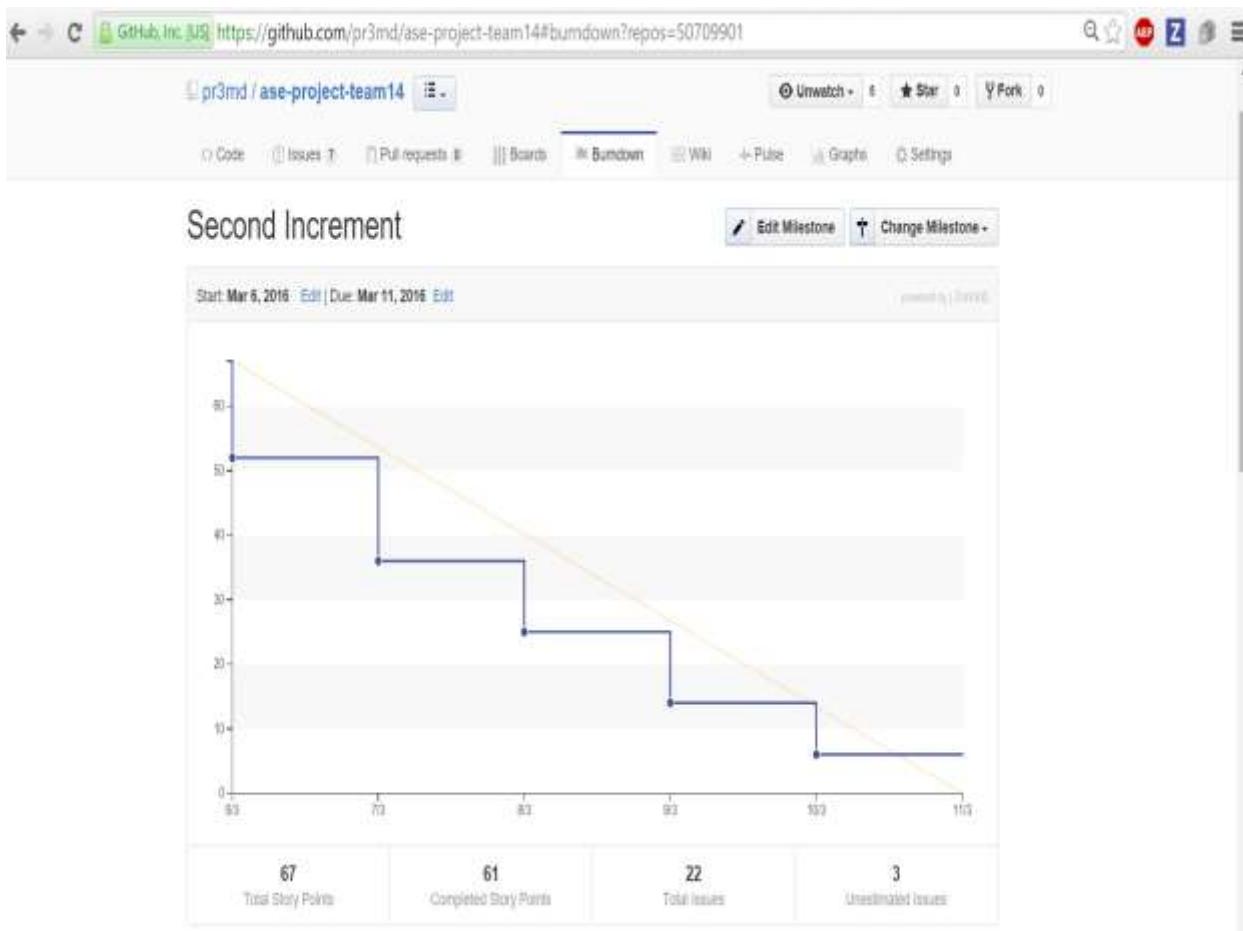
In the next increment, the following features will be added:

1. List of books added by the user.
2. Uploading image for Books.
3. User Profiles.
4. Wishlist implementation.
5. Deployment of Nodejs backend to our own servers.
6. Secure transmission of data over HTTPS.
7. Modification of Account Details.
8. Browsing books by Categories.
9. User session management.

More details about third increment are added here:

<https://github.com/pr3md/ase-project-team14/milestones/Third%20Increment#boards?repos=50709901>

BURNDOWN CHART:



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

BIBLIOGRAPHY:

Wireframes design: <http://draw.io.com/>

Android studio: <http://developer.android.com/sdk/index.html>

ZenHub – Agile project management tool for GitHub: <https://www.zenhub.io>



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.

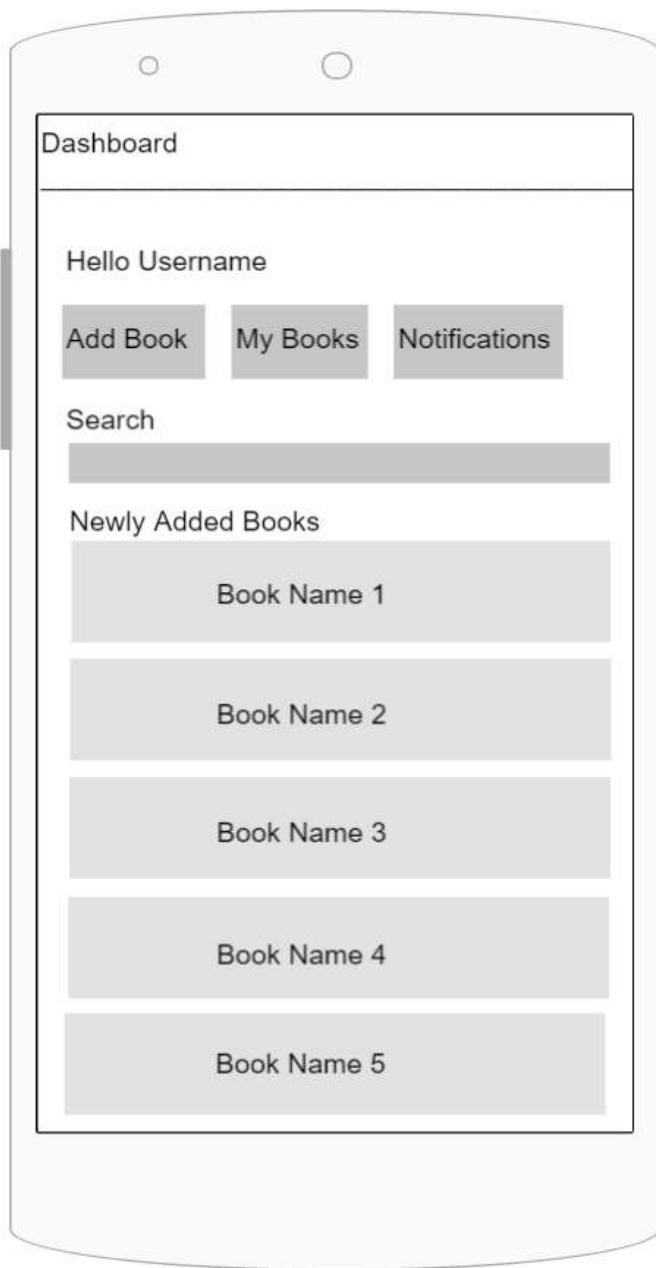
DETAIL DESIGN OF FEATURES (Using Tools):

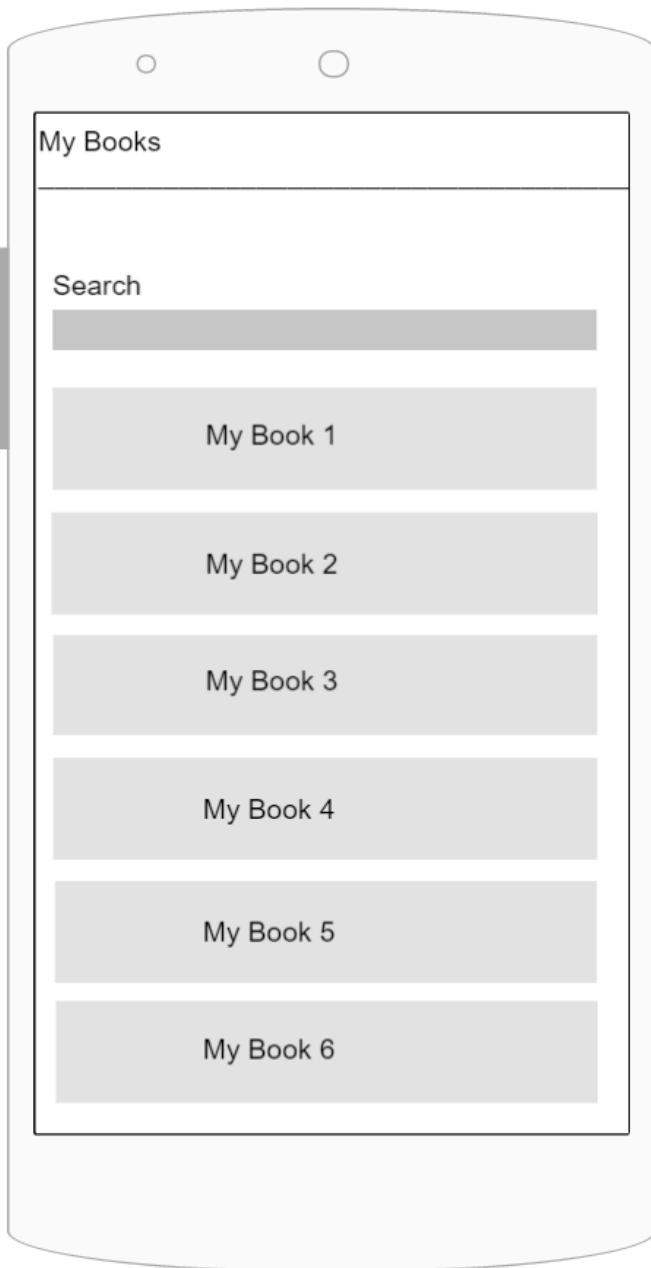
WIREFRAMES:

