**yBOOK BARTER**

**CSE 5324: SOFTWARE ENGINEERING : ANALYSIS, DESIGN, AND TESTING**

**PROJECT GROUP 12: THE FIVERS**

**Iteration 1 - 10/06/2023**

****

**Team Members**

**APP TITLE: BOOK BARTER**

"Book Barter" is the solution to the age-old quest for the perfect book exchange. It connects book enthusiasts, addresses the need to find, exchange, and explore books effortlessly in a sustainable development way. Users register to establish a personalized reading identity, searching and browsing allows them to locate their next literary adventure. Messaging facilitates seamless communication and location services make in-person exchanges simple. Books Rating empowers readers to make informed choices, and safety & privacy features ensure secure interactions. Wishlist Management guarantees one will never miss a desired read. In short, "Book Barter" transforms book sharing into a hassle-free, enjoyable experience.

**Function 1: User Registration & Profile Creation:**

* Allows new users to sign up and existing users to log in. Profiles may include personal reading preferences (comedy, thrillers, fiction, etc.), book collection lists, and ratings or reviews.

**Function 2:** **Search & Browse Capability:**

* This enables the users to search for specific titles, authors, or genres and browse through available books based on categories or user recommendations.

**Function 3:** **Matching System:**

* Automatically matches users who have books desired by others, facilitating potential exchanges based on book availability and genre preferences.

**Function 4: Direct Messaging:**

* Aids users to communicate directly, discussing book conditions, barter terms, or arranging meetup/pickup location details.

**Function 5:** **Location Services:**

* Empowers the users to find potential barter partners nearby themselves, offering convenience and reducing the need for long-distance exchanges.

**Function 6:** **Book Reviews & Ratings:**

* Allows the lender to rate and review books they have read, offer insights, and help others make informed barter choices.

**Function 7:** **Notification System:**

* Notifies users of potential matches, new book listings in their preferred genre, or when someone expresses interest in their listed books. Users can unsubscribe from the notification service at any time.

**Function 8: Barter Confirmation:**

* Once a barter is agreed upon, users can confirm the exchange and update the status of the book(available or not), ensuring transparency and accountability.

**Function 9: Safety & Privacy Features:**

* This function ensures users' personal details remain confidential, offering features like safe meeting point suggestions and anonymous browsing.

**Function 10: History & Wishlist Management:**

* Users can keep track of past barters and manage or update their wishlist of books they're interested in.

**Function 11: Postal services:**

* This function introduces an option for users to request a book by using postal services like FedEx. To maintain transparency and integrity, user ratings and reviews are employed to gain trust

**Resources used for the application:**

**Database:**

* Specifically Firebase’s Cloud Firestore is used to organize and provide data to Users in real time

**Camera:**

* Camera service provided by Android is utilized for capturing the picture of the book for reference

**Location Services:**

* Location service offered by android is utilized for knowing the pickup location of the book lender

**Notification Services:**

* Notification service is utilized to notify about new books that match their genre, the status of their book request.

**Team’s Introduction:**

**Kushwanth Sai Bollepalli:** During my under graduation, with my enthusiasm for learning Android development, I created a few mini Android applications using Android Studio and Java. Subsequently, as a member of the Entrepreneurship Cell, a student club, I designed an Android app to streamline event registration and provide event details for an event called E-Summit. By working on that Android application I got familiarized with the Firebase’s Cloud Firestore database, Firebase Authentication, fragments, Intents, and various layout configurations.

**Srujan Chinta**: As a recent undergraduate with a strong passion for Android development, I've gained valuable experience through coursework, culminating in a comprehensive E-commerce app project. This project improved my knowledge of Java, Kotlin, and Android Studio while also exposing me to crucial concepts like UML design, user authentication, and user profiles. I've successfully included user profiles, UML design, and authentication into a simple e-commerce application. I created an organized and scalable application architecture using UML design principles. I also included user profiles, which enhanced the app's personalization and provided users with a seamless experience.

**Niharika Dandu:** During my undergraduate studies, I worked on an Android Studio project and created an app called Bus Pass Management System. Having a passion for creating unique and user-friendly mobile applications as an Android app developer. Using the robust platform, Android Studio, I've had the luxury of navigating the fascinating and always-changing field of Android app development. I am knowledgeable in both Java and Kotlin, which are the two main programming languages utilized in the creation of Android apps. I can write readable, streamlined, and maintainable code in both languages. My goal in developing apps was to transform original concepts.

**Sreelakshmi Mopuri:** During my undergrad, I started learning how to create apps. I became intrigued by the potential of mobile technology to have a significant positive impact on people's lives from the beginning, which drove me to create the "Toll Gate App For Android-Based Payment" app. To develop this app, I used the tools Android Studio, databases- Firebase Real-Time Database & SQLite/Room, and Java programming language. In addition, I excel in problem-solving, Testing and Debugging, Firebase, documentation, user experience design, API integration, App Deployment, version control, and database administration.

