



**Computer Science Department
Final Project Report – Spring 2024**

Title of the project:	“BamBook: Personalized Book Recommendation and Engagement Platform”
Team Members:	Dias Saiynov Nurzhan Nurtayev Dauren Amanbek Shakhnazar Shayakhmet Yeldar Shayakhmet
Project Advisor/Co-Advisors	M. Fatih Demirci
Executive Summary (10%)	
<p>In this report we are going to present a developed book recommendation mobile application system “BamBook”. Purpose of BamBook is to address the gap in digital platforms offering comprehensive book recommendations. Our mobile application is powered by Go, Python and Swift programming languages, while utilising Google’s YouTube Retrieval model for recommendation feature, trained on GoodReads dataset. In the process of development, we encountered challenges mostly in choosing, integrating the model, handling extensive datasets and optimising for the iOS platform. Our system architecture was improved and an iterative method was used to overcome these problems. Although our product still needs more improvements in both back and front parts, evaluation from user’s shows that BamBook meets our main objective which is to provide for users an application with easily understandable interface and relevant recommendation.</p>	
Introduction (10%)	
<p><i>Problem description:</i></p> <p>Today, on the scale of a journey from deciding to read to actually reading, there is a wide group of people that lose an intention to read, since there is not any platform to provide a qualitative, reliable, objective compilation of literature, while not pursuing any commercial or personal goals. People face challenges trying to find the right book.</p> <p><i>Project motivation:</i></p> <p>Our primary motivation is to develop an easy-accessible application, which strives to simplify the user's journey towards his own preferences and knowledge that perfectly suits his goals. Aiming to make this step of finding a perfect book much more comfortable, we believe that there might be much more readers, therefore, much more self-aware, competent and culturally reinforced people.</p>	
Background and Related Work (15%)	
<p><i>Models:</i></p> <p>We have researched many recommendation models and implementations: including Facebook’s Deep Learning Recommendation Model, DeepFM, Wide & Deep, Google’s Neural Collaborative Filtering (NCF)</p>	



among other simpler variants. Due to complexities, certain issues and code availability, which are elaborated later in the report, we decided to settle on Google's YouTube Retrieval model.

Related platforms:

When brainstorming the issue, we have examined existing platforms with recommendations for readers. One of the main "players" is Goodreads. The platform takes users' email and preferred genres, then asks to rate books and provides recommendations based on the input. Users can create their own book "shelves" with various book groups, the default shelves are "Want to read", "Currently reading" and "Done", which makes it easy to sort various literature. There is a great base of books and reviews and several options to discover literature in the "Discover" section. Yet, Goodreads provides mostly English book recommendations and is transferring to other platforms for purchases, not letting users to read/purchase books right on that platform. It also starts to suggest recommendations only after you have rated 20 books, neglecting those who did not read or can not recall all 20 books. Nevertheless, it was a good foundation for our application.

Project Approach (20%)

Tools used to develop our mobile application would include:

- Programming language: Swift, Go, Python
- Integrated Development Environment (IDE): Xcode, Jupyter Notebooks, Pycharm, GoLand
- Libraries: UIKit, Alamofire, SDWebImage, SwiftyJSON, LibRecommender
- Server and database technology:
 - Backend server: Go with standard library,
 - Model server: Python with Sanic, Database: PostgreSQL
- Hardware: laptops, smartphones with iOS

System architecture:

The architecture consists of three components: Backend#1 running the recommendation model, Backend #2 serving user, book, recommendation data, and the mobile iOS app.

The recommendation algorithm uses YouTube Retrieval model to serve recommendations, which is trained on [Goodreads](#) dataset. The dataset contains a book collection and users' interaction data with books. The book collection is used to construct a database on PostgreSQL that contains detailed data on them. The interaction data is used to train the recommendation model.

The backend stores the book collection and serves them to the user based on recommendations retrieved from the model server.

App workflow:

Authentication Screens where users can either login or register a new account. In the login section, the user enters information like email and password. Then these values pass to the backend server and the application gets either successful login or error (ex. Incorrect password).