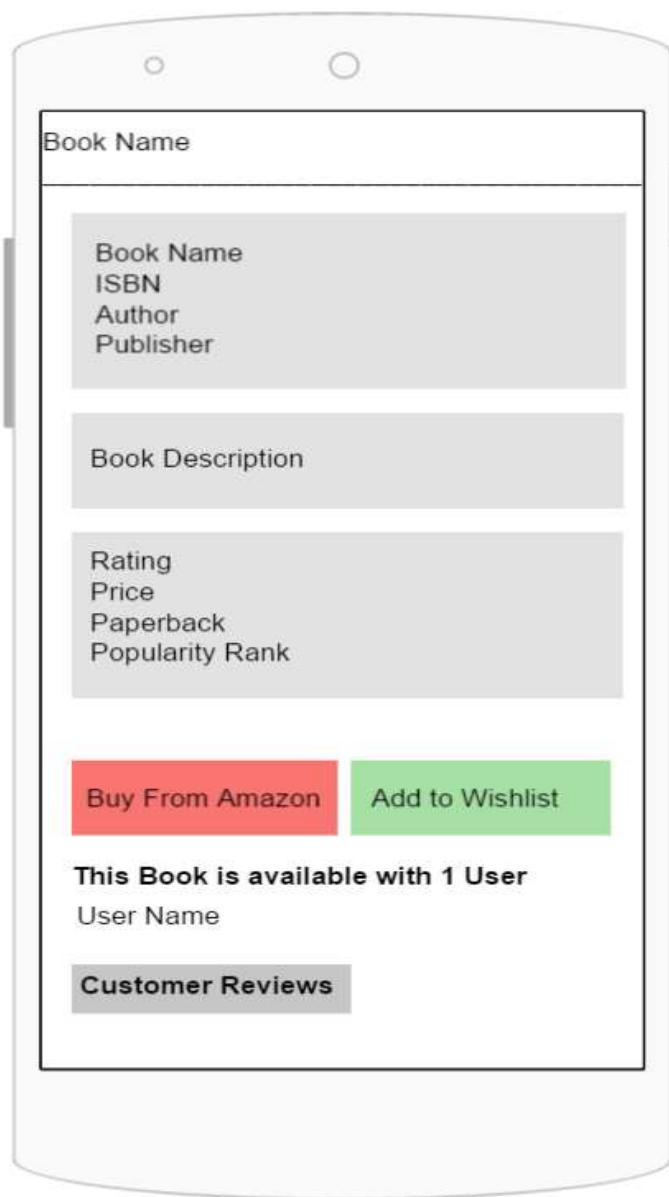


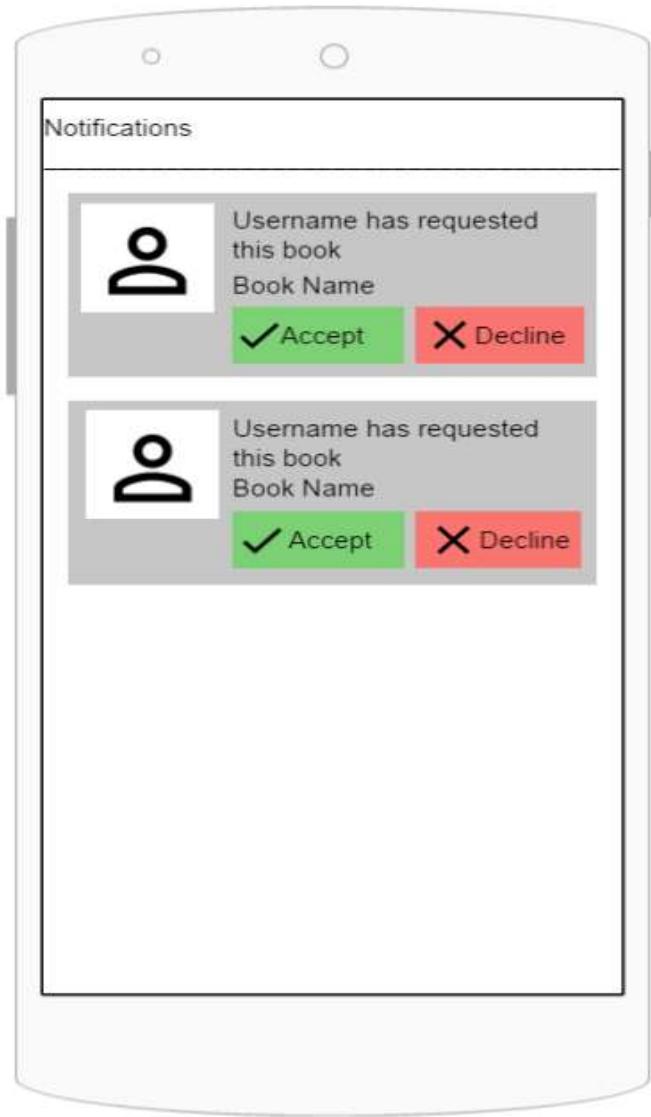






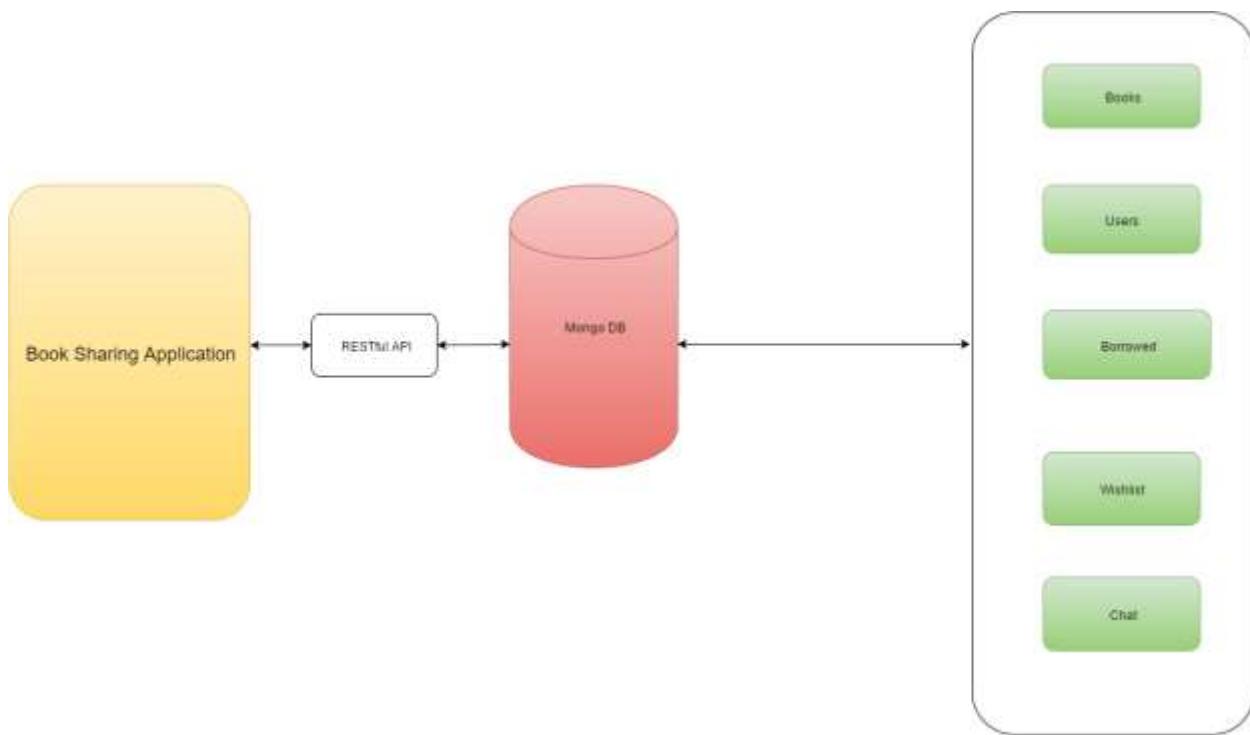




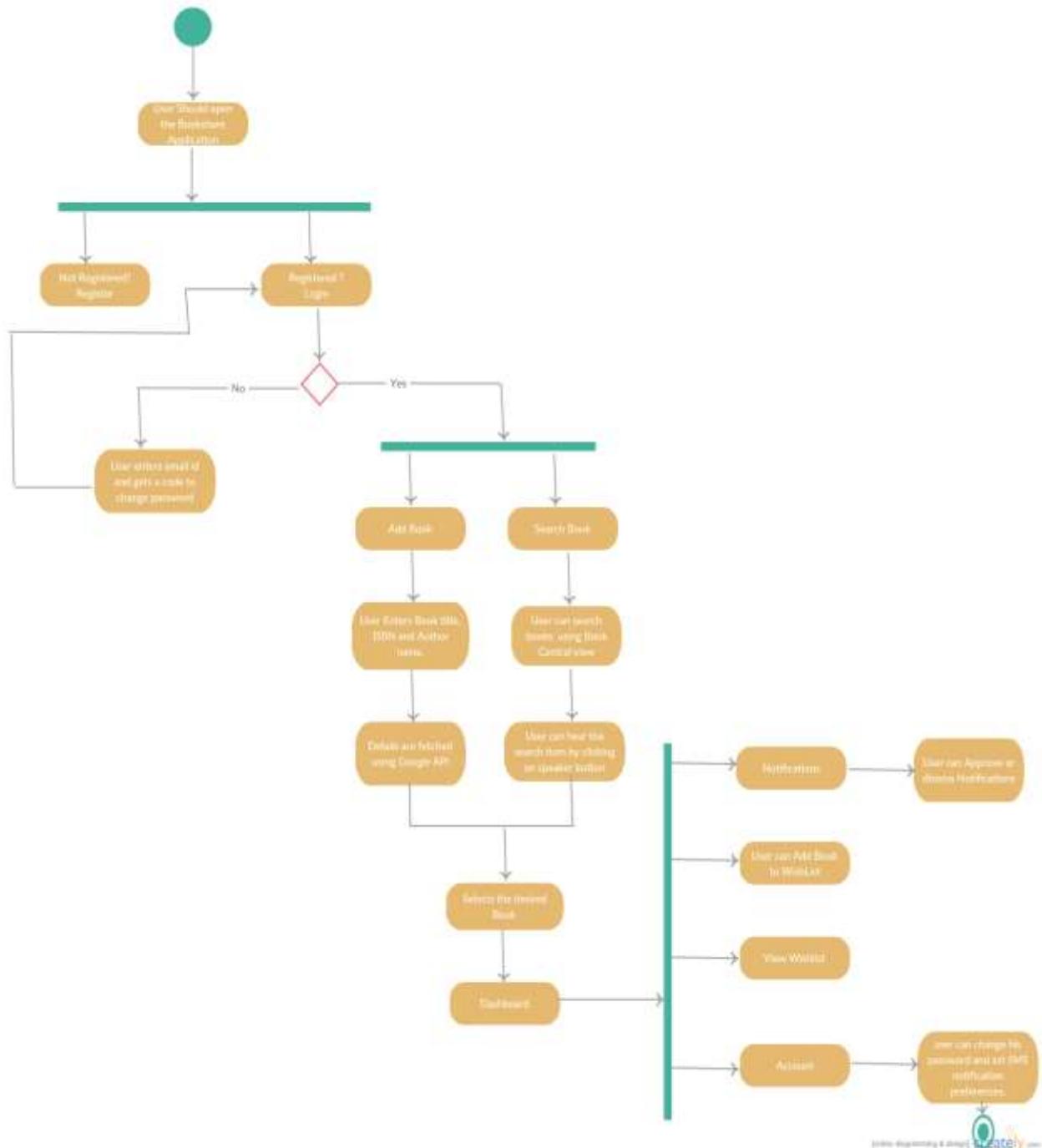




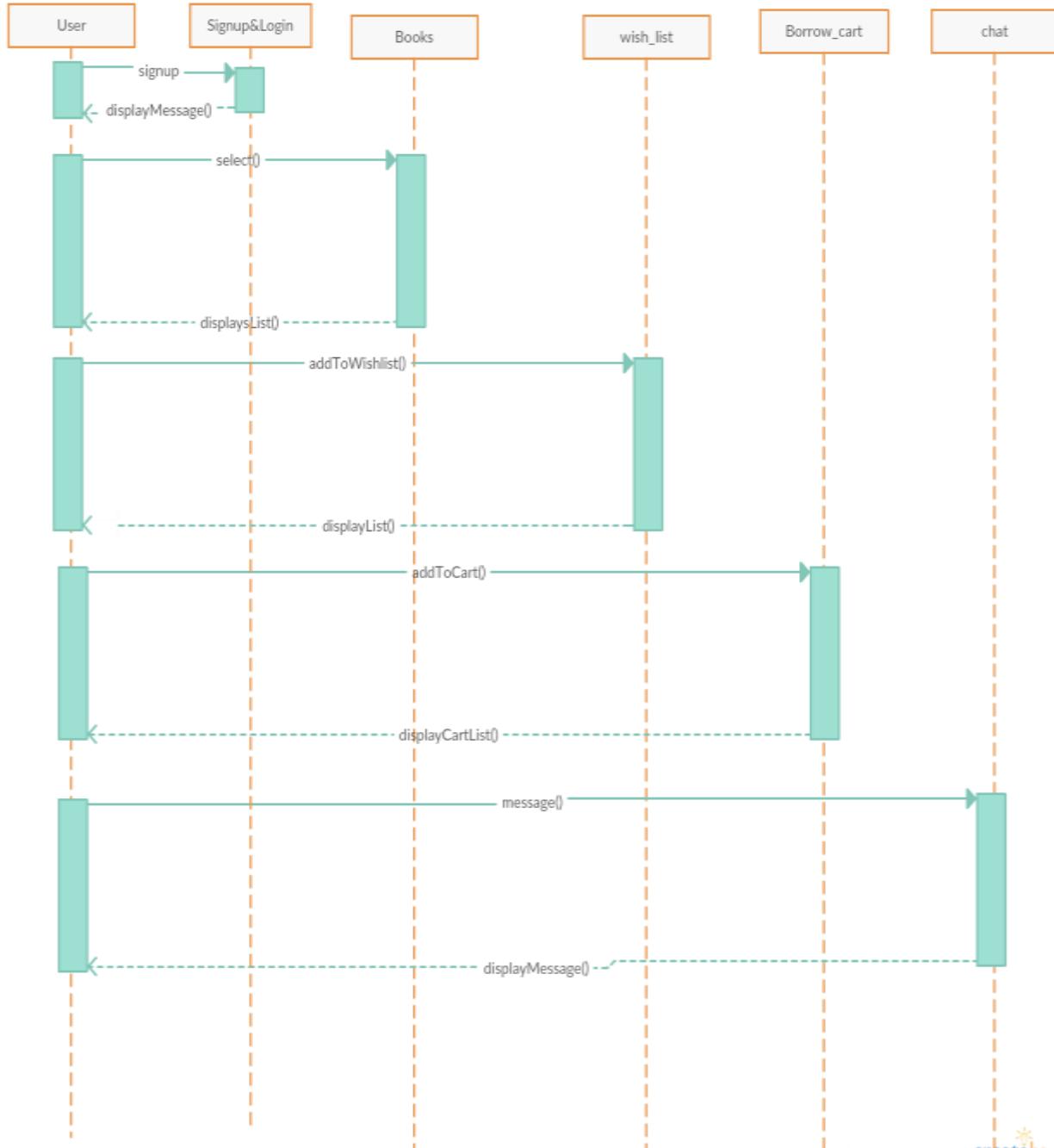
ARCHITECTURE DIAGRAM:



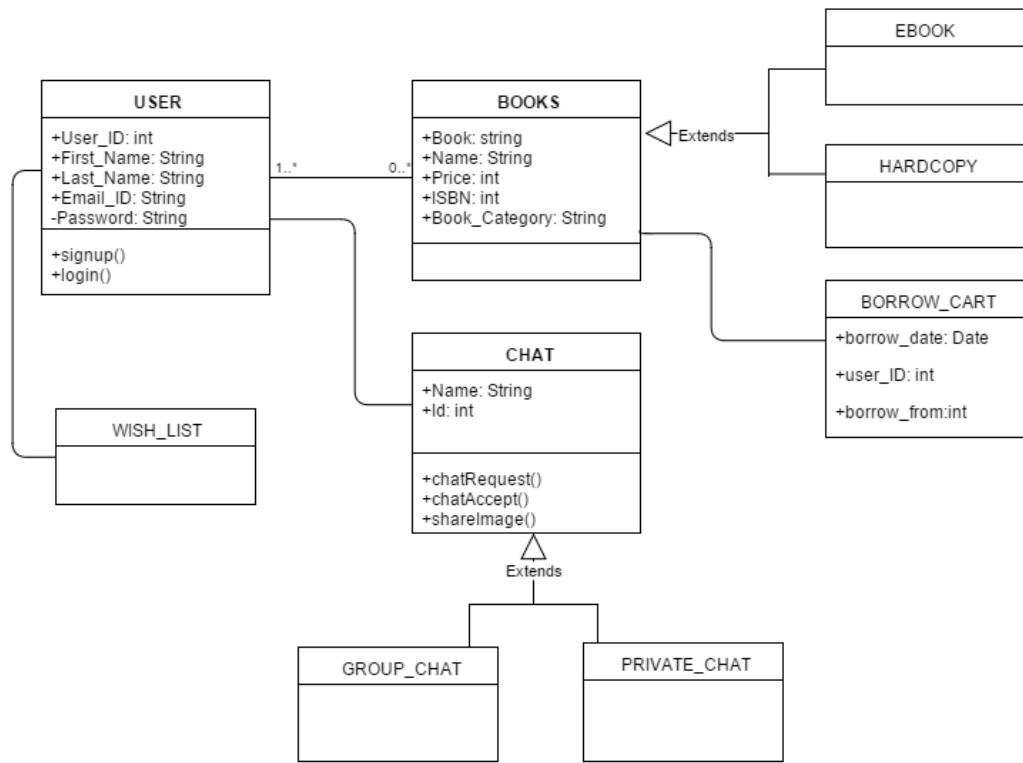
ACTIVITY DIAGRAM:



SEQUENCE DIAGRAM:



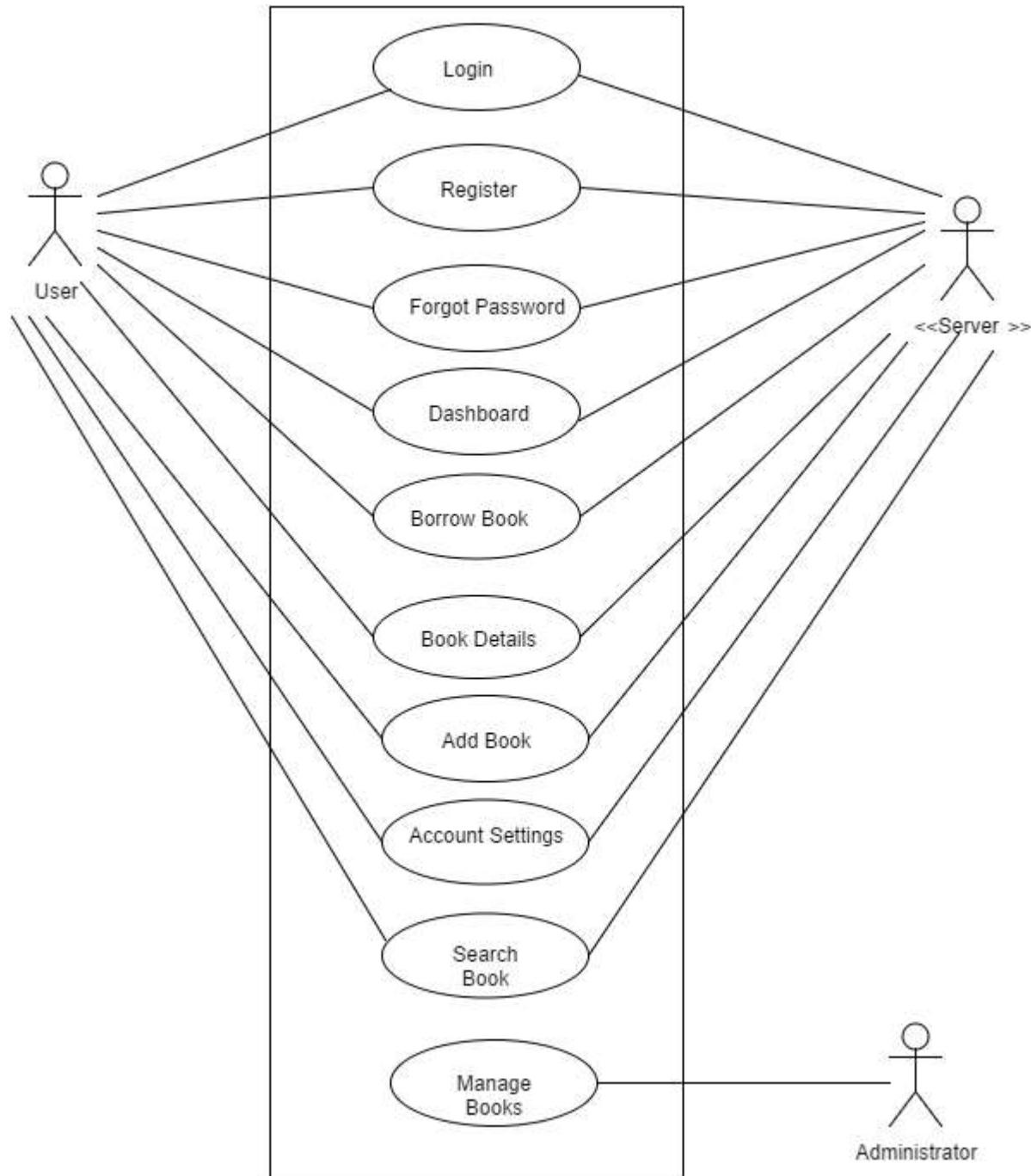
CLASS DIAGRAM:



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         it('should call login on userDashboard', function() {
13             expect(userDashboard.login).toHaveBeenCalledWith('pr3md@mail.umkc.edu', 'password1');
14         });
15
16         describe('when the login is executed,', function() {
17             it('if successful, should change state to dashboard', function() {
18                 expect(stateMock.go).toHaveBeenCalledWith('user-dash.home');
19             });
20
21             it('if login is unsuccessful, should show a popup', function() {
22                 expect(ionicPopupMock.alert).toHaveBeenCalled();
23             });
24         });
25     });
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 screenshot shows a browser window displaying the PageSpeed Insights dashboard at localhost:8100/#/dashboard/home. The dashboard has a header with a back button, forward button, refresh button, and a search bar containing the URL. Below the header is a title "Dashboard". The main content area features three colored cards: a red card for "Add New Book", a blue card for "My Books", and a green card for "Notifications 0". Below these cards, the page displays the analysis results for the specified URL. The analysis summary shows a Grade of C with an overall performance score of 76. It indicates that a ruleset named "YSlow(V2)" was applied. The page contains 23 components, with 6 being scripts. A sidebar on the left lists specific optimization recommendations, each with a letter grade (B, F, A, F, F, A, D) and a brief description.

localhost:8100/#/dashboard/home

Dashboard

Hello Swetha Chandra

Add New Book

My Books

Notifications 0

Grade C 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 \(6\)](#)

[Tweet](#) [Share](#)

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](#)

The screenshot shows a web browser window with the URL localhost:3100/#/dashboard/book-details/57054fa6f925ef11004bae2b. The page title is "How to Fix Everything For Dummies". On the left, there's a thumbnail image of the book cover. Below it, the book's details are listed: "How to Fix Everything For Dummies", ISBN: 076452091, By Gary Hedstrom/Peg Hedstrom/Judy Ondrea Tremore/Judy Tremore - House & Home, Publisher: John Wiley & Sons.

A large text box contains a summary: "A guide to troubleshooting and fixing a wide variety of appliances, gadgets, and machinery shows readers how to repair lamps, vacuum cleaners, dishwashers, dryers, garbage disposals, blenders, washers, televisions, radios, and other household items." A small icon of a person is next to the word "disposals".

The screenshot shows the Lighthouse performance audit tool interface. The overall grade is "Grade G" with an overall performance score of 75. The audit is run against the URL <http://localhost:3100/#/dashboard/book-details/57054fa6f925ef11004bae2b>. The audit results for "Make fewer HTTP requests" are displayed, showing 8 external JavaScript scripts. The audit items listed are:

- 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

The main content area shows a summary: "Grade B on Make fewer HTTP requests". It states: "This page has 8 external Javascript scripts. Try combining them into one." Below this, there's a note: "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." A "Read More" link is present.

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.

SERVER IMPLEMENTATION:

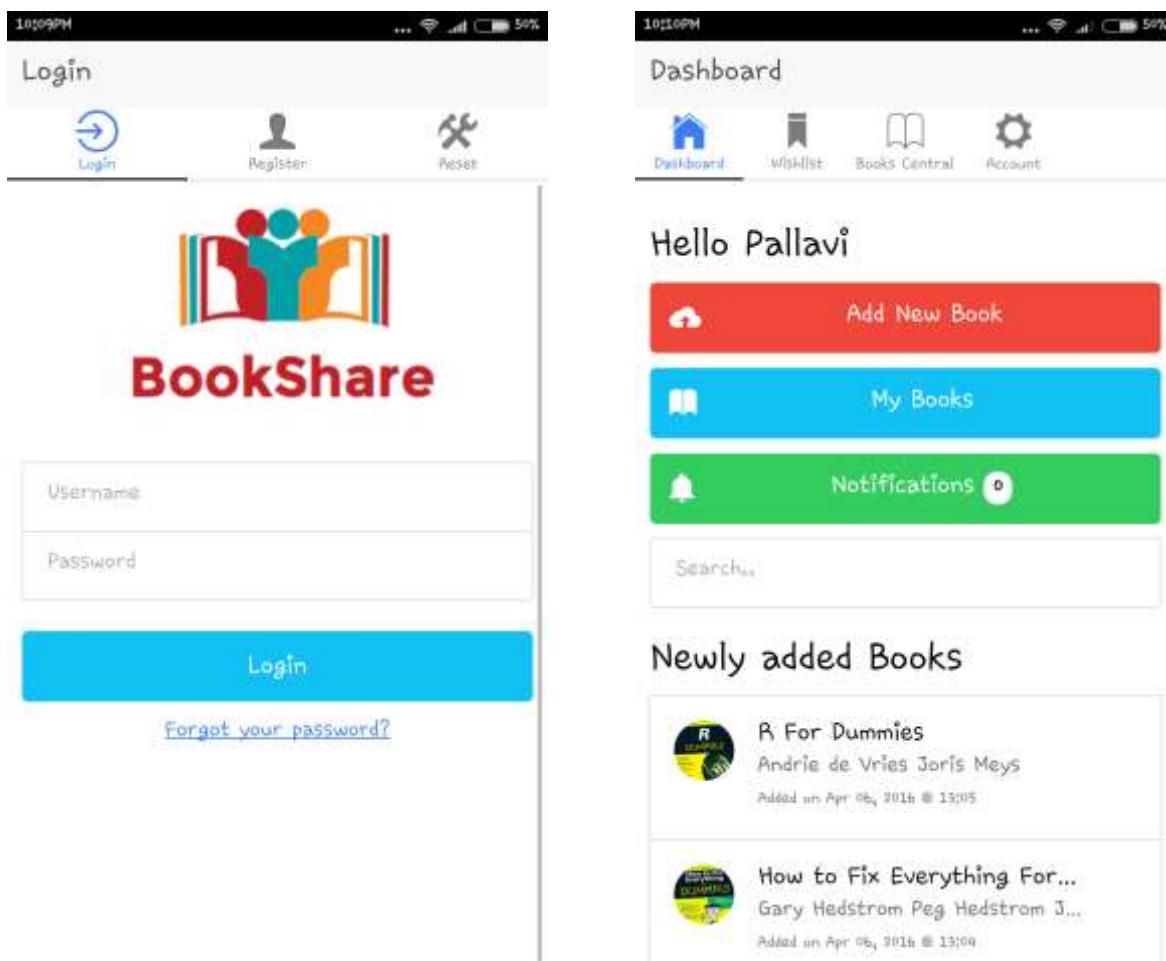
- Database:** MongoDB is being used for storing information about users, books and wish list. The database is currently hosted on mlab servers. As of now there are 2 collections defined. They are users and books. In Users collection, the document will have the fields: email, password, full name with 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).
- 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.