**Varad Nair:** With my passion for learning, during my under graduation, I created a few mini and a web app as a major project. It was a ‘Cloud Based Notes and Assignment Sharing system’ with the help of Web Development technologies, Postgres, and Django(Python) built on the ORM Model. I also have almost 2 years of work experience not only limited to development but analytics domain as well. I have knowledge of Java, Object Oriented Concepts, UML, hands-on Agile methodology, and DBMS. My critical assets are self-confidence, exposure to effective leadership, and problem-solving skills.

**Requirements:**

| Req ID | Requirement Description | Line Reference |
| --- | --- | --- |
| R1 | The system shall provide a registration / sign-up option for the new users | 10 |
| R2 | The system shall provide a log-in option for existing users | 10 |
| R3 | The system shall allow users to reset their password in case they forget | Derived |
| R4 | The system shall allow users to include or edit their personal reading preferences (Wishlist) like comedy, thrillers, fiction, etc in profile section | 10 to 11 |
| R5 | The system shall allow users to search for specific books based on book title, author name, or genre. | 14 |
| R6 | The system shall provide an option for users to request or initiate exchange | Derived |
| R7 | The system shall provide a chatting facility for users to discuss barter terms, book condition and meeting/pickup locations | 20 to 21 |
| R8 | The system shall access the location service to find potential barters nearby, willing to exchange | 23 |
| R9 | The system shall facilitate Book Lender’s to post new books along with their review and rating. | 26 to 27 |
| R10 | The system shall notify users about new book listings in their preferred genre, status of their request and if someone expresses interest in their listed books | 29 to 30 |
| R11 | The system shall allow users to unsubscribe from notification service | 30 to 31 |
| R12 | The system shall update the availability of book once barter is agreed upon | 33 to 34 |
| R13 | The system shall store and provide information about past barters | 39 |
| R14 | The system shall provide an option for users to request book, by using postal services like FedEx instead of fixing a pickup location | 42 to 43 |
| R15 | The system shall let users provide ratings and feedback about their experience with other users | 43 |

**Use Case List**

| Use Case # | Use Case Name |
| --- | --- |
| UC1 | New User Registration |
| UC2 | User Login |
| UC3 | Update User Profile & Wishlist |
| UC4 | Search for Books |
| UC5 | Book posted by User (Lender) |
| UC6 | Initiate Book Exchange |
| UC7 | Direct Messaging |
| UC8 | Barter Confirmation |
| UC9 | Utilize Location Services |
| UC10 | Rating and Feedback of Users |
| UC11 | Manage Notifications |
| UC12 | Manage Barter History |
| UC13 | Request Book via Postal Service |
| UC14 | Reset User Password |

**High-Level Use Case**

TUCBW: The use case begins with

TUCEW: The use case ends with

* **UC1: New User Registration**
* TUCBW the new users who can create an account by providing registration details and personal reading preferences
* TUCEW the new user is registered
* **UC2: User Login**
* TUCBW the user clicking the login button and entering details given during registration
* TUCEW user logging in successfully
* **UC3: Update User Profile & Wishlist**
* TUCBW users update their profile information and reading preferences (Wishlist)
* TUCEW user profile & wishlist have been updated
* **UC4: Search for Books**
* TUCBW enabling users to search for specific book titles, authors, or genres
* TUCEW list of books that match with the search keyword
* **UC5: Book posted by User (Lender)**
* TUCBW book lender fills out book name, book author, description, genre and rating along with the book cover picture and submits it
* TUCEW book details will be made public along with the ratings
* **UC 6: Initiate Book Exchange**
* TUCBW users express their interest in exchanging/receiving a book with another user by scrolling through their feed
* TUCEW users decide according to their interest
* **UC7: Direct Messaging**
* TUCBW communication between users to discuss book conditions, barter terms, or arrange meetup/pickup details
* TUCEW confirmation/denial of book exchange
* **UC8: Barter Confirmation**
* TUCBW users confirm the exchange/donation of books and update the status of the book (available or not), ensuring transparency
* TUCEW barter confirmation
* **UC9: Utilize Location Services**
* TUCBW  users can find potential barter nearby, reducing the need for long-distance exchanges
* TUCEW finding book barter
* **UC 10: Rating and Feedback of Users**
* TUCBW users rate other genuine users with feedback after the book is delivered
* TUCEW rating the users
* **UC11: Manage Notifications**
* TUCBW users customize or control their notifications
* TUCEW notifications being customized or turned on/off
* **UC12: Manage Barter History**
* TUCBW users keep track of past barters and view their history
* TUCEW maintaining barter history
* **UC13: Request Book via Postal Service**
* TUCBW users requesting a book using postal services like FedEx
* TUCEW book is being delivered
* **UC14: Reset User Password**
* TUCBW users can enter email for password reset
* TUCEW password being updated

