TBR: Tackling ‘Tsundoku’ and ‘Choice Overload’ within the Reading Community

Venniz Sta Maria Valente

18/10/2024

## **1 Product Vision**

TBR is a web application that encourages recreational reading for the reading community. TBR helps eliminate Tsundoku through attainable reading goals that are user-preference-centred. This is achieved through features that allow users to archive a select number of books to be read within a select amount of time. TBR focuses on user preferences by considering the reading proficiency of each user to suggest a reading goal and timeframe that is feasible and gratuitous. TBR guides the user in selecting and allocating books that they want to read according to their reading mood. TBR encourages mindful expression through the journalling feature which allows users to express and recognise the thoughts and emotions they had while reading. TBR focuses on the elimination of issues such as Tsundoku and Choice Overload by motivating fun and manageable reading goals.

## **2 Analysis**

Reading has surged in popularity, particularly during the Covid-19 lockdown pandemic [[1]](#_References). Online communities formed through #BookTok, #Bookstagram, and #BookTube have thrived, especially with fiction novels at the forefront. The rise of "bookfluencers" has boosted sales for publishers. [[2]](#_References) Ridzuan's research found that BookTok users amassed extensive TBR (To Be Read) lists, highlighting a trend where readers buy more books than they can read. This leads to "Tsundoku," the practice of acquiring books but not reading them, resulting in overwhelming choices, and "reading slumps" [[3]](#_References). Chernev's research on "Choice Overloading" shows how too many options can paralyze decision-making [[4]](#_References). These issues have therefore resulted in modern solutions, Perschak's paper discusses how the application Goodreads deploys online reading challenges to tackle this issue whereas a simpler approach such as "book buying bans" aims to tackle this through the practice of self-restraint [[5]](#_References). The study by Ju, Lee, Kim, and Park indicates that physical artefacts can motivate reading but also create pressure when progress stalls [[6]](#_References).

Researching existing applications in the App Store [[7]](#_References) and Play Store [[8]](#_References) highlighted popular features and their resulting feedback from the reading community which helped me to identify potential improvements that TBR could implement for increased user satisfaction.

*Table 1 Gap Area Analysis*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Goodreads[[9]](#_References) | Fable[[10]](#_References) | Bookshelf[[11]](#_References) | Bookly[[12]](#_References) | Bookmory[[13]](#_References) |
| Facilitates adding books onto a shelf | x | x | x | x | x |
| Users can add a reading goal they wish to achieve | x | x | x | x | x |
| Users can adjust the length of their reading challenge |  |  |  | x | x |
| Prompts the user with periodic reminders to check in on their reading goal |  | x | x | x |  |
| Book selection from the unread shelf |  |  |  |  |  |
| Facilitates book journalling capabilities |  |  |  | x | x |

*x = feature is addressed*

## Table 1 indicates that "shelving" is a popular feature, allowing users to add books to their digital shelves. However, user feedback reveals this process is time-consuming, even with bulk scanning options. The Gap Area Analysis shows a lack of features encouraging users to select books from their TBR list, with most apps focusing on discovering new books, exacerbating tsundoku. It also highlights that while some apps let users set their reading timeframe, others, like Goodreads, have fixed annual timeframes. Notification features often receive negative feedback [[14]](#_References), as noted by Neff and Wintersberger, who found that the context and type of notification heavily influence user response, with many finding them repetitive and overwhelming.

## TBR aims to include features that help readers effortlessly decide which book to pick from their existing physical TBR list. This feature will focus on making the reading experience light-hearted and enjoyable, steering away from competitive challenges that often feel pressured. By incorporating playful elements and gentle nudges, my app will turn choosing a book into a fun and stress-free activity, helping readers rediscover the joy of reading without overwhelming choices.

## **3 Specification**

Laplante and Kassab [[15]](#_References) highlight three levels of abstraction in requirements: user, system, and design specifications. These can be categorised as functional and non-functional requirements. By analysing five existing applications, the user requirements for the TBR system were established: 1) Adding books to a TBR shelf. 2) Randomize the next book to read. 3) Reducing TBR size with reading challenges. 4) Enabling communication for reading clubs. 5) Journaling capabilities. Due to resource constraints, a survey with five reading community members prioritised user requirements 1, 2, and 5.

**3.1 Functional & Non-Functional Requirements**

Taking the user requirements and further categorising them between functional and non-functional requirements resulted in 17 Functional requirements and 16 Non-Functional requirements.