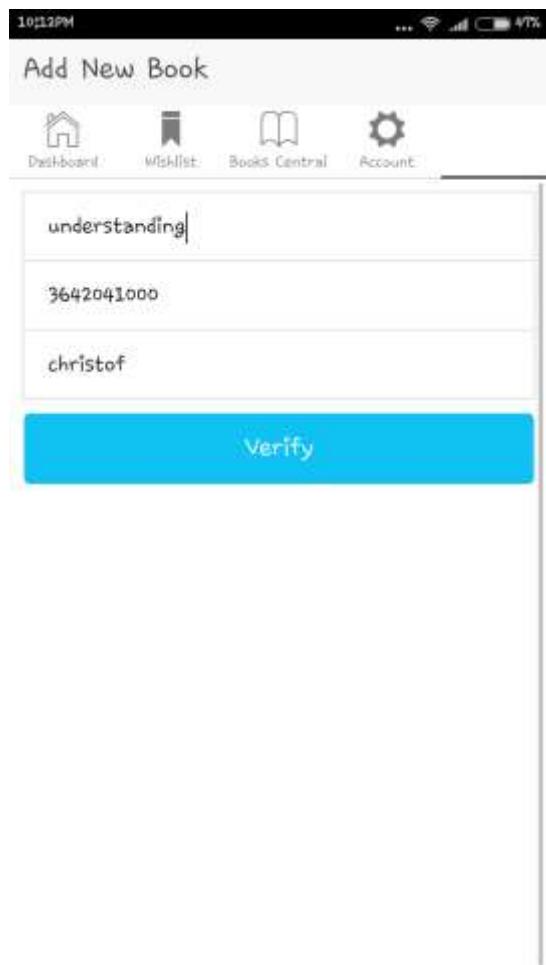
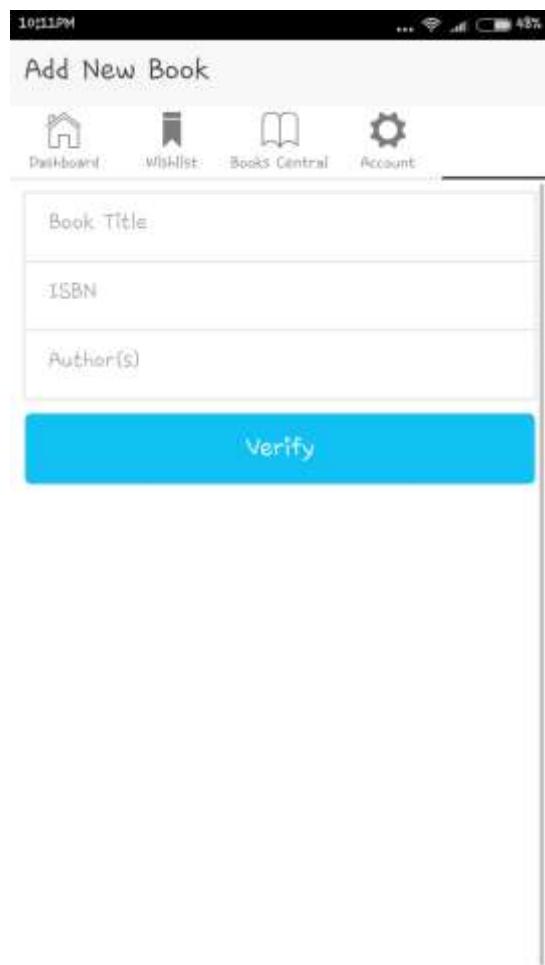
Service	Method	Service End Point	Request Payload	Response
Login	GET	https://floating-plateau-55000.herokuapp.com/bookshare/api/auth/login	Username, Password	Success – Fullname, Email
Register	POST	https://floating-plateau-55000.herokuapp.com/bookshare/api/auth/register	Email, Password, Full Name	Success – 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.
2. Web Portal for user administration has been deployed to DigitalOcean droplet.
3. Backend services are deployed to Heroku Nodejs platform.
4. MongoDB has been hosted on mlab.
5. All the source code has been pushed to Github.

SCREENSHOTS:





10:12PM

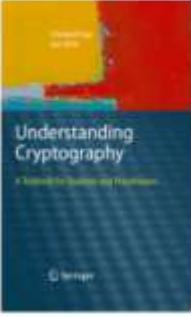
Add New Book

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

Add New Book

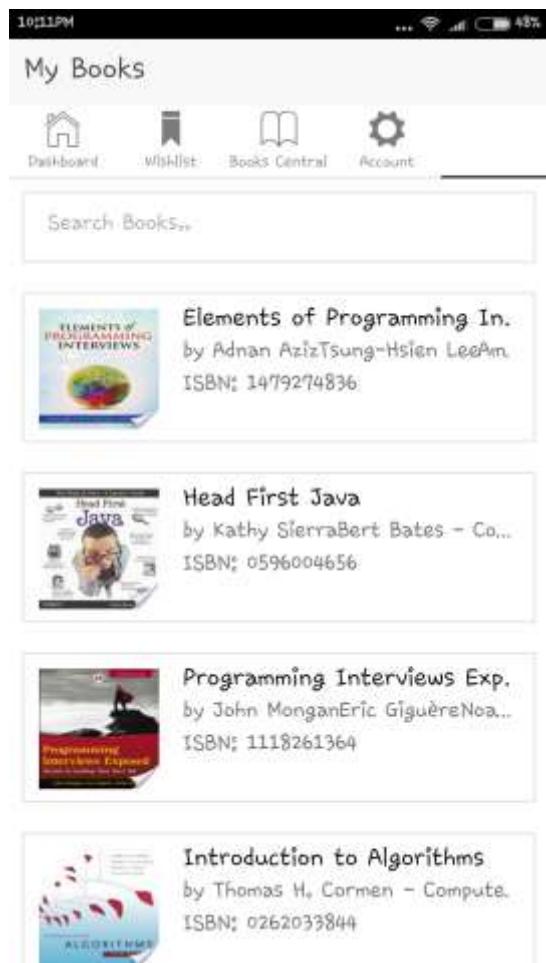
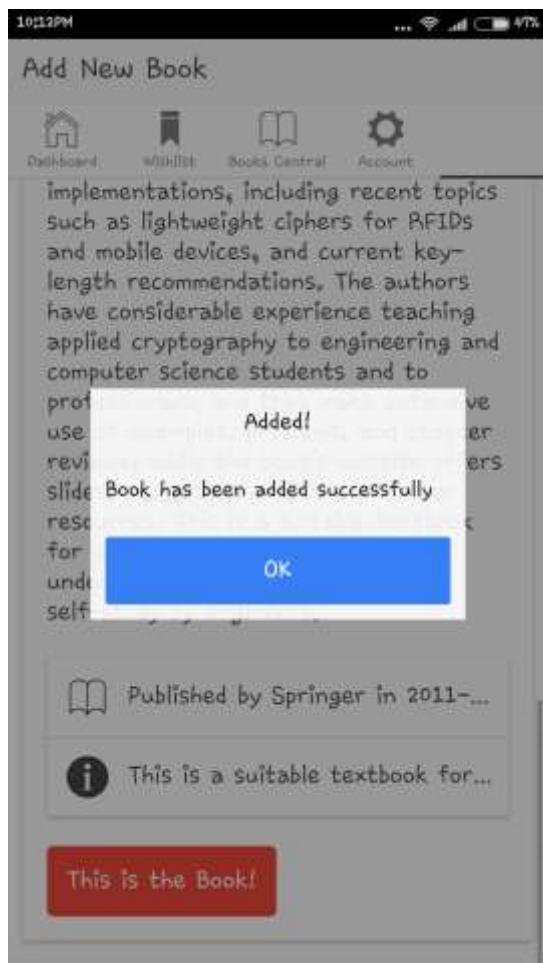
Dashboard WishList Books Central Account

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!



10:13PM



47%

Understanding Cryptography



Dashboard



Wishlist



Books Central



Account



Understanding Cryptography

ISBN: 97842041000

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

10:13PM



47%

Understanding Cryptography



Dashboard



Wishlist



Books Central



Account

Rating: 4 / 5

Price: \$49.95

Paperback: 372 pages

Popularity Rank: 28110



Buy from Amazon



Add to Wishlist



Reviews

This book is Available with
1 users.



Pallavi

10:14PM

How to Fix Everything For Dummies

Dashboard Wishlist Books Central Account


How to Fix Everything For ...
ISBN: 0764572091
By Gary HedstromPeg Hedstrom...
Publisher: John Wiley & Sons

A guide to troubleshooting and fixing a wide variety of appliances, gadgets, and machinery shows readers how to repair lamps, vacuum cleaners, dishwashers, dryers, garbage disposals, blenders, washers, televisions, radios, and other household items. (1))

Rating: 4 / 5
Price: \$19.99 
Paperback: 384 pages
Popularity Rank: 68829

Buy Now

10:14PM

How to Fix Everything For Dummies

Dashboard Wishlist Books Central Account

Price: \$19.99 

Paperback: 384 pages
Popularity Rank: 68829

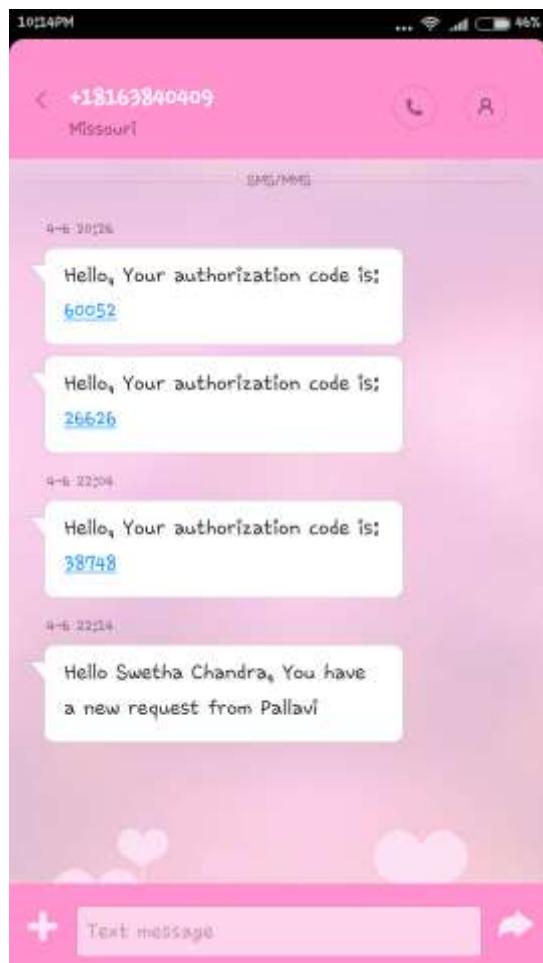
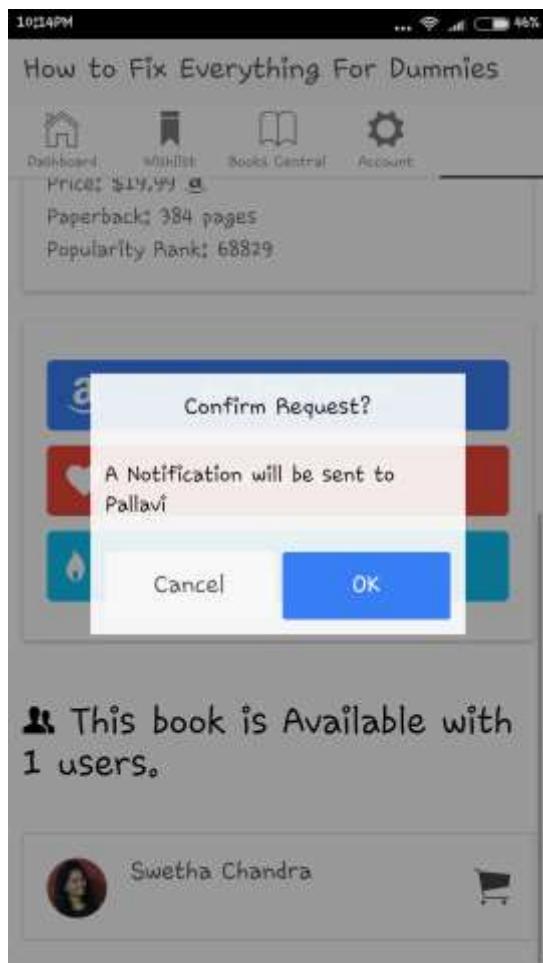
 Buy from Amazon
 Add to Wishlist
 Reviews

 This book is Available with
1 users.



Swetha Chandra







Hello Pallavi

Add New Book

My Books

Notifications 1

Search...

Newly added Books



R For Dummies

Andrie de Vries Joris Meys

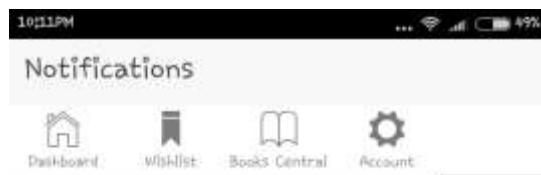
Added on Apr 06, 2016 @ 13:05



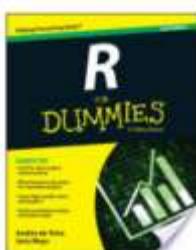
How to Fix Everything For...

Gary Hedstrom Peg Hedstrom J...

Added on Apr 06, 2016 @ 13:04



Swetha Chandra has requested ...
Apr 06, 2016 @ 22:10

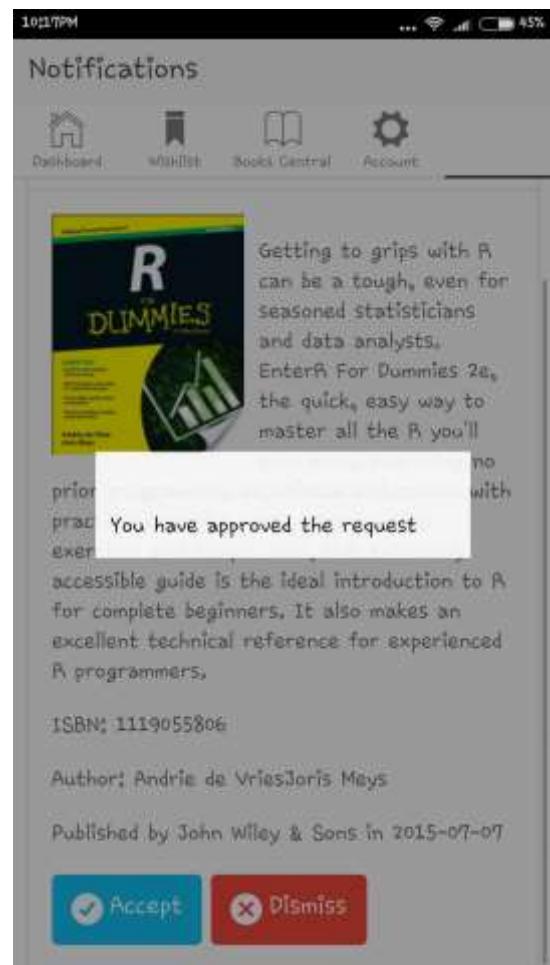
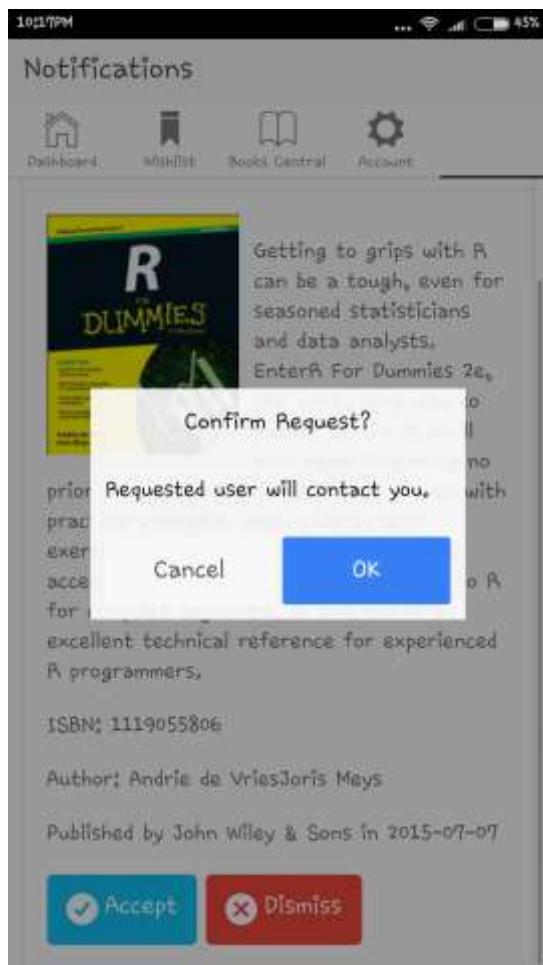


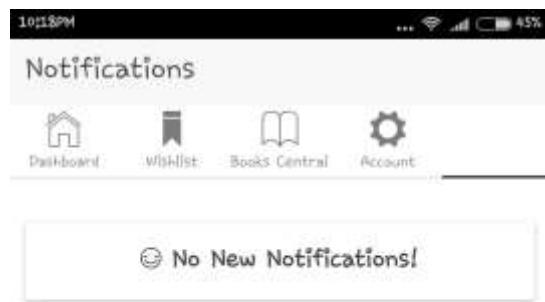
Getting to grips with R can be a tough, even for seasoned statisticians and data analysts. Enter R For Dummies 2e, the quick, easy way to master all the R you'll ever need. Requiring no prior programming experience and packed with practical examples, easy, step-by-step exercises and sample code, this extremely accessible guide is the ideal introduction to R for complete beginners. It also makes an excellent technical reference for experienced R programmers.