**Use Case Diagram**

**Requirements to Use Case Traceability Matrix:**

| **Req ID** | **Priority** | **UC1** | **UC2** | **UC3** | **UC**  **4** | **UC5** | **UC6** | **UC**  **7** | **UC**  **8** | **UC**  **9** | **UC10** | **UC11** | **UC12** | **UC13** | **UC14** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **R1** | **2** | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **R2** | **3** |  | **X** |  |  |  |  |  |  |  |  |  |  |  |  |
| **R3** | **4** |  |  |  |  |  |  |  |  |  |  |  |  |  | **X** |
| **R4** | **4** |  |  | **X** |  |  |  |  |  |  |  |  |  |  |  |
| **R5** | **1** |  |  |  | **X** |  |  |  |  |  |  |  |  |  |  |
| **R6** | **1** |  |  |  |  |  | **X** |  |  |  |  |  |  |  |  |
| **R7** | **1** |  |  |  |  |  |  | **X** |  |  |  |  |  |  |  |
| **R8** | **2** |  |  |  |  |  |  |  |  | **X** |  |  |  |  |  |
| **R9** | **1** |  |  |  |  | **X** |  |  |  |  |  |  |  |  |  |
| **R10** | **3** |  |  |  |  |  |  |  |  |  |  | **X** |  |  |  |
| **R11** | **4** |  |  |  |  |  |  |  |  |  |  | **X** |  |  |  |
| **R12** | **2** |  |  |  |  |  |  |  | **X** |  |  |  |  |  |  |
| **R13** | **4** |  |  |  |  |  |  |  |  |  |  |  | **X** |  |  |
| **R14** | **1** |  |  |  |  |  |  |  |  |  |  |  |  | **X** |  |
| **R15** | **3** |  |  |  |  |  |  |  |  |  | **X** |  |  |  |  |
|  | **SCORE** | **2** | **3** | **4** | **1** | **1** | **1** | **1** | **2** | **2** | **3** | **7** | **4** | **1** | **4** |

REQUIREMENTS USE CASE TRACEABILITY MATRIX: Priority weights are assigned such that 1= highest priority weight and 4= lowest priority weight.R= Requirement ID UC = Use Case

**Increment Matrix:**

| **Use Case** | **Priority** | **Effort (person-**  **week)** | **Depends On** | **Assigned to** | **Iteration 1**  **(10-06-23)** | **Iteration 2**  **(11-03-23)** | **Iteration 3**  **(12-04-23)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **UC 1** | **2** | **2** | **None** | **VN,SM** | **1** | **1** |  |
| **UC 2** | **3** | **2** | **UC1** | **KB,ND** | **1** | **1`** |  |
| **UC 3** | **4** | **2** | **UC2** | **SC,VN** | **1** | **1** |  |
| **UC 4** | **1** | **4** | **UC2, UC5** | **SM,KB** |  | **2** | **2** |
| **UC 5** | **1** | **5** | **UC2** | **SC,ND** |  | **3** | **2** |
| **UC 6** | **1** | **3** | **UC2,UC5** | **VN,KB** |  | **2** | **1** |
| **UC 7** | **1** | **6** | **UC2,UC8** | **SM,ND** |  | **2** | **4** |
| **UC 8** | **2** | **3** | **UC2** | **SC,KB** |  | **2** | **1** |
| **UC 9** | **2** | **5** | **UC2,UC4** | **ND,VN** |  | **1** | **4** |
| **UC 10** | **3** | **4** | **UC2,**  **UC13** | **SM,SC** |  | **2** | **2** |
| **UC 11** | **7** | **3** | **UC1,UC5** | **KB,ND** |  | **1** | **2** |
| **UC 12** | **4** | **3** | **UC2,UC8** | **SC,VN** |  | **2** | **1** |
| **UC 13** | **1** | **2** | **UC2,UC8** | **SM,KB** |  | **1** | **1** |
| **UC 14** | **4** | **2** | **UC1** | **VN,SM** | **1** | **1** |  |
| **Total Effort** | | **46** |  |  | **4** | **22** | **20** |

1 Person-Week = 5 hrs.

**Team Members:**

VN- **V**arad **N**air

KB- **K**ushwanth Sai **B**ollepalli

SC- **S**rujan **C**hinta

ND- **N**iharika **D**andu

SM- **S**reelakshmi **M**opuri

**Domain Model:**

