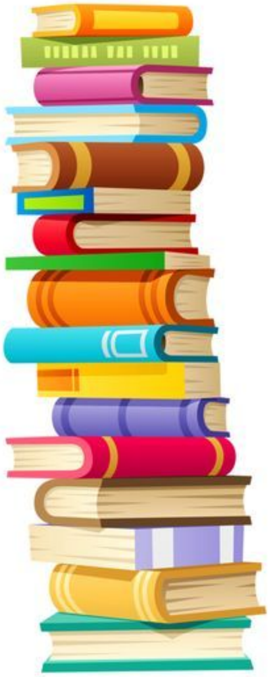


TOWER

Requirements Plan

Youssef Elmougy, Michael Cappeller, Liam Carlton, Raj Bedi

Version 3

Spring 2020

Table of Contents

Revisions

Removals

- Buyers will be able to create a wishlist of books they wish to purchase but may not be available on the TOWER. The user will be notified when this book becomes available.
- The Seller will not be able to contact the buyer as it is the responsibility of the buyer to contact the seller for their desired textbooks. A Push Notification will be sent to the buyer if a seller puts a desired book on the market.
- The TOWER will provide users with a friends list, allowing them to quickly contact users they have previously exchanged with
- A database will be used to store all offers currently proposed by sellers. A similar database will be used to contain the wish-lists of buyers.
- When a user is browsing through books, they will have the option to use a drop-down menu to sort based on the following options in ascending or descending order:
  - Alphabetical order of the **title** of the textbook.
  - Alphabetical order of the **author(s)** of the textbook.
  - The date the textbook was added.
  - Price of the textbook.
- TOWER will manage a database that contains all current offers proposed by users looking to sell books.
- TOWER will allow a user to upload at most 30 textbooks at a single time. Textbooks will be removed from the database after 2 weeks.
- TOWER will use an Advanced Encryption Standard (AES) algorithm in order to ensure that a user's username and password are secure.
- User review implementation to rate Sellers based on experiences.
- When this happens, the seller will be alerted with a pop-up on their home page. They will be prompted with two buttons: "accept" and "decline"

- Once the seller taps “accept”, they will be directed to a page where they can email the other user. The remainder of the transaction then takes place independent of TOWER.

1. Preface

This Android-based TOWER Platform, that is available primarily to the Hofstra community, will allow peer-to-peer textbook exchange without any fee charges. In the modern-day, the process of purchasing and receiving textbooks from e-commerce sites proves expensive and time-consuming. Through the use of this innovative platform, students are introduced to a cheaper and more efficient service that prioritizes their needs of quick transfer and exchange of books as well as potentially beating secondary market prices, hence enhancing their online experience.

2. Introduction

The purpose of this platform is to encourage reliable collaboration within the Hofstra community in promptly exchanging textbooks that are needed to be used for classes and/or research. TOWER allows certain students to declutter their desks of unnecessary textbooks and convert them into money, while also allowing other students to spend less time browsing online for the best deals. Moreover, they will be able to arrange a time to grab the book from another student on-campus. Initially, students are prompted to create an account and fill out their contact information that is used for exchange. Afterward, if a student wishes to post a book for sale on the platform, they are prompted to insert information regarding the book which may include its title, ISBN number, author, price, and condition. On the other hand, if a student wishes to browse for a book to buy, they can utilize the search request using information related to the book, and proceed to compare the options regarding the book’s price and condition. Once a student has pronounced interest in an option, the platform will facilitate a secure exchange where both the potential buyer and seller will be able to communicate via email, allowing them to arrange the details of their exchange.

3. Functional Requirements

3.1. User Requirements

3.1.1. TOWER will allow users to shop for used textbooks from fellow students.

3.1.2. Users who wish to participate in the textbook exchange app will have the ability to sign up to the app by providing their first and last name, Hofstra User ID#, their email address, and a password of their choice. Any user has the ability to buy and sell textbooks.

3.1.3. Sellers will be able to submit their used textbooks to the app indicating that they wish to sell it in which they would provide all the relevant details. These details are the Title, Author, ISBN-13, Condition, Price and Description. Users looking to buy a specific book will have the ability to browse the latest books added to the app or they can perform a search. Users can search based off the Title, Author, or ISBN-13

3.1.4. Users of TOWER who wish to purchase a textbook will have the ability to tap on a button to contact a seller. This button will redirect the user to their email app with the seller's email address already filled in.

3.1.5. TOWER will allow users to report the malicious conduct of other users that will be reviewed by TOWER administrators.

3.2. System Requirements

3.2.1. This application will be an Android platform based application written entirely in Java.

3.2.2. A Firebase database will be used to store the following information for each textbook:

- Title of the book
- Author(s) of the book
- ISBN-13 of the book
- Condition of the book
- Image URL of the book
- Price decided by the seller
- Description of the book
- Unique Identifier

3.2.3. Conditions of the book will include the following options:

3.2.3.1. Poor: a book that is poor may show visible signs of wear including stains, scuffs, rips, worn covers, writing or highlighting on the pages, and/or damage to the binding. The seller must make it clear to the buyer if any pages of the textbook are missing or are illegible.