ISBN: 1119055806

Author: Andrie de VriesJoris Meys

Published by John Wiley & Sons in 2015-07-07





10:18PM

How to Fix Everything For Dummies

Dashboard WishList Books Central Account

Price: \$19.99

Paperback: 384 pages

Popularity Rank: 68829

Buy from Amazon

Book has been added to your Wishlist.

This book is Available with 1 users.

Swetha Chandra

10:15PM



46%

My Wishlist



Dashboard



Wishlist



Books Central



Account

Search Books...



[How to Fix Everything For ...](#)

by Gary HedstromPeg Hedstro...

ISBN# 0764572091

10:14PM



46%

My Account



Dashboard



Wishlist



Books Central



Account



Pallavi

pr3md@mail.umkc.edu

New Password

Change Password

Logout

DEPLOYING WEB APPLICATION:

A screenshot of a web browser window titled "BookShare". The address bar shows the URL "104.131.189.244/index.html#/". The page content displays a table with two columns: "Email" and "Full Name". The data rows are:

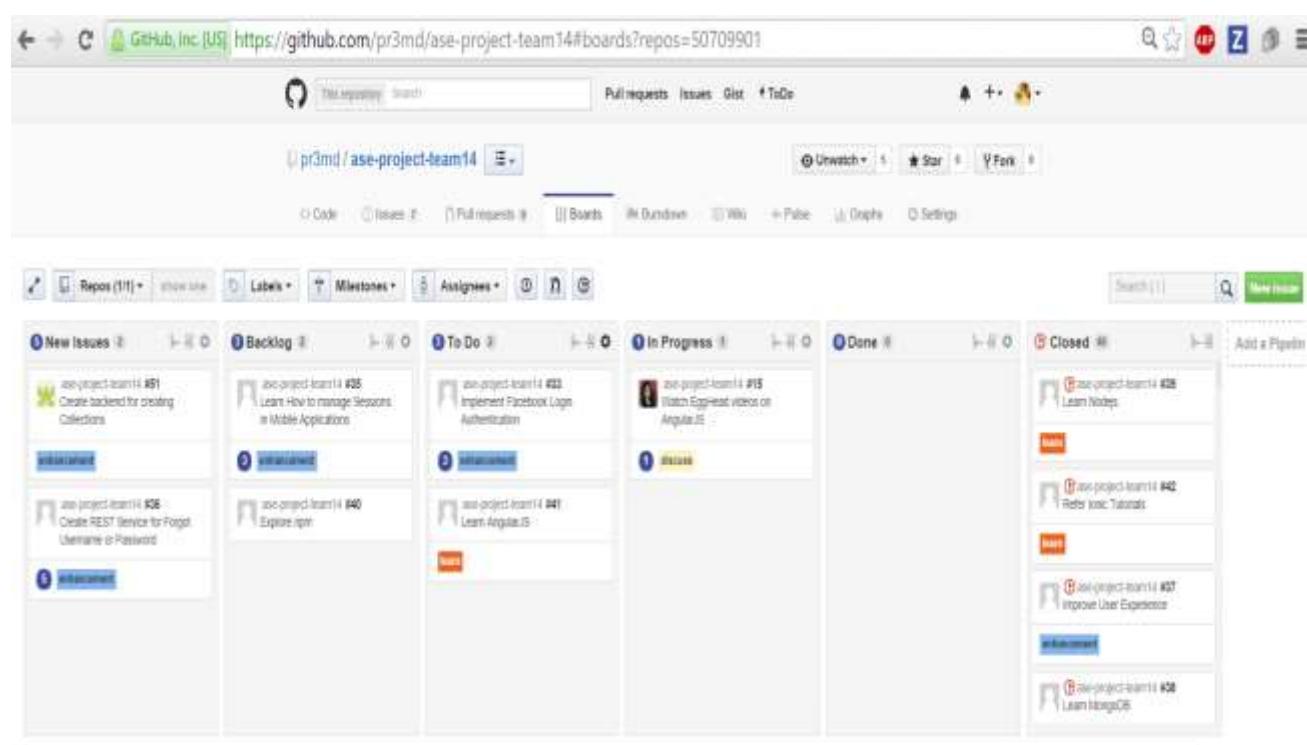
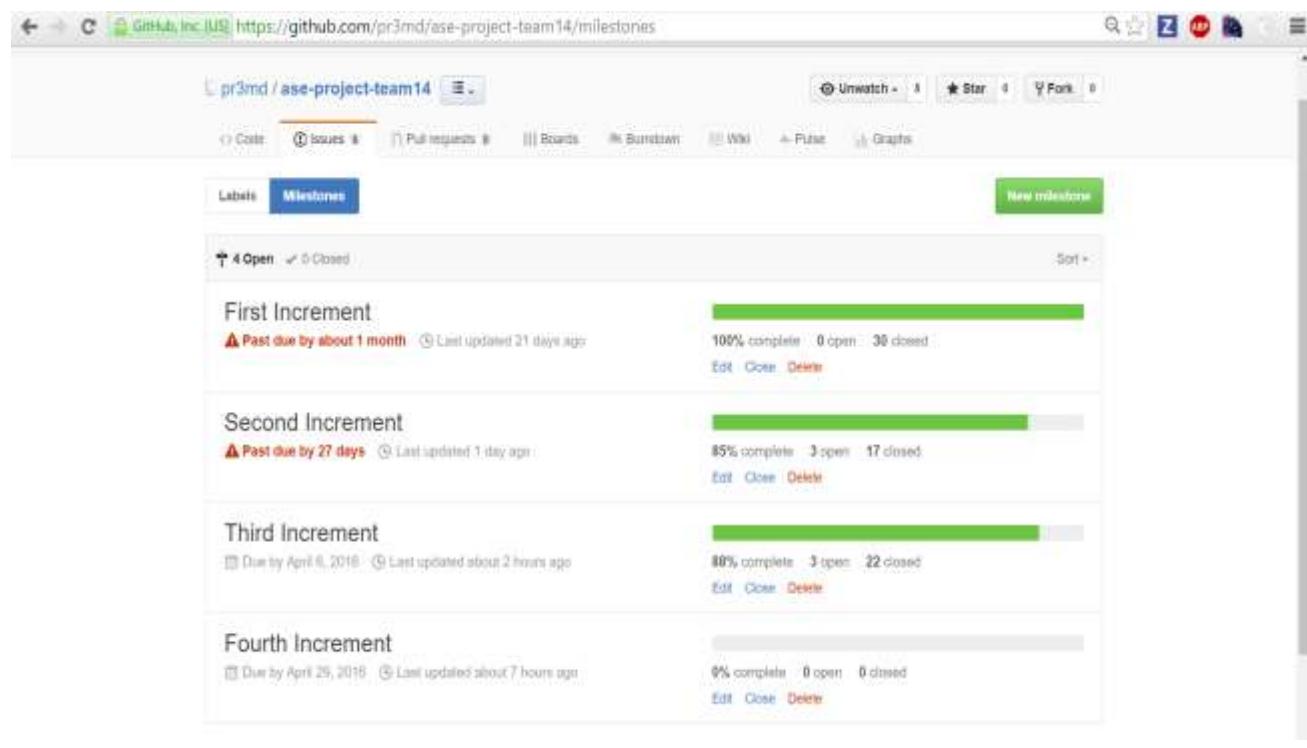
Email	Full Name
p03mg@mail.umkc.edu	Palavi
sckqn8@mail.umkc.edu	Swetha Chandra
swethachandra9494@gmail.com	Swetha Chandra K
palavi@gmail.com	RPalay
mrb9@mail.umkc.edu	Latha M
rph6@mail.umkc.edu	Raja Ramya

A screenshot of a web browser window titled "BookShare". The address bar shows the URL "104.131.189.244/index.html#/books". The page content displays a table with three columns: "Book Title", "Author", and "Added". The data rows are:

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 Issues page for the repository `pr3md/ase-project-team14`. The URL in the address bar is `https://github.com/pr3md/ase-project-team14/issues`. The page displays a list of 6 open issues, each with a title, description, and status. The issues are categorized by label: enhancement (5 issues) and bug (1 issue). The issues are listed in chronological order, from oldest at the top to newest at the bottom. The interface includes filters for 'Issue is open', 'Labels' (enhancement, bug), and 'Milestones'. A green 'New issue' button is located in the top right corner.

Issue #	Title	Type	Author	Milestone	Status
#6	Create backend for creating Collections	enhancement	ab1	Second Increment	Open
#41	Learn AngularJS	bug	ab1	Second Increment	Closed
#40	Explore npm	enhancement	ab1	Second Increment	Open
#39	Create REST Service for Forgot Username or Password	enhancement	ab1	Fourth Increment	Open
#38	Learn How to manage Sessions in Mobile Applications	enhancement	ab1	Second Increment	Open
#37	Implement Facebook Login Authentication	enhancement	MudduLathe	Second Increment	Open

WORK COMPLETED:

Developed RESTful services using Nodejs for handling registration, login and adding new books. 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 and Account Settings are developed.

Service Design: The base design of this application concentrates on User Interface and Application architecture. All the user screens are designed and backend functionality will be built in further increments.

Service Implementation: At this stage, fake data is being used in `services.js` file to fill the user interface, once backend services are established data will be retrieved from the Database.

WORK TO BE COMPLETED:

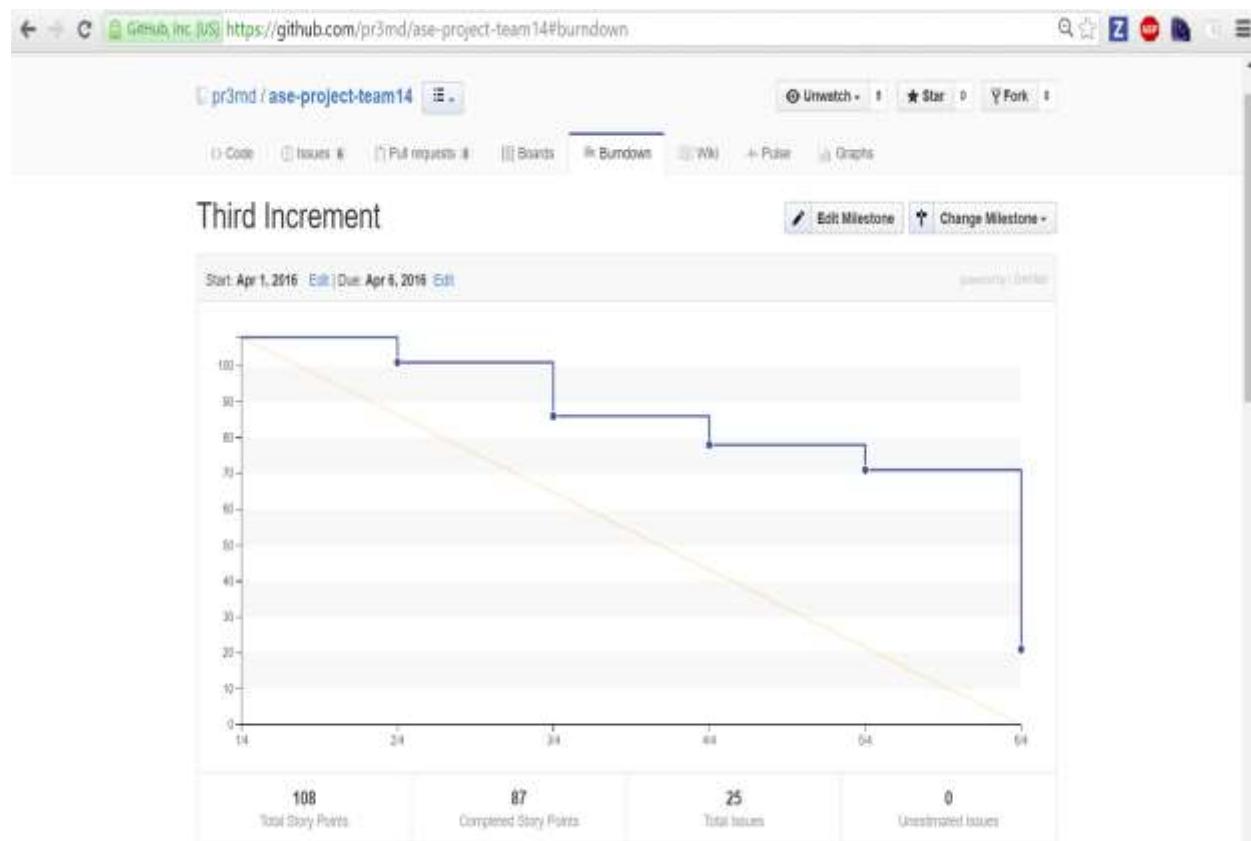
In the next increment, the following features will be added:

1. Secure transmission of data over HTTPS.
2. Browsing books by Categories.
3. User session management.
4. Bug fixing.
5. Splash Screen.

More details about fourth increment are added here:

<https://github.com/pr3md/ase-project-team14/milestones/Fourth%20Increment#boards?repos=50709901>

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



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.

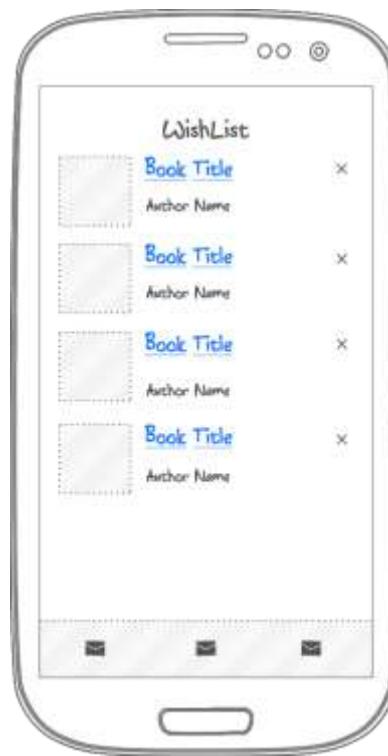
EXISTING SERVICES/API:

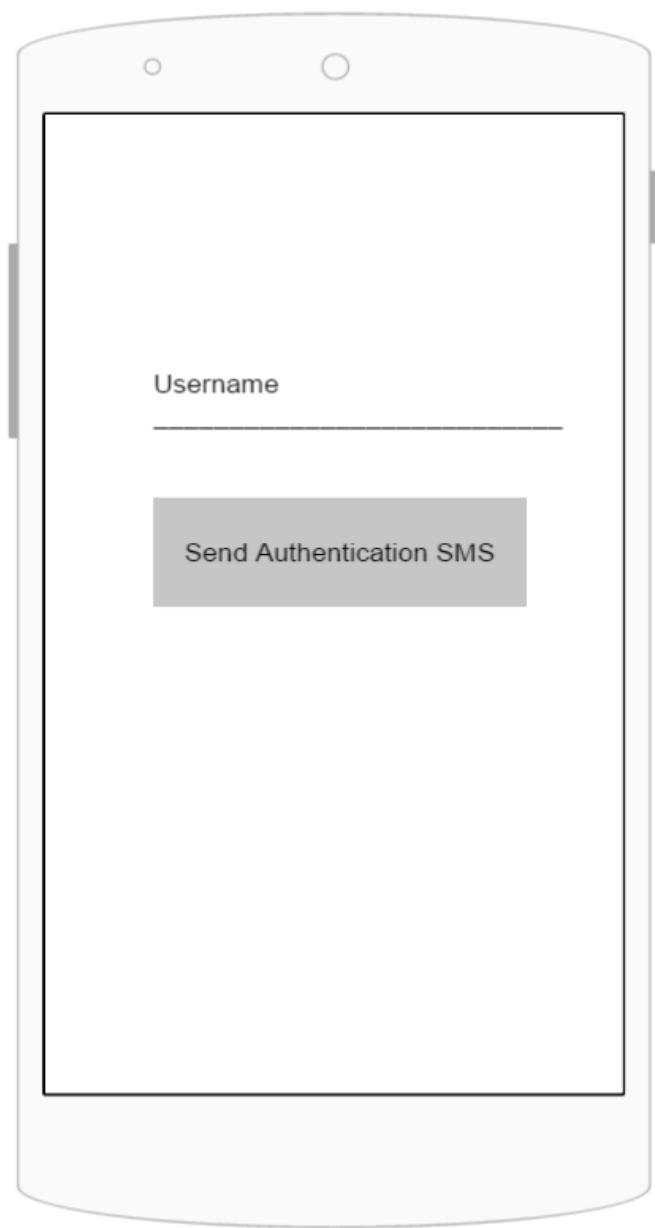
1. Application was developed using Ionic Hybrid Mobile apps Framework on Jet Brains WebStorm IDE.
2. MongoDB is used for storing the data, MongoDB is being hosted on mlab.
3. Backend Services are built using Nodejs, Nodejs is hosted on Heroku platform.
4. Gravatar API is being used to pull the images 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.