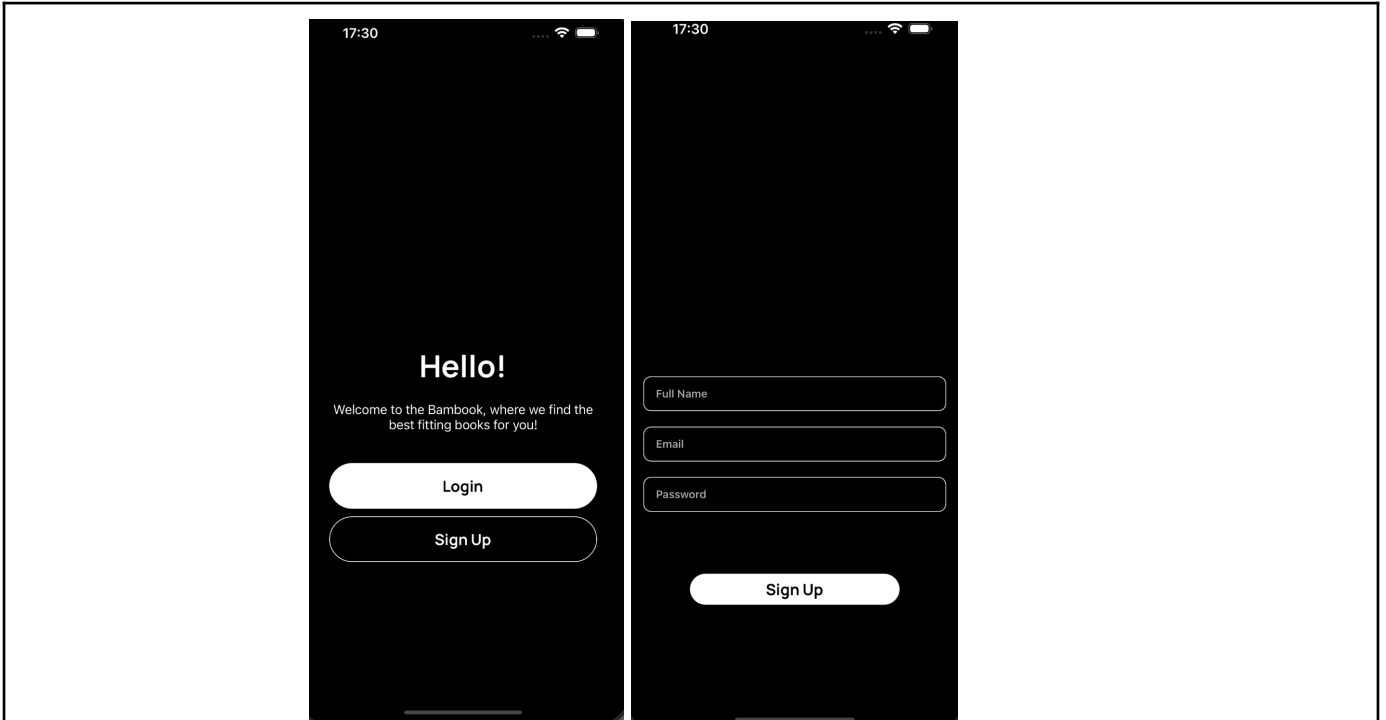


Fig #1. Authentication Screens.

The next step for the user is to write book names that the user read/liked. These books will be passed to the backend server as user preferences and will be used to evaluate recommendations in our model.