3.2.3.2. Good: a book that is good will show usual signs of wear which may be minimal cosmetic damages such as scuff marks and creasing. There are no missing pages and there is little if any writing/highlighting on the pages.

3.2.3.3. Like new: A book that looks as good as new with very limited signs of wear. All of the pages are intact and there are no tears or creasing. The binding is in excellent condition and there is no writing or highlighting on any of the pages.

3.2.4. Once a buyer has selected an offer they are content with, they can tap a button labeled “Contact The Seller” that will redirect to a page in which they can email the seller.

3.2.5. TOWER will contain a page labeled “Home” which will contain the latest textbooks that users uploaded to the app.

3.2.6. TOWER will contain a page labeled “Search” in which the user can enter a title, author, or ISBN-13 to find all books matching their search criteria.

3.2.7. TOWER will manage a Firebase database that contains the account information of all users. This information will include:

- Full Name of the user
- Email Address of the user
- Hofstra ID (without the h)
- Password of the User

3.2.8. TOWER will utilize a user interface in order to display necessary information to the user in an organized manner.

3.2.9. TOWER will use the Google Books API to get additional information about books from its database. This information will include the description and the image url.

3.3. Visual Requirements

3.3.1. Upon logging in, the home page will contain the following buttons: “Home”, “Search”, “Profile”, “Settings”.

3.3.2. Tapping the “Home” button will bring up a grid of the 20 latest books added to the database.

3.3.3. On the search page there will be a search bar used to search for books based on title, author, or ISBN

3.3.4. TOWER will use the Google Books API to show information about the book including pictures and a description. If the book is not found in the Google Books API database, a user must enter all the information manually and the textbook cover will be set to the default image.

4. Non-Functional Requirements

4.1. Performance Requirements

4.1.1. TOWER will operate 24 hours a day, 7 days a week. Any downtime will be announced at least 48 hours in advance and will not affect any of the textbooks that a user has uploaded.

4.1.2. The search function will take at most 5 seconds to find a book in the database.

4.1.3. The database designed to store account information will be able to accommodate a minimum of 12,000 different users.

4.2. Operational Requirements

4.2.1. TOWER will be able to withstand thousands of simultaneous users on the application at any given time.

4.2.2. TOWER will be accessible and easy to use, even for those with visual impairment.

4.3. Security Requirements

4.3.1. TOWER will have administrators who will monitor the currently uploaded textbooks to ensure nobody has entered an invalid book.

4.3.2. TOWER will also feature a “report” function that any user can use if they find an invalid textbook. This will notify an administrator who can decide if the textbook should be removed.

4.3.3. An administrator has complete discretion over which books can be removed from TOWER.

4.3.4. Administrators will have no access to a user’s password. They will only have the ability to reset their password.

4.4. System Evolution

4.4.1. TOWER will allow buyers to communicate with sellers to barter a trade price.

4.4.2. Implement a machine learning algorithm that will predict invalid books automatically.

4.4.3. Find Books by Genre

5. User Stories

5.1. The user wants to buy "Book X"

5.1.1. Buyer signs into their account.

5.1.2. The buyer taps the "Search" tab.

5.1.3. Buyer types the name of the book they want (Book X in this case) into the search bar, and hits enter. This brings the buyer to a list of offers made by other users, all looking to sell Book X.

5.1.4. The buyer notices multiple copies of the same book and chooses the one with the best price and condition.

5.1.5. Buyer taps the offer that they prefer and are prompted with a button labeled "contact seller".

5.1.6. Buyer taps the "contact seller" button, which brings them to a page that allows them to email the seller. The remainder of the transaction then takes place independent of TOWER.

5.2. The user wants to sell "Book X"

5.2.1. Seller signs into their account.

5.2.2. The seller taps the "Add a Book" button.

5.2.3. Seller is redirected to a page where they enter the title of their book.

5.2.4. The Google API generates a list of the most relevant books closest to that search. The user finds that their book is in the Google database and taps another button labeled "This Looks Like My Book."

5.2.5. The application will automatically fill out the title, author, ISBN-13, and the description for the seller.

5.2.6. Seller will then provide the remaining information including

-Condition of the book

-Price

5.2.7. If the Google Books API doesn’t detect the book the seller wishes to sell, then the seller can enter all information manually:

-Title of the book

-Author(s) of the book

-ISBN-13 of the book

-Description of the book

-Price

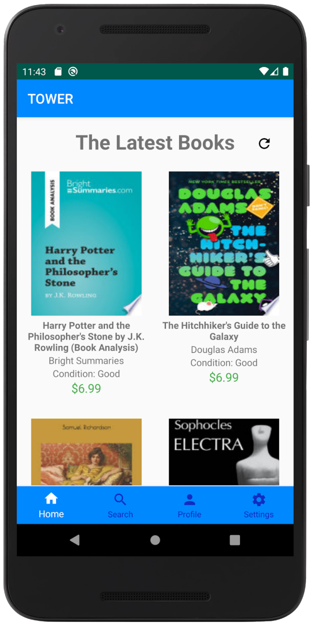
-Condition of the book

5.2.8. After filling in the blanks, the seller will tap the “submit” button at the bottom right of the screen. The seller’s offer is then added to the list that buyers can browse from.