1. **Create Account:** Users can create an account with a username, email, and password.
2. **Login: Users** can log in using their credentials.
3. **Add to TBR:** Users can add books to their TBR shelf.
4. **Set Timeframe:** Users can set a reading timeframe for their TBR books.
5. **Start Reading:** Users can start the reading countdown.
6. **Finish Reading:** Users can stop the countdown upon completing their books.
7. **Start Book:** Users can be allocated a book from their TBR based on their reading mood.
8. **Finish Book:** Users can mark a book as read and add it to the read shelf.
9. **Reading Proficiency:** Users can select their reading level: beginner, intermediate, or expert.
10. **Display TBR:** Users can view all books on their TBR shelf.
11. **Display Read:** Users can view all books they have read.
12. **Check-in Note:** Users can add notes from their book.
13. **Display Notes:** Users can view all their check-in notes.
14. **Remove Book:** Users can remove books from their TBR shelf.
15. **Edit Note:** Users can edit their notes.
16. **Request Different Books:** The system prevents users from requesting a different book.
17. **Delete Note:** The system prevents users from deleting their check-in notes.

**Usability:**

1. Measure usability with a SUS scale.
2. Include a user feedback section.
3. Require user confirmation for

permanent actions

1. Ensure the system is easy to use and

navigate.

1. Keep the system simple, with no

unnecessary levels.

**Accessibility:**

1. Use contrasting colours.
2. Employ easy-to-read font and size.
3. Provide meaningful and relevant titles.
4. Offer simple instructions.
5. Identify interactive elements.
6. Ensure adequate spacing and content differentiation.

**Security:**

1. Use HTTPS for protection.
2. Avoid deprecated APIs.
3. Do not support notifications.
4. Enforce strong password criteria (uppercase, lowercase, number, special character, etc…).
5. Integrate password reset functionality.

### **3.2 Technical Choice**

The Stack Overflow 2024 developer survey [[16]](#_References) highlights trending web development components. Popular components are well-documented, supported, and frequently updated, adhering to industry standards. This enhances the success and scalability of the TBR web application.

*Table 2 Technical Choices*

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Choice | Reason | Alternative |
| Frontend Framework | React  [[17]](#_References) | React is a top choice in software development for its intuitive UI and flexibility, seamlessly integrating with other libraries and frameworks. | Vue.js |
| Frontend CSS Framework | React Bootstrap  [[18]](#_References) | React Bootstrap integrates seamlessly with React, enabling easy creation of responsive and attractive interfaces without complex CSS. | Tailwind |
| Backend | Python  [[19]](#_References) | Python remains a favourite for its strength, power, and readability. Clear, concise syntax makes coding and maintenance user-friendly. | Node.js |
| Database | MySQL  [[20]](#_References) | MySQL is widely used for its reliability and efficiency. Its popularity ensures extensive documentation and abundant resources. | MongoDB |
| Integration API | FastAPI  [[21]](#_References) | FastAPI is powerful, fast, and efficient, ensuring smooth performance under heavy loads. Its intuitive syntax and built-in validation keep everything running smoothly and error-free. | Flask |
| Book API | Google Books  [[22]](#_References) | Google Books offers a vast repository of book data with minimal limitations. Its flexibility and interoperability make it easy to integrate with various applications. | Open Library |
| IDE/Editor | VSCode  [[23]](#_References) | VSCode boasts an intuitive, user-friendly interface. Its extensive extensions enable customisation and enhanced functionality while remaining lightweight and fast. Additionally, offering seamless Git integration for version control and automation. | Visual Studio |