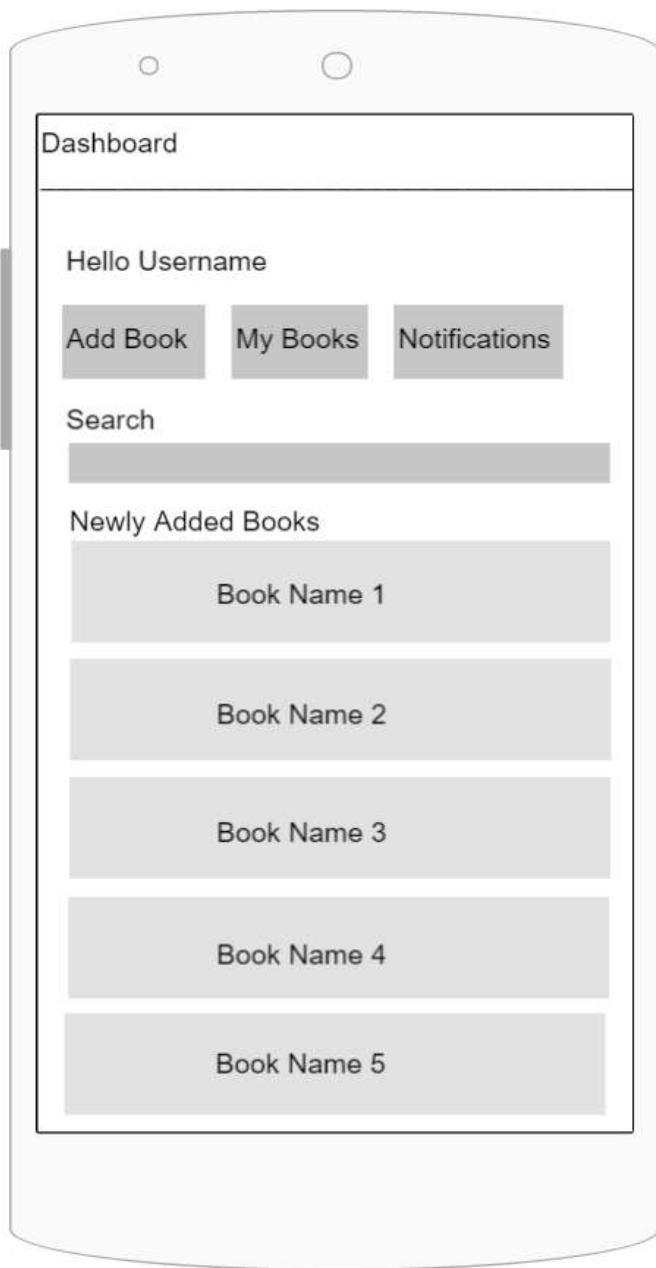
DETAIL DESIGN OF FEATURES (Using Tools):

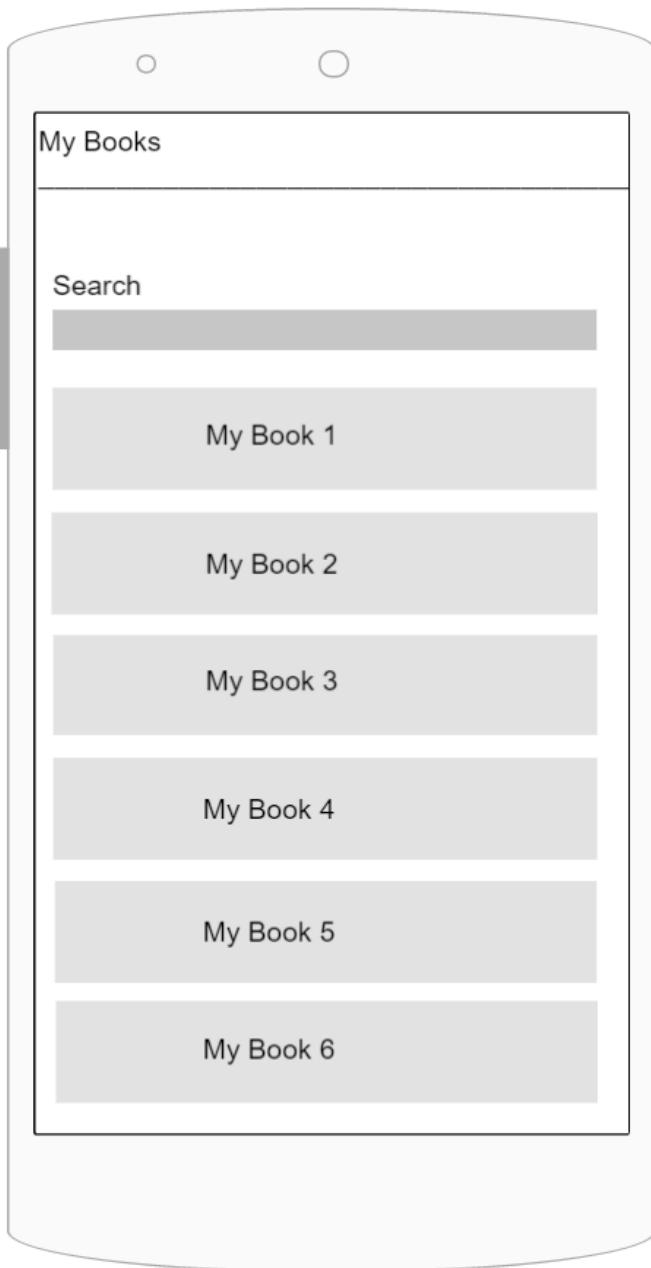
WIREFRAMES:

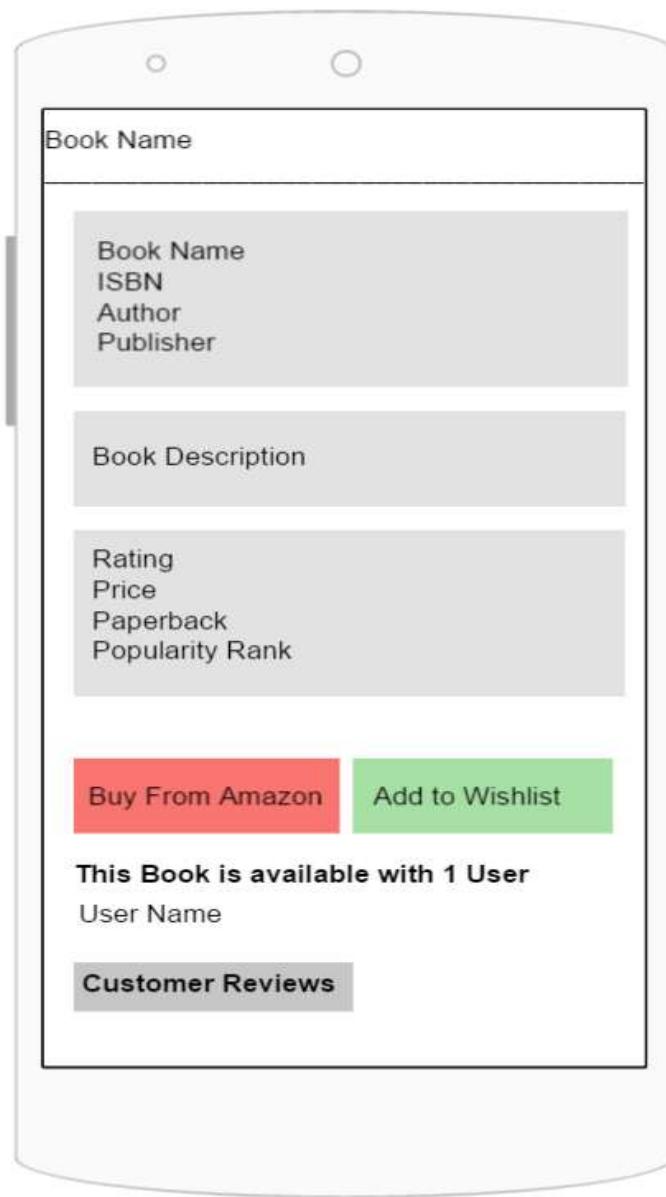


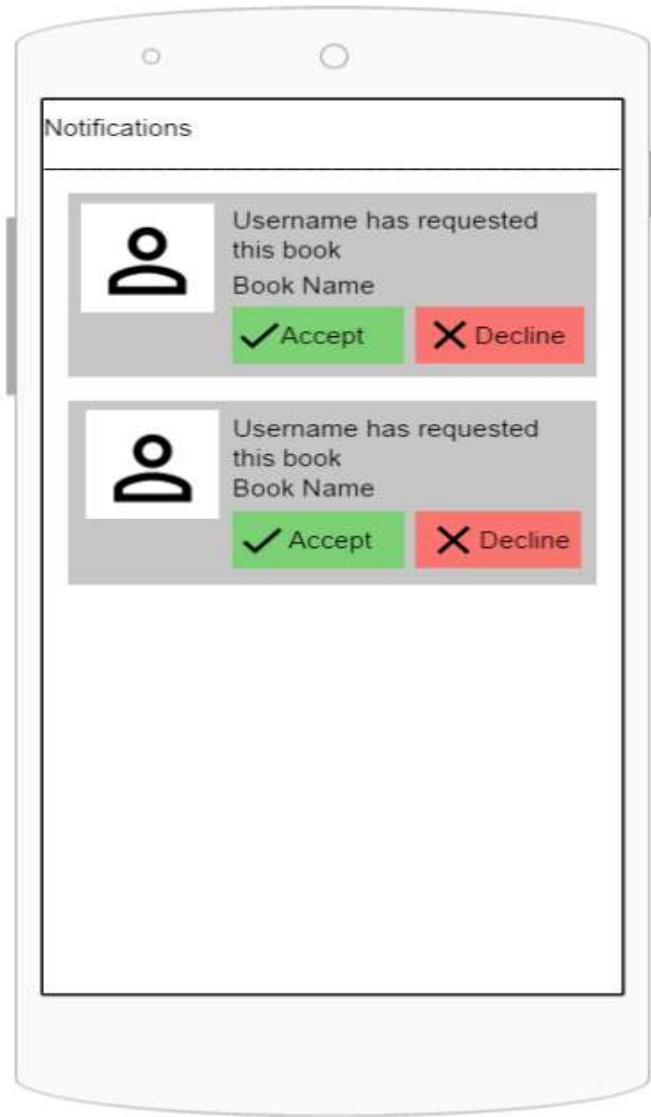


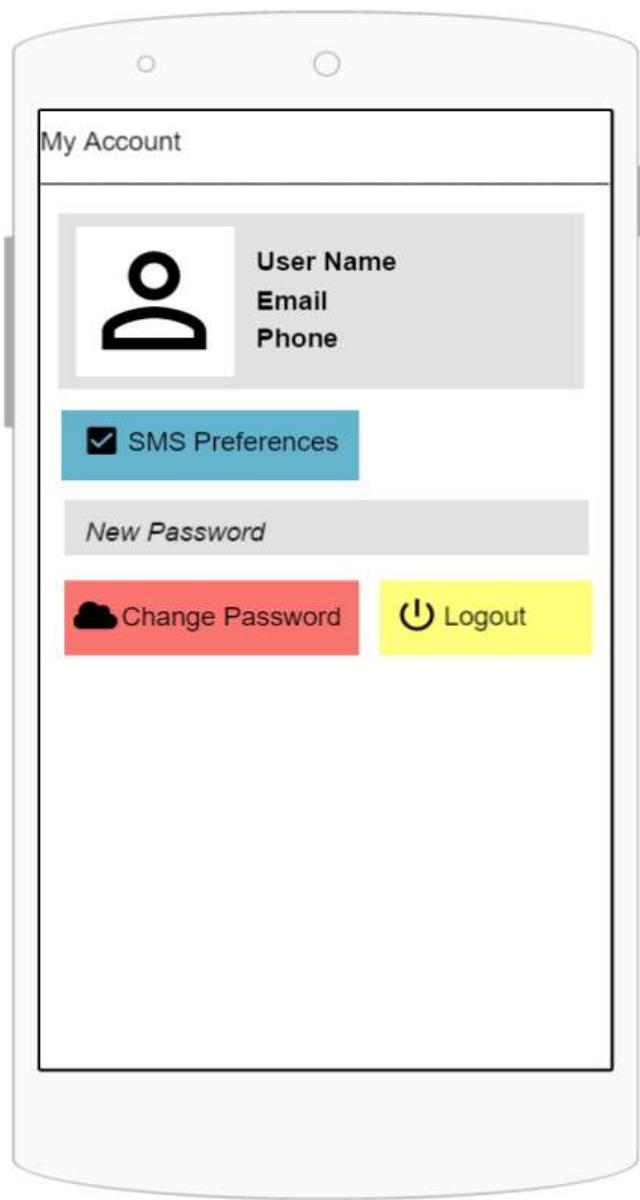


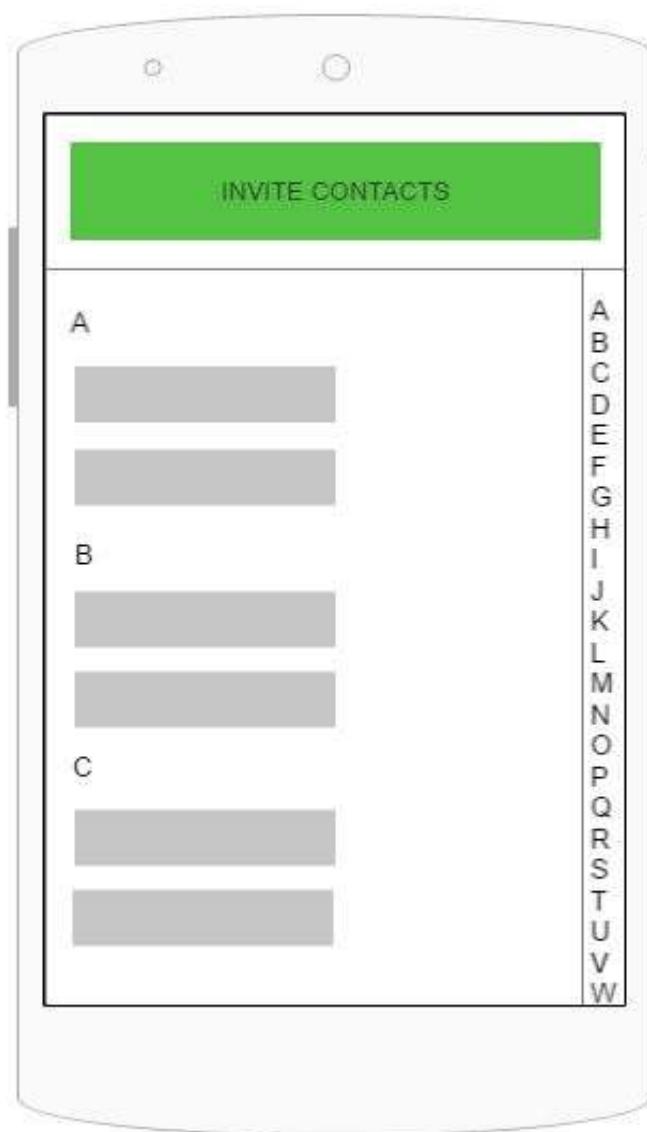


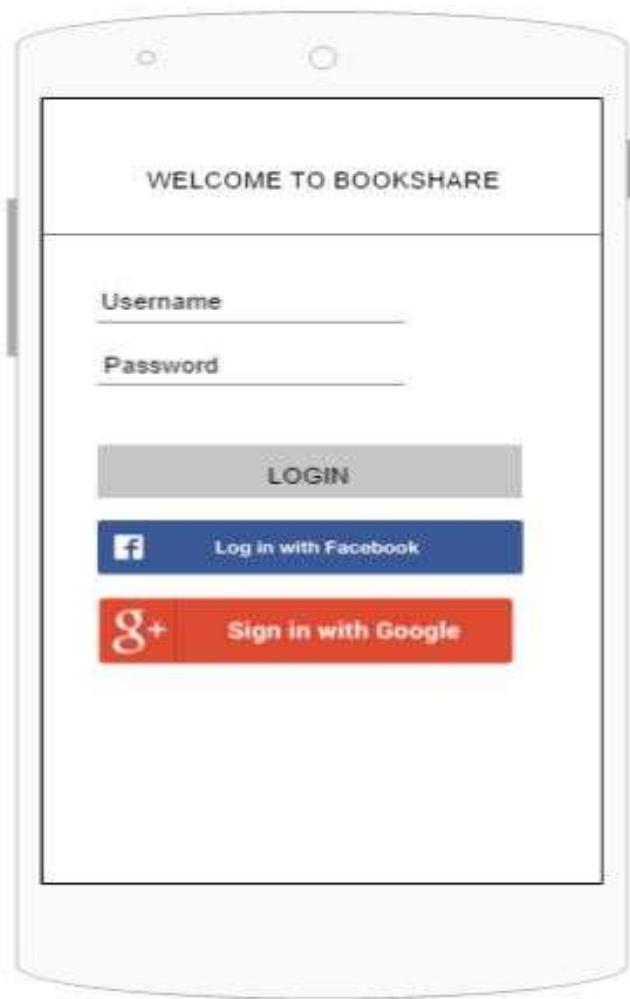




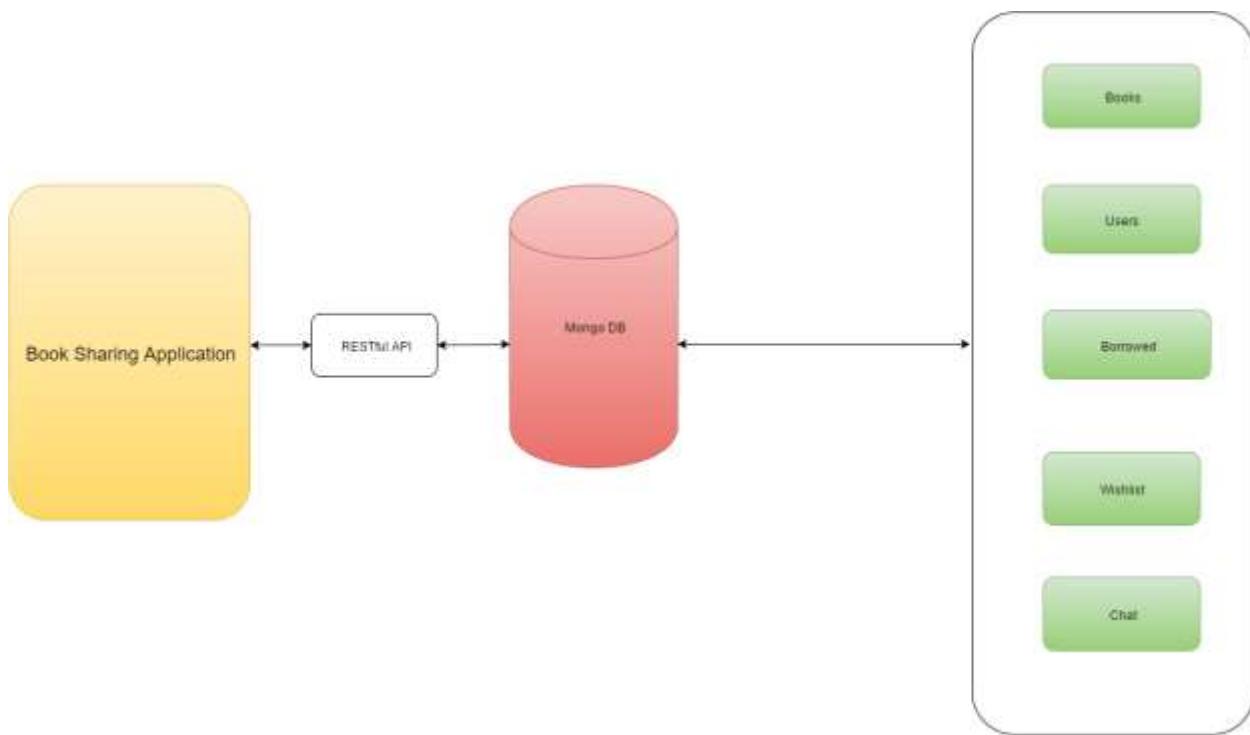




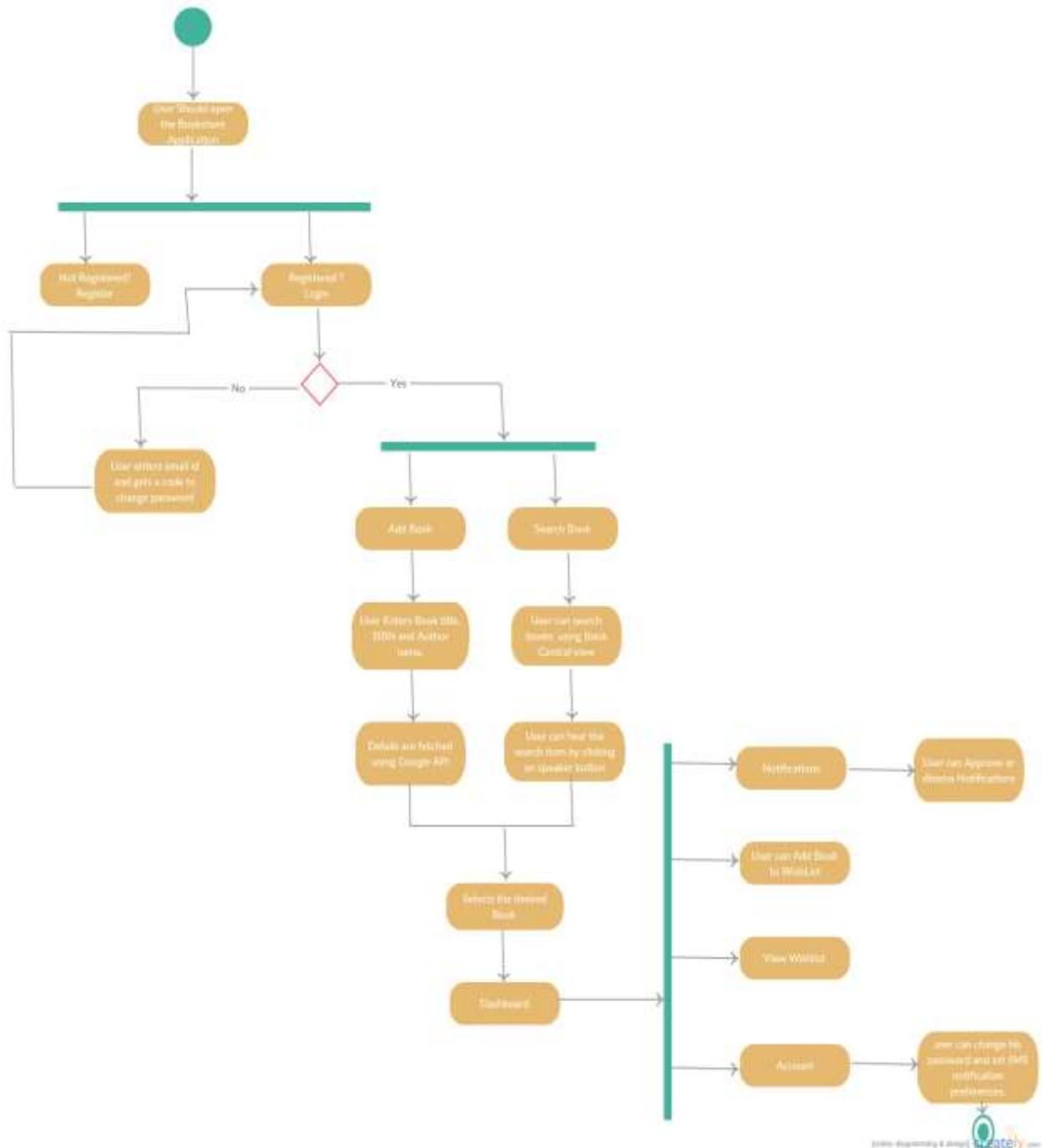




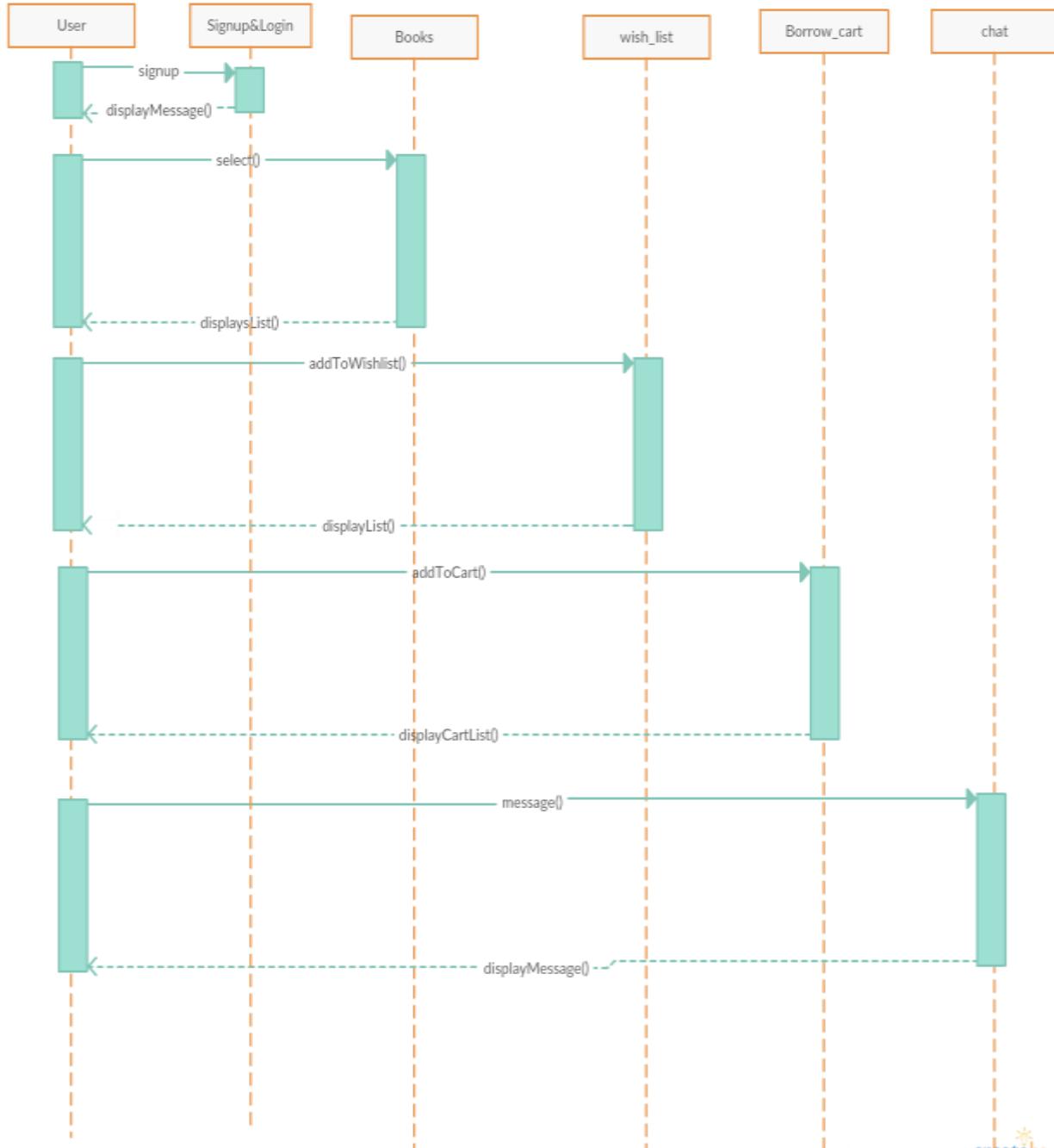
ARCHITECTURE DIAGRAM:



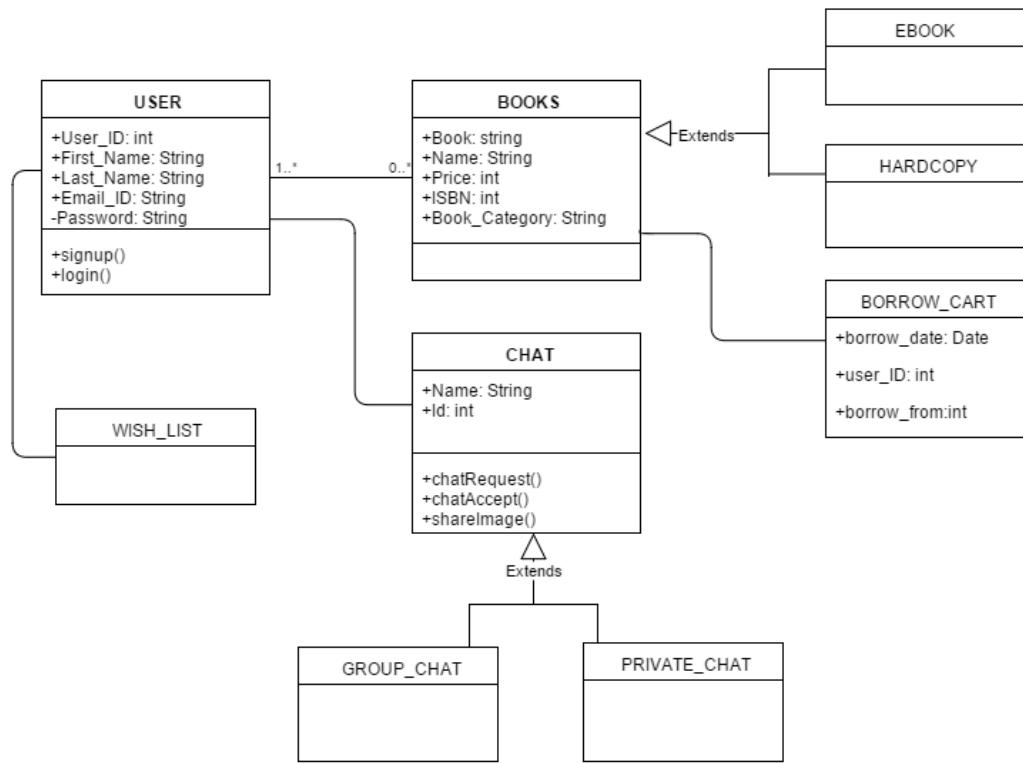
ACTIVITY DIAGRAM:



SEQUENCE DIAGRAM:



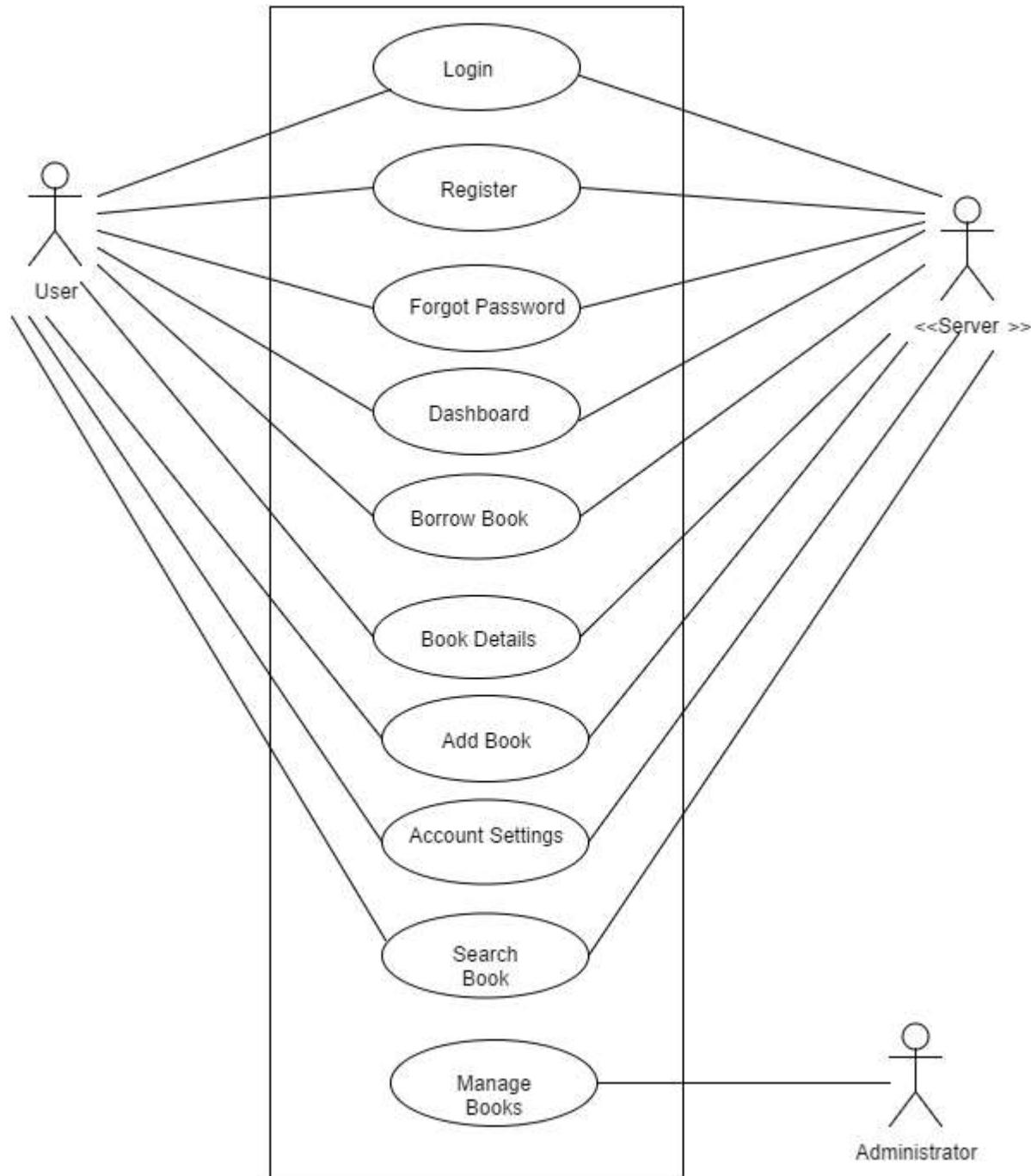
CLASS DIAGRAM:



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         it('should call login on userDashboard', function() {
13             expect(userDashboard.login).toHaveBeenCalledWith('pr3md@mail.umkc.edu', 'password1');
14         });
15
16         describe('when the login is executed,', function() {
17             it('if successful, should change state to dashboard', function() {
18                 expect(stateMock.go).toHaveBeenCalledWith('user-dash.home');
19             });
20
21             it('if login is unsuccessful, should show a popup', function() {
22                 expect(ionicPopupMock.alert).toHaveBeenCalled();
23             });
24         });
25     });
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 screenshot shows a browser window displaying the PageSpeed Insights dashboard. The URL in the address bar is `localhost:8100/#/dashboard/home`. The dashboard has three main sections: "Add New Book" (red), "My Books" (blue), and "Notifications 0" (green). Below the dashboard, the main content area shows a performance report for the URL `http://localhost:8100/#/dashboard/home`. The grade is **B** with an overall performance score of 76. The report highlights 8 external JavaScript scripts. A sidebar on the left lists optimization steps: **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, and **D** Put JavaScript at bottom.

Overall performance score 76 · Ruleset applied: YSlow(V2) · URL: <http://localhost:8100/#/dashboard/home>

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

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](#)

The screenshot shows a web browser window with the URL localhost:3100/#/dashboard/book-details/57054fa6f925ef11004bae2b. The page title is "How to Fix Everything For Dummies". On the left, there's a thumbnail image of the book cover. Below it, the book's details are listed: "How to Fix Everything For Dummies", ISBN: 0764572091, By Gary Hedstrom/Peg Hedstrom/Judy Ondrea Tremore/Judy Tremore - House & Home, Publisher: John Wiley & Sons.

A large text box contains a summary: "A guide to troubleshooting and fixing a wide variety of appliances, gadgets, and machinery shows readers how to repair lamps, vacuum cleaners, dishwashers, dryers, garbage disposals, blenders, washers, televisions, radios, and other household items." A small icon of a person is next to the word "disposals".

The screenshot shows the Lighthouse performance audit tool interface. The overall grade is "Grade B" with an overall performance score of 75. The audit is run against the URL <http://localhost:3100/#/dashboard/book-details/57054fa6f925ef11004bae2b>. The audit results for "Make fewer HTTP requests" are displayed, showing 8 external JavaScript scripts. The audit items listed are:

- 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

The main content area shows a summary: "Grade B on Make fewer HTTP requests" and a note: "This page has 8 external Javascript scripts. Try combining them into one." It also provides tips: "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." A "Read More" link is present.

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.