5.2.9. The seller then waits for a buyer to accept their offer.

6. System Modelling

6.1. Home Screen

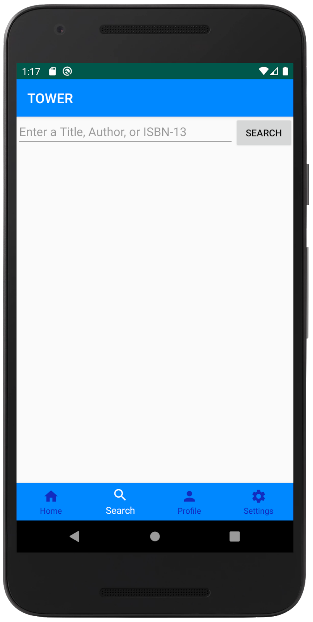


The Latest Books

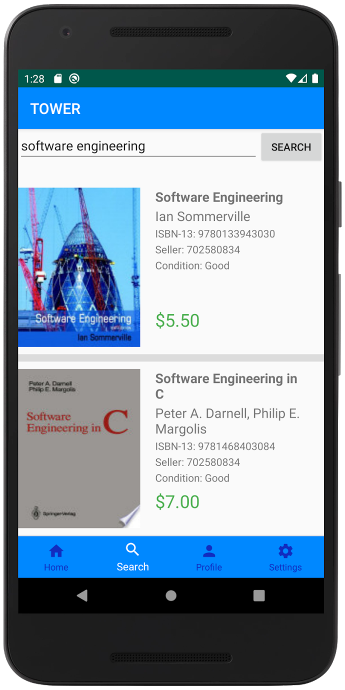
Harry Potter and the Philosopher's Stone by J.K. Rowling (Book Analysis)
Bright Summaries
Condition: Good
$6.99

The Hitchhiker's Guide to the Galaxy
Douglas Adams
Condition: Good
$6.99

6.2. Search Screen Before User Input



6.3. Search Screen After User Input

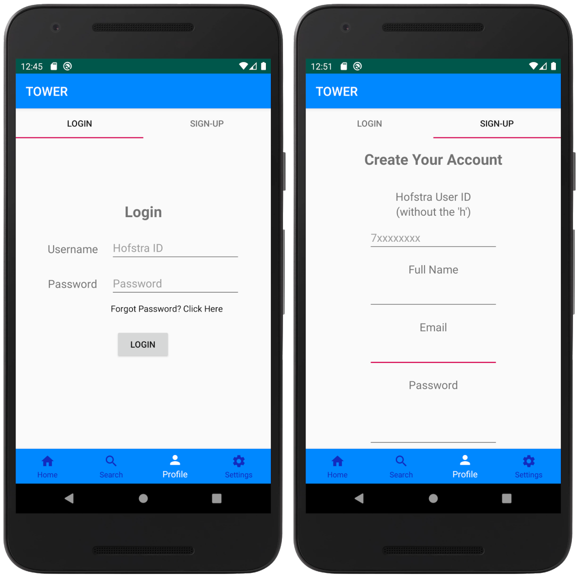


software engineering

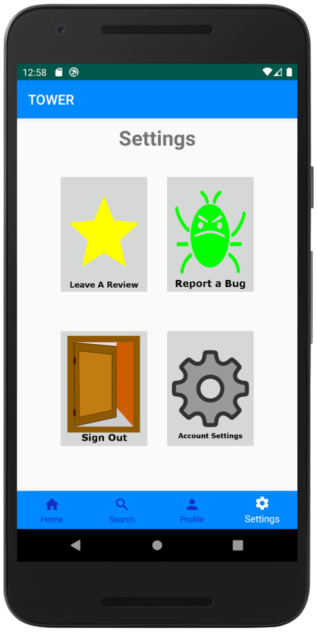
Software Engineering
Ian Sommerville
ISBN-13: 9780133943030
Seller: 702580834
Condition: Good
$5.50

Software Engineering in C
Peter A. Darnell, Philip E. Margolis
ISBN-13: 9781468403084
Seller: 702580834
Condition: Good
$7.00

6.4. Account Login/SignUp Screen



6.5. Settings Screen


12:58
TOWER
Settings
Leave A Review
Report a Bug
Sign Out
Account Settings
Home
Search
Profile
Settings