## **A screenshot of a graph Description automatically generated4 Roadmap**

## **GitLab Repo:** https://gitlab2.eeecs.qub.ac.uk/40297923/CSC3002\_TBR\_40297923

## **References**

1. Felice, B.B. (2024). Booktok and the Growth of Culture of Reading. [PDF] pp.1–60. Available at: https://scholarworks.calstate.edu/downloads/2b88qm397 [Accessed 1 Oct. 2024].
2. Ridzuan, M. (2023). The Phenomenon of Booktok and Its Impact on The Transformation of Literary Culture and The Publishing Industry. Forum Komunikasi, [online] 18(2), pp.74–83. Available at: https://forumkomunikasi.uitm.edu.my/images/FK18No2Oct23/FKOCT\_2023\_06\_BOOKTOK.pdf [Accessed 1 Oct. 2024].
3. Cambridge English Dictionary. (n.d.). Tsundoku. Retrieved from https://dictionary.cambridge.org/dictionary/english/tsundoku
4. Chernev, A., Böckenholt, U. and Goodman, J. (2015). Choice overload: A conceptual review and meta-analysis. Journal of Consumer Psychology, [online] 25(2), pp.333–358. Available at: file:///C:/Users/vvale/Desktop/computer%20science%20project/Examples%20-%20Dr%20Marie/Chernev-Choiceoverload-2015.pdf [Accessed 2 Oct. 2024].
5. Perschak , K.E.M. (2017). Heavy Readers and Their Unread Books. Explaining a Common Habit. [online] Literacy in the New Landscape of Communication – Research, Education, and the Everyday. Conference Proceedings of the 19th European Conference on Literacy. Alpen-Adria-Universität Klagenfurt, Austria , pp.69–75. Available at: https://www.researchgate.net/profile/Katharina-Perschak-2/publication/328449798\_Heavy\_Readers\_and\_Their\_Unread\_Books\_Explaining\_a\_Common\_Habit/links/5bcee6de299bf1a43d9a6f24/Heavy-Readers-and-Their-Unread-Books-Explaining-a-Common-Habit.pdf [Accessed 2 Oct. 2024].
6. Ju, S., Lee, K.R., Kim, S. and Park. Y.W., (2019). Bookly: An Interactive Everyday Artifact Showing the Time of Physically Accumulated Reading Activity. In: CHI ’19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. [online] pp.1–8. Available at: https://dl.acm.org/doi/pdf/10.1145/3290605.3300614 [Accessed 1 Oct. 2024].
7. Apple (United Kingdom). (n.d.). App Store - Developing for the App Store. [online] Available at: https://www.apple.com/uk/app-store/developing-for-the-app-store/.
8. play.google. (n.d.). How Google Play Works. [online] Available at: https://play.google/howplayworks/.
9. Goodreads (2010). Goodreads: Book Reviews. [online] App Store. Available at: https://apps.apple.com/gb/app/goodreads-book-reviews/id355833469 [Accessed 6 Oct. 2024].
10. Fable (2021). Fable: Books & TV. [online] App Store. Available at: https://apDamir Juretic (2019). Bookshelf-Your virtual library. [online] App Store. Available at: https://apps.apple.com/gb/app/bookshelf-your-virtual-library/id1464032274 [Accessed 6 Oct. 2024].
11. ‌ps.apple.com/gb/app/fable-books-tv/id1488170618 [Accessed 6 Oct. 2024].
12. ‌ Games, T. (2016). Bookly - TBR Book Tracker. [online] App Store. Available at: https://apps.apple.com/gb/app/bookly-tbr-book-tracker/id1085047737 [Accessed 6 Oct. 2024].
13. ‌ Dong su Mun (2020). Bookmory - book tracker & TBR. [online] App Store. Available at: https://apps.apple.com/gb/app/bookmory-book-tracker-tbr/id1515533482 [Accessed 6 Oct. 2024].
14. Neff, A. and Wintersberger, P. (2022). An Experience Sampling Study to Evaluate Why Users Dismiss Smartphone Notifications. In: MobileHCI ’22: Adjunct Publication of the 24th International Conference on Human-Computer Interaction with Mobile Devices and Services. pp.1–5.
15. Laplante, P. A. & Kassab, M., 2022. Requirements Engineering for Software and Systems. 4th ed.
16. survey.stackoverflow.co. (n.d.). 2024 Stack Overflow Developer Survey. [online] Available at: <https://survey.stackoverflow.co/2024/>.
17. Meta Open Source (2024). React. [online] react.dev. Available at: https://react.dev/.
18. react-bootstrap.netlify.app. (n.d.). React Bootstrap | React Bootstrap. [online] Available at: https://react-bootstrap.netlify.app/.
19. Python (2019). Python. [online] Python.org. Available at: https://www.python.org/.
20. Oracle (2024). MySQL. [online] Mysql.com. Available at: https://www.mysql.com/.
21. FastAPI (n.d.). FastAPI. [online] fastapi.tiangolo.com. Available at: https://fastapi.tiangolo.com/.
22. Google Developers. (n.d.). Google Books APIs. [online] Available at: https://developers.google.com/books.
23. ‌ Microsoft (2024). Visual Studio Code. [online] Visualstudio.com. Available at: https://code.visualstudio.com/.