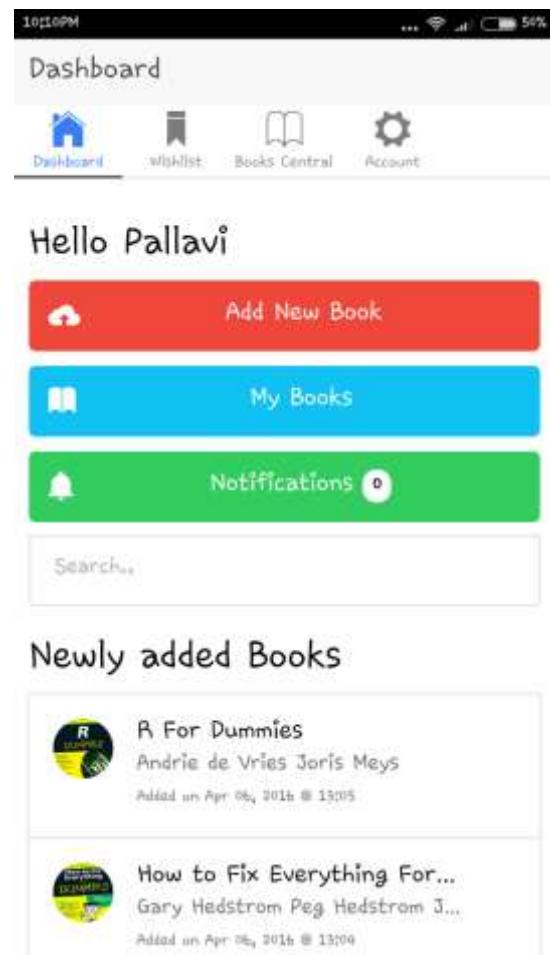
Service	Method	Service End Point	Request Payload	Response
Login	GET	https://floating-plateau-55000.herokuapp.com/bookshare/api/auth/login	Username, Password	Success – Fullname, Email
Register	POST	https://floating-plateau-55000.herokuapp.com/bookshare/api/auth/register	Email, Password, Full Name	Success – 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

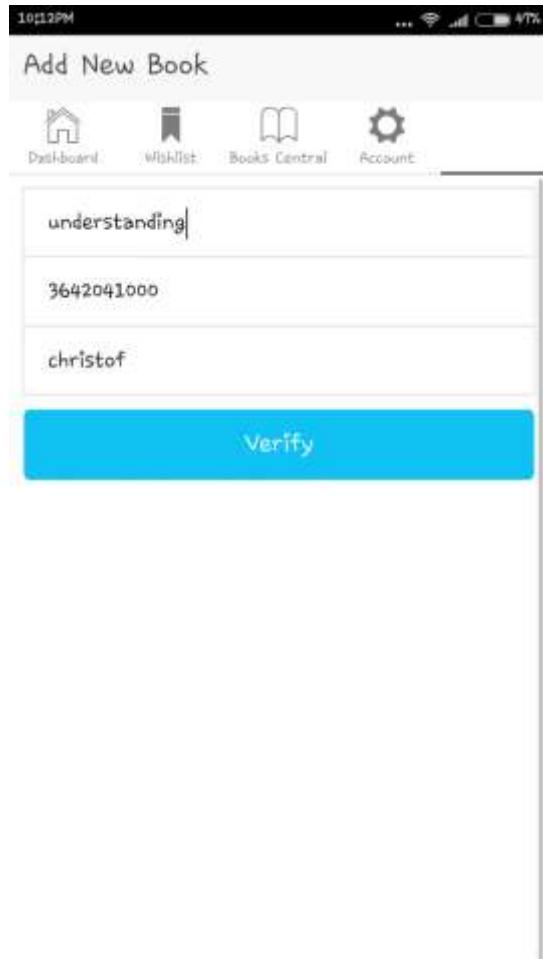
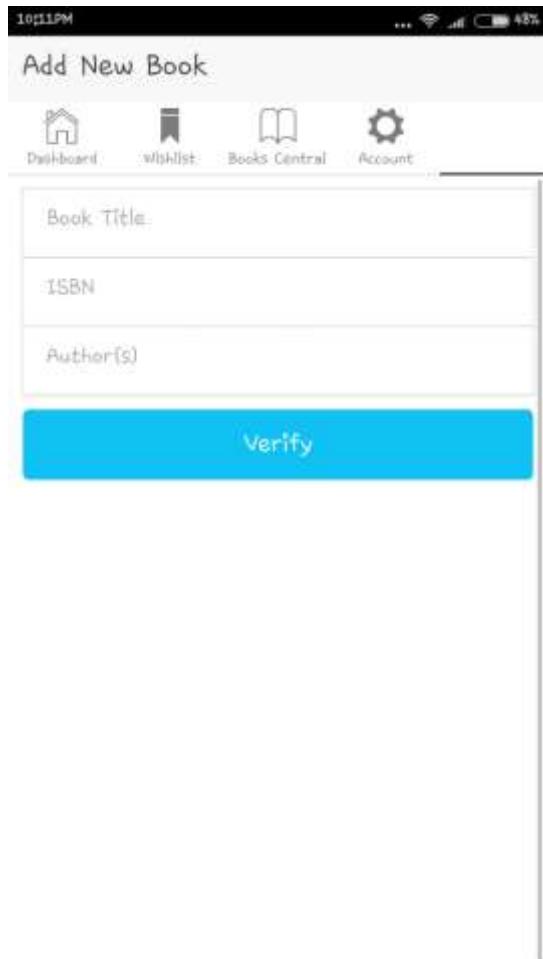
Books	GET	https://floating-plateau-55000.herokuapp.com/bookshare/api/allbooks	-	Book Title, Author, Owner

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:





10:12PM

Add New Book

Dashboard Wishlist Books Central Account

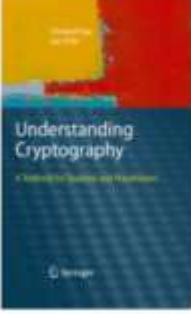
understanding

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

Add New Book

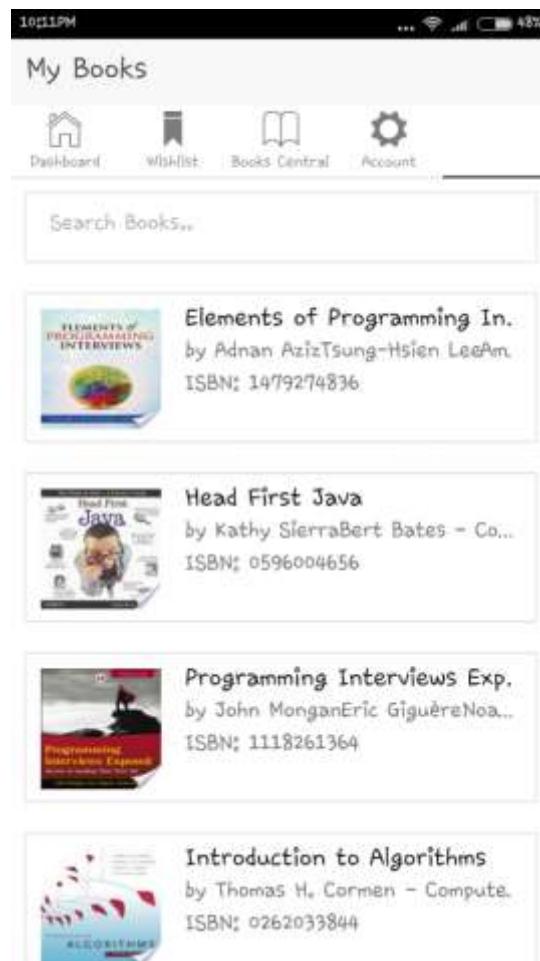
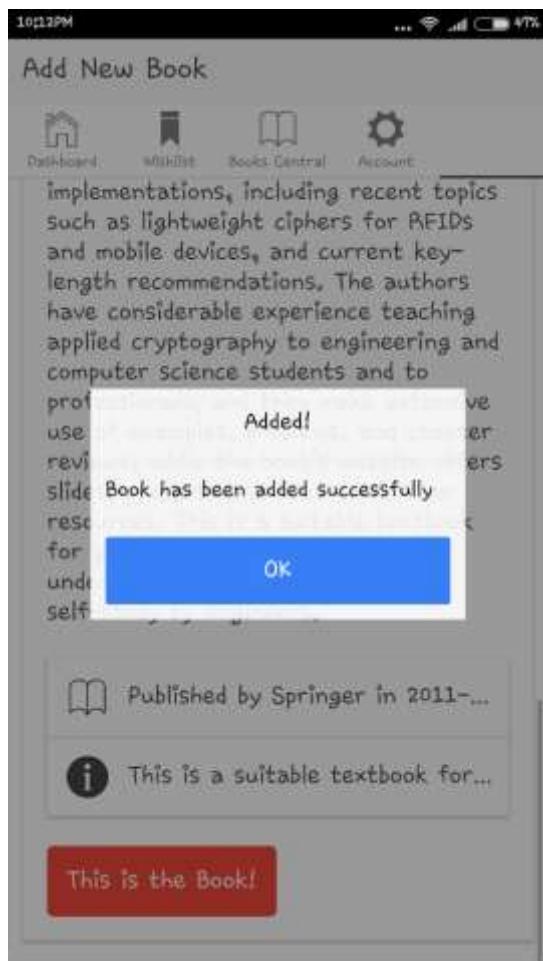
Dashboard Wishlist Books Central Account

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!



10:13PM

... WiFi 4G 47%

Understanding Cryptography

Dashboard Wishlist Books Central Account

Understanding Cryptography
ISBN: 978-3642041000
By Christof Paar, Jan 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

10:13PM

... WiFi 4G 47%

Understanding Cryptography

Dashboard Wishlist Books Central Account

Rating: 4 / 5
Price: \$49.95

Paperback: 372 pages
Popularity Rank: 28110

Buy from Amazon

Add to Wishlist

Reviews

This book is Available with 1 users.

Pallavi

10:14PM

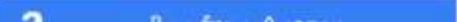
How to Fix Everything For Dummies

Dashboard Wishlist Books Central Account

 How to Fix Everything For ...
ISBN: 0764572091
By Gary Hedstrom, Peg Hedstrom...
Publisher: John Wiley & Sons

A guide to troubleshooting and fixing a wide variety of appliances, gadgets, and machinery shows readers how to repair lamps, vacuum cleaners, dishwashers, dryers, garbage disposals, blenders, washers, televisions, radios, and other household items.

Rating: 4 / 5
Price: \$19.99 
Paperback: 384 pages
Popularity Rank: 68829

... 

10:14PM

How to Fix Everything For Dummies

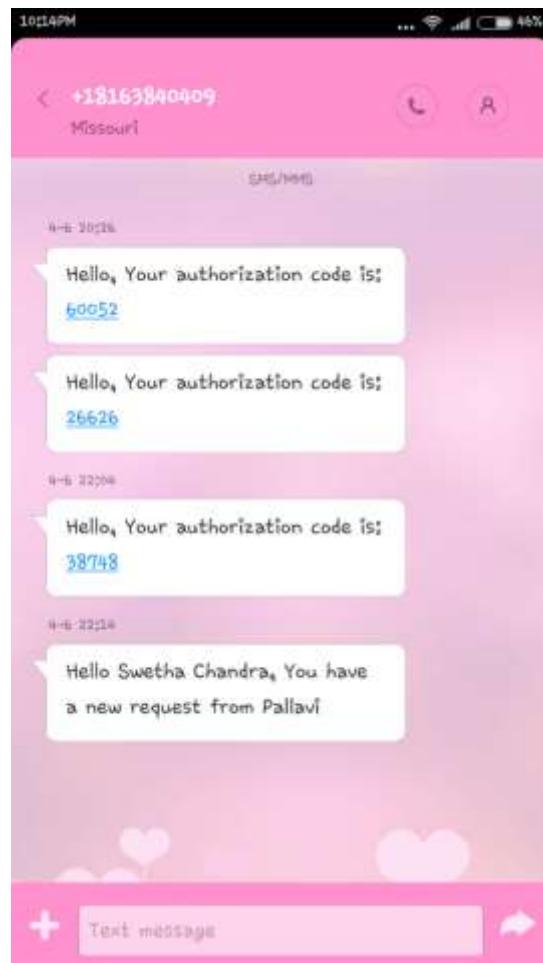
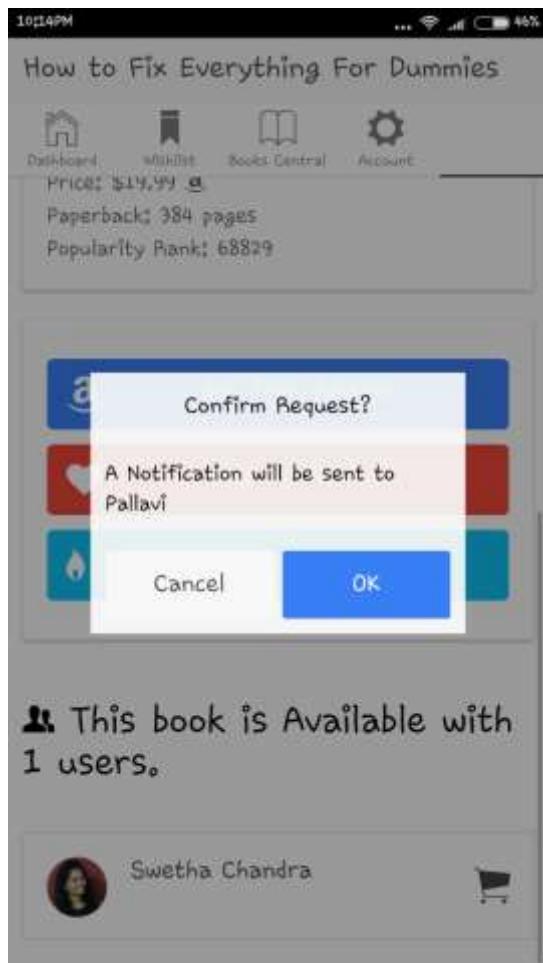
Dashboard Wishlist Books Central Account

Price: \$19.99 
Paperback: 384 pages
Popularity Rank: 68829

 Buy from Amazon
 Add to Wishlist
 Reviews

 This book is Available with
1 users.

 Swetha Chandra 



10:11PM ... WiFi 4G 49%

Dashboard

Dashboard Wishlist Books Central Account

Hello Pallavi

Add New Book

My Books

Notifications 1

Search...

Newly added Books



R For Dummies
Andrie de Vries Joris Meys
Added on Apr 06, 2016 @ 13:05



How to Fix Everything For...
Gary Hedstrom Peg Hedstrom J...
Added on Apr 06, 2016 @ 13:00

10:11PM ... WiFi 4G 49%

Notifications

Dashboard Wishlist Books Central Account

Swetha Chandra has requested ...
Apr 06, 2016 @ 22:10

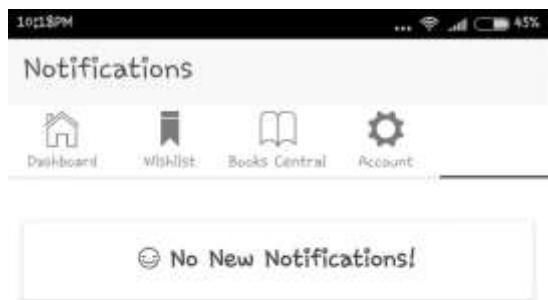
Getting to grips with R can be a tough, even for seasoned statisticians and data analysts. Enter R For Dummies 2e, the quick, easy way to master all the R you'll ever need. Requiring no prior programming experience and packed with practical examples, easy, step-by-step exercises and sample code, this extremely accessible guide is the ideal introduction to R for complete beginners. It also makes an excellent technical reference for experienced R programmers.

ISBN: 1119055806

Author: Andrie de VriesJoris Meys

Published by John Wiley & Sons in 2015-07-07





10:15PM

How to Fix Everything For Dummies

Dashboard Wishlist Books Central Account

Price: \$14.99

Paperback: 384 pages

Popularity Rank: 68829

Buy from Amazon

Book has been added to your Wishlist.

This book is Available with 1 users.

Swetha Chandra

10:15PM

... WiFi 4G 46%

My Wishlist

Dashboard **Wishlist** Books Central Account

Search Books...

 [How to Fix Everything For ...](#)
by Gary HedstromPeg Hedstro...
ISBN: 0764572091

10:14PM

... WiFi 4G 46%

My Account

Dashboard **Wishlist** Books Central **Account**

Pallavi
pr3md@mail.umkc.edu

New Password

 Change Password  Logout

DEPLOYING WEB APPLICATION:

A screenshot of a web browser window titled "BookShare". The address bar shows the URL "104.131.189.244/index.html#/". The page content displays a table with two columns: "Email" and "Full Name". The data rows are:

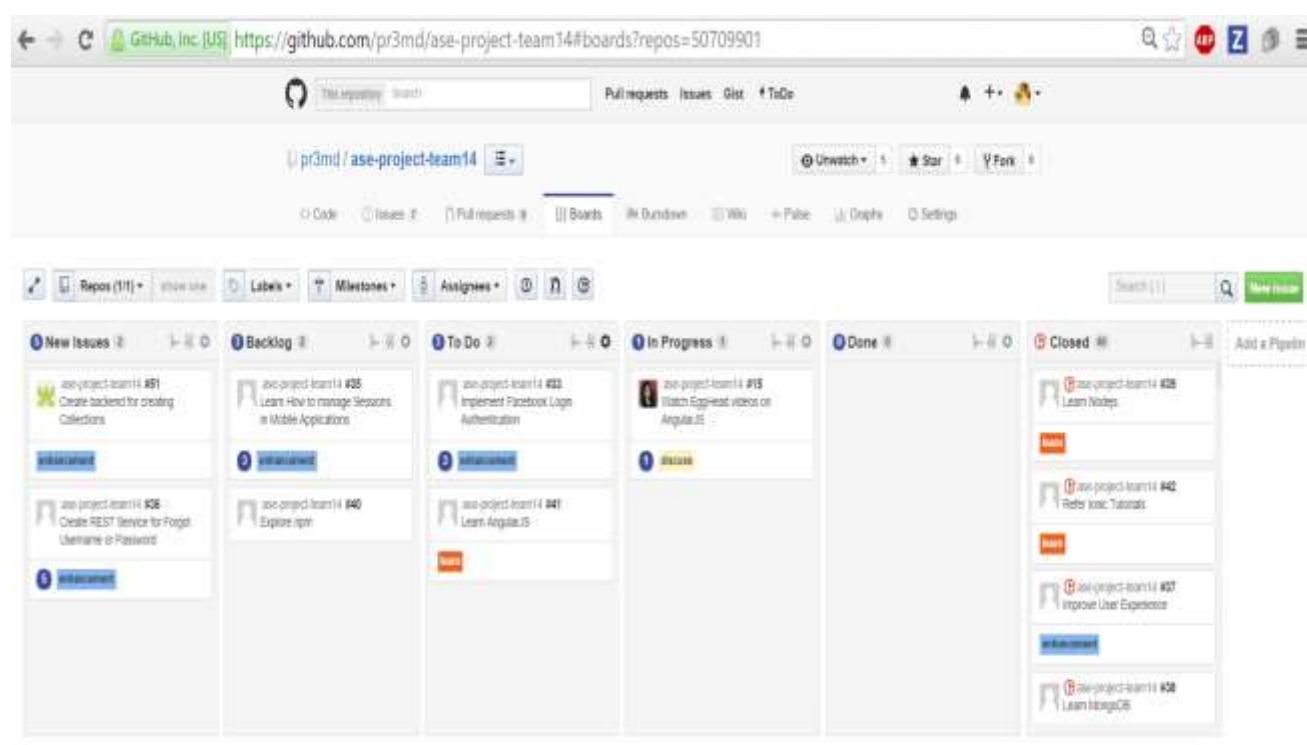
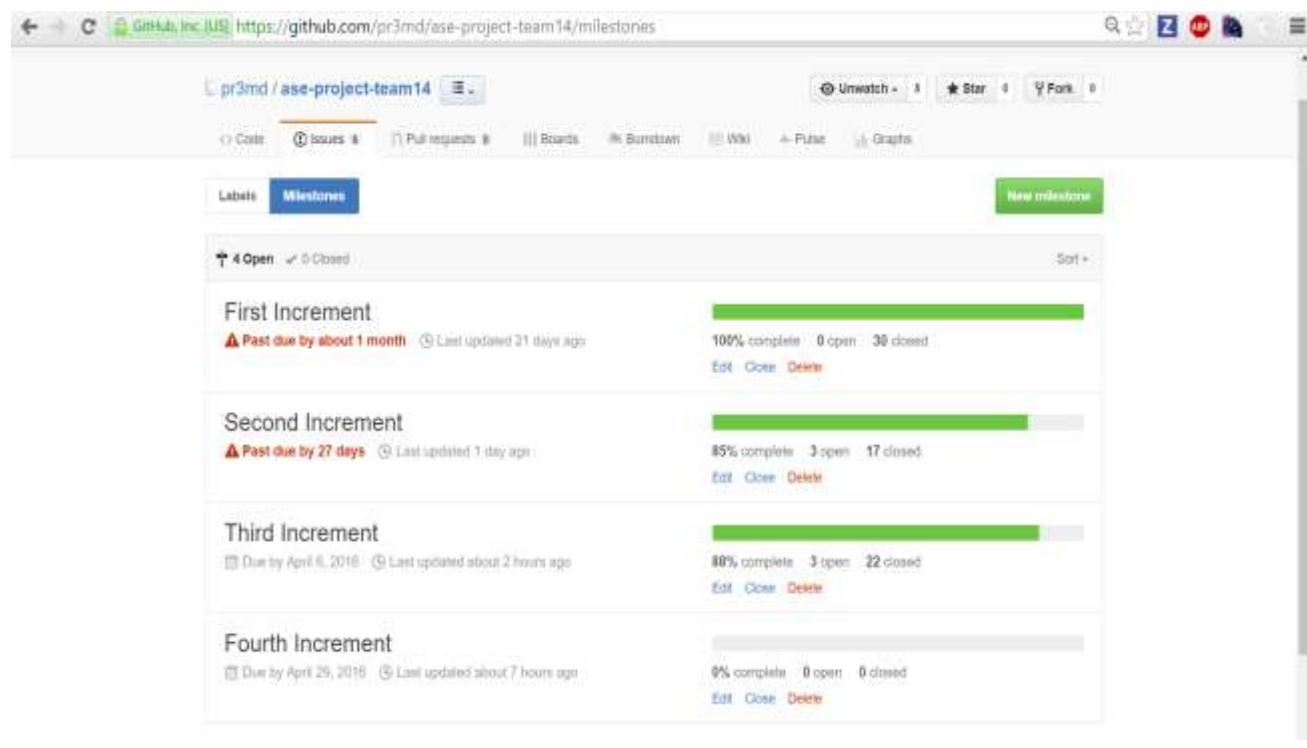
Email	Full Name
p03mg@mail.umkc.edu	Palavi
sckqn8@mail.umkc.edu	Swetha Chandra
swethachandra9494@gmail.com	Swetha Chandra K
palavi@gmail.com	RPalay
mrb9@mail.umkc.edu	Latha M
rph6@mail.umkc.edu	Raja Ramya

A screenshot of a web browser window titled "BookShare". The address bar shows the URL "104.131.189.244/index.html#/books". The page content displays a table with three columns: "Book Title", "Author", and "Added". The data rows are:

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 a GitHub repository page for 'pr3md/ase-project-team14'. The 'Issues' tab is selected. There are 6 open issues and 47 closed issues. The open issues listed are:

- Create backend for creating Collections [enhancement]
- Learn AngularJS [improvement]
- Explore npm
- Create REST Service for Forgot Username or Password [enhancement]
- Learn How to manage Sessions in Mobile Applications [enhancement]
- Implement Facebook Login Authentication [enhancement]

Each issue includes details like the author, creation date, increment it belongs to, and its status.

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.

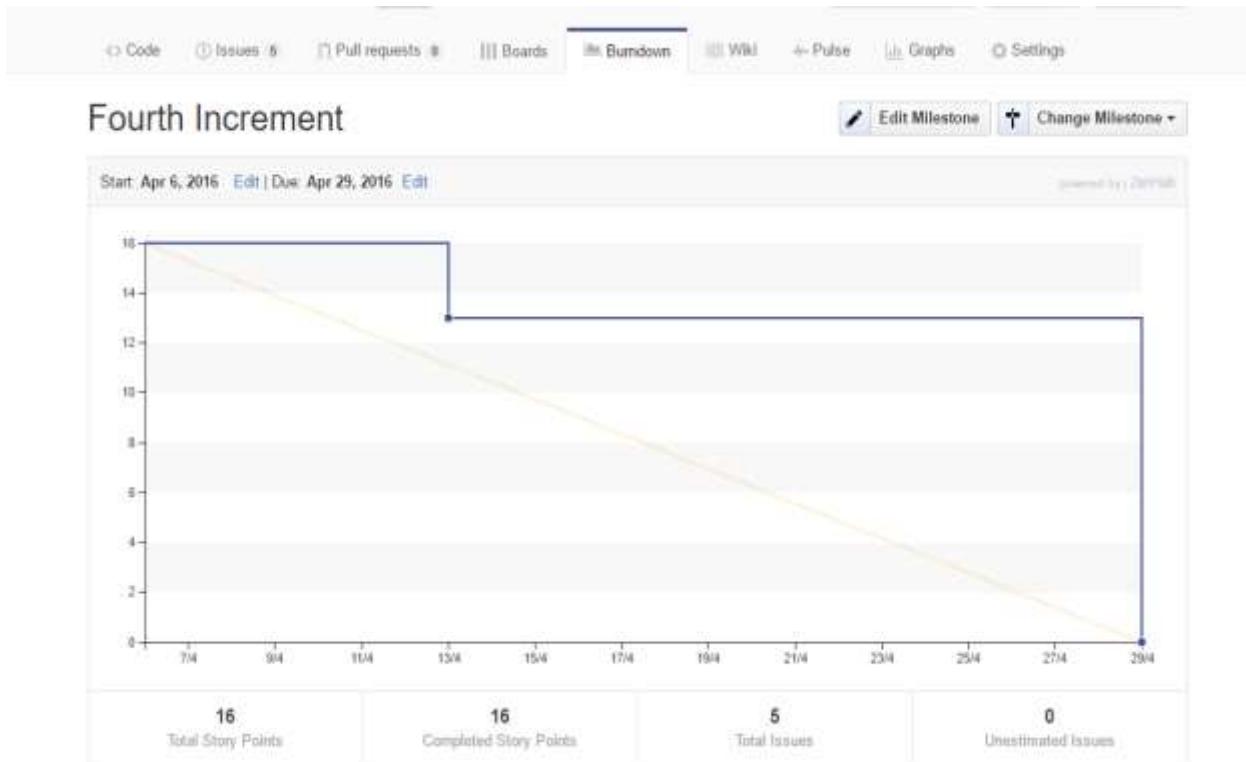
Service Design: All the user functionality has been built and the app is fully functional.

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.



BookShare

Latha Muddu-29

Pallavi Ramineni-49

Rajaramya Janagama-17

Swetha Chandra Karroti-21

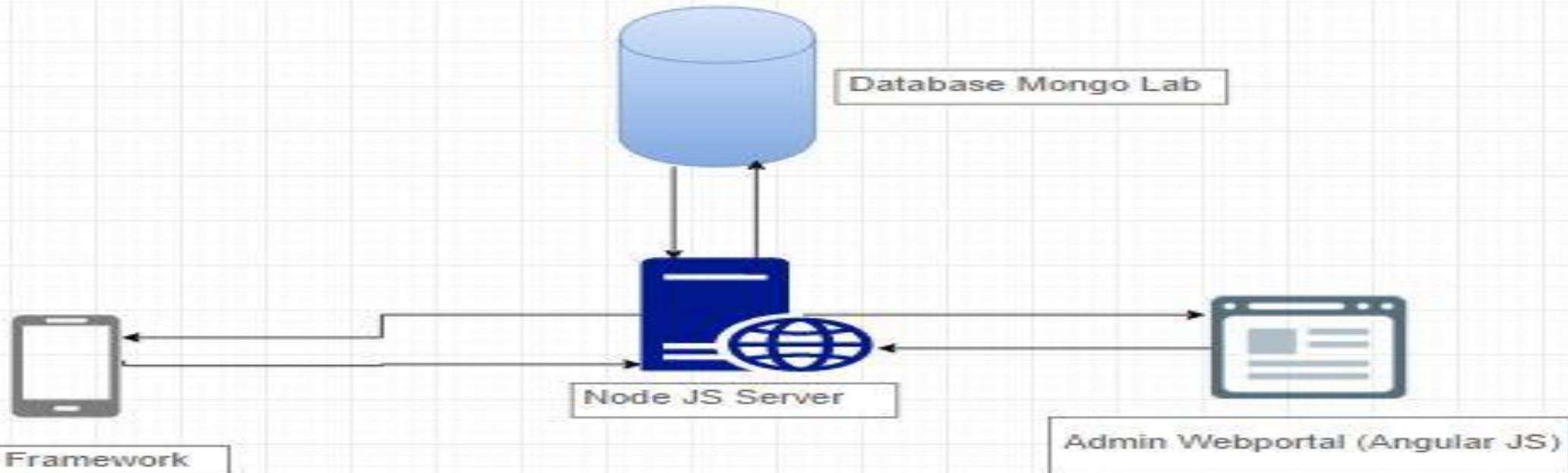
Motivation

- The motive of this application is to share books among registered users.
- With just one touch the user could select the desired book and request the users for the same.
- Our basic idea for selecting this project is to utilize time effectively and increase the accessibility of the books.
- Based on the various features available, user can select the best book he needs in an easy way.

Features

- Register & login
- Forgot Password and Reset
- Dash Board
 - Add New Book
 - My Books
 - Notifications
 - Invite Contacts
- Wish List
- Book Central
- Book Reviews
- Account
 - Change Password

System Architecture



Screenshots



A screenshot of the BookShare mobile application's dashboard. At the top, it shows the time as 10:10PM and battery level at 56%. The dashboard features a 'Hello Pallavi' greeting, three main buttons ('Add New Book', 'My Books', 'Notifications'), and a search bar. Below these are sections for 'Newly added Books' (listing 'R For Dummies' by Andrie de Vries Joris Meys) and 'Books Central' (listing 'How to Fix Everything For Dummies' by Gary Hedstrom Peg Hedstrom).

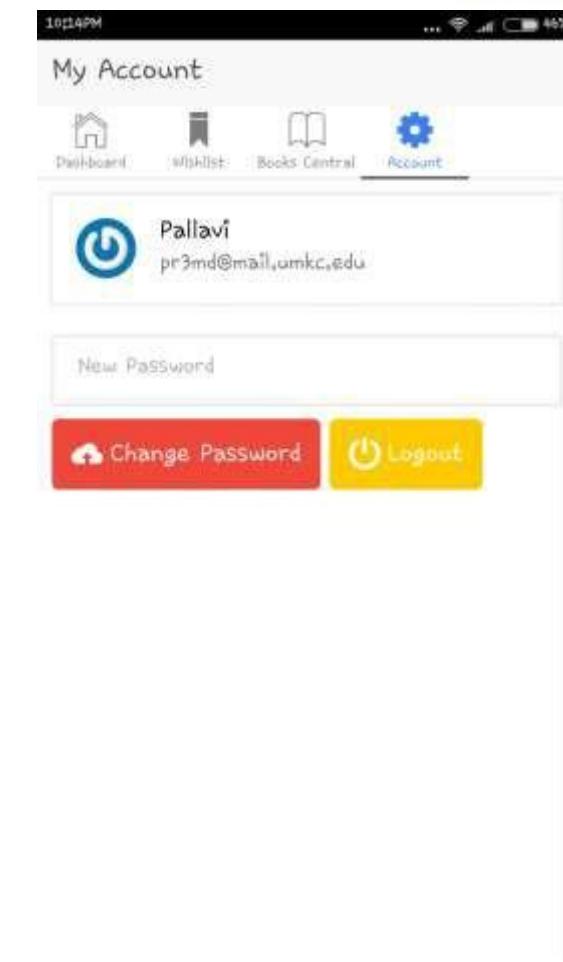
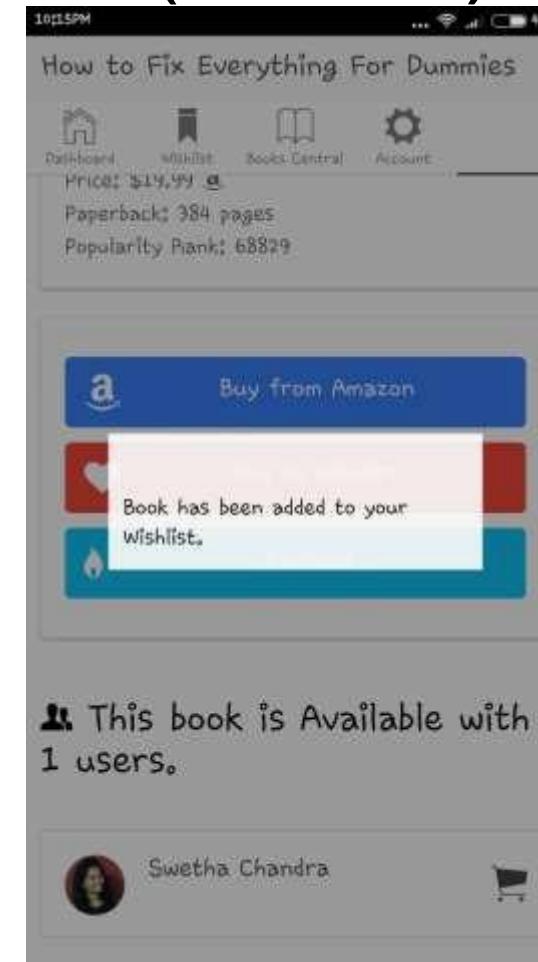
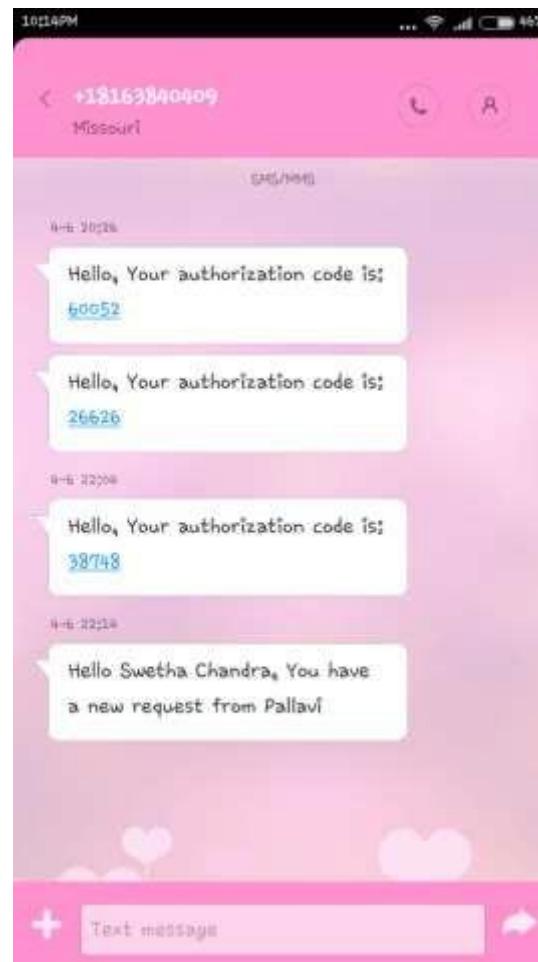
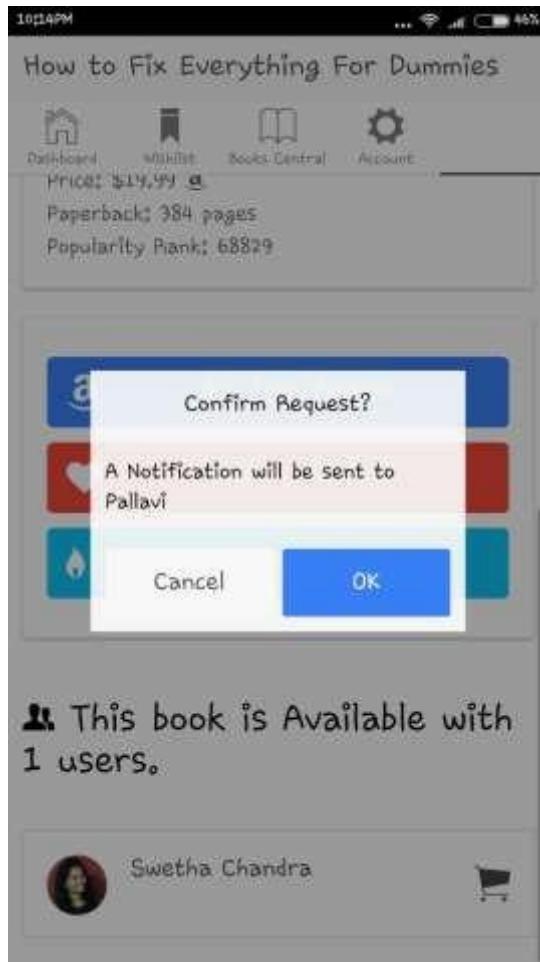
A screenshot of the BookShare mobile application's 'Add New Book' screen. At the top, it shows the time as 10:12PM and battery level at 47%. It has tabs for 'Dashboard', 'Wishlist', 'Books Central', and 'Account'. The main area contains fields for book details: 'understanding' (title), '3642041000' (ISBN), and 'christof' (author). A large blue 'Verify' button is at the bottom. Below this is a section for the book 'Understanding Cryptography' by Christof Paar.

A screenshot of the BookShare mobile application's book detail screen for 'How to Fix Everything For Dummies'. At the top, it shows the time as 10:14PM and battery level at 47%. It has tabs for 'Dashboard', 'Wishlist', 'Books Central', and 'Account'. The book details include: Price: \$19.99, Paperback: 384 pages, Popularity Rank: 68829. Below this are three buttons: 'Buy from Amazon' (blue), 'Add to Wishlist' (red), and 'Reviews' (light blue).

This book is Available with 1 users.

A screenshot showing a user profile card for 'Swetha Chandra'. It includes a small profile picture, the name 'Swetha Chandra', and a shopping cart icon.

Screenshots(Cont..)



THANK
YOU!