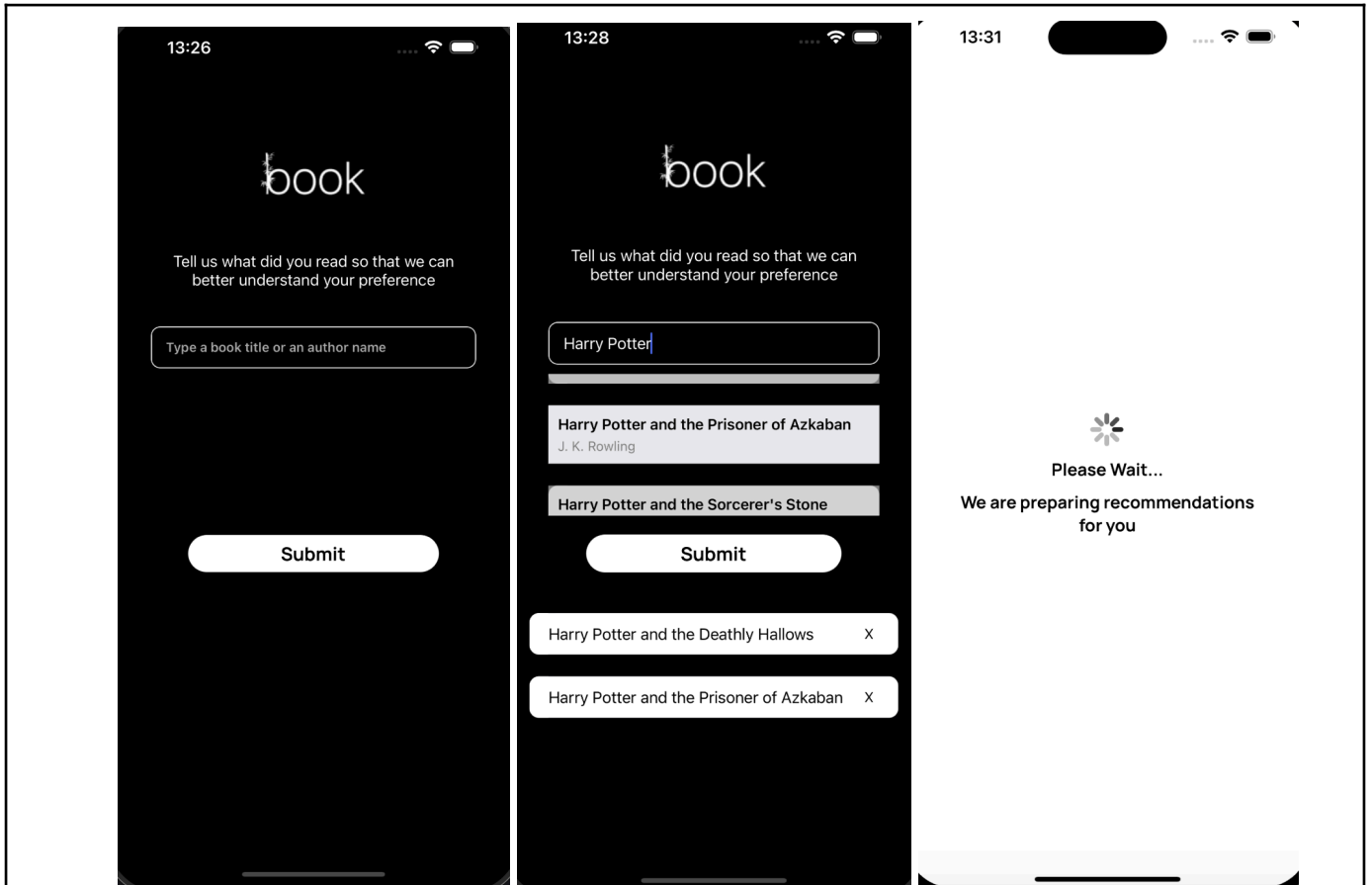


Fig #2. Preferences screen.

Then our model passes generated recommended books to the backend server. Our application gets notified and loads recommended books to the user's feed view. Users can use the search bar for searching a book he would like to read. Also, the feed view contains a section where users can find popular books among all users.

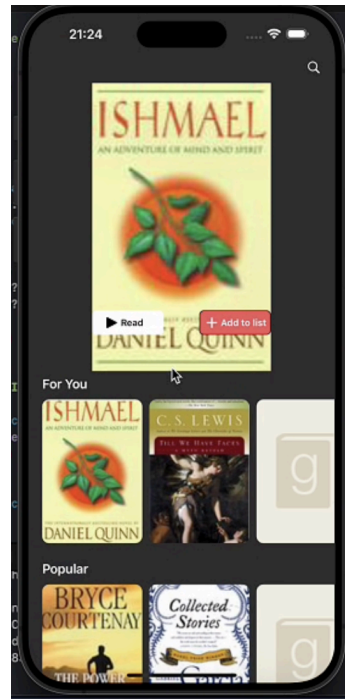


Fig #3. Feed screen.

When a user taps on a book, it navigates to the book information screen where the user can save this book to read later, share with friends or read the information about this book on amazon. Moreover, you can find the following information about the book: ratings count, average rating (in stars), name, page count, description of the book, how much does it match for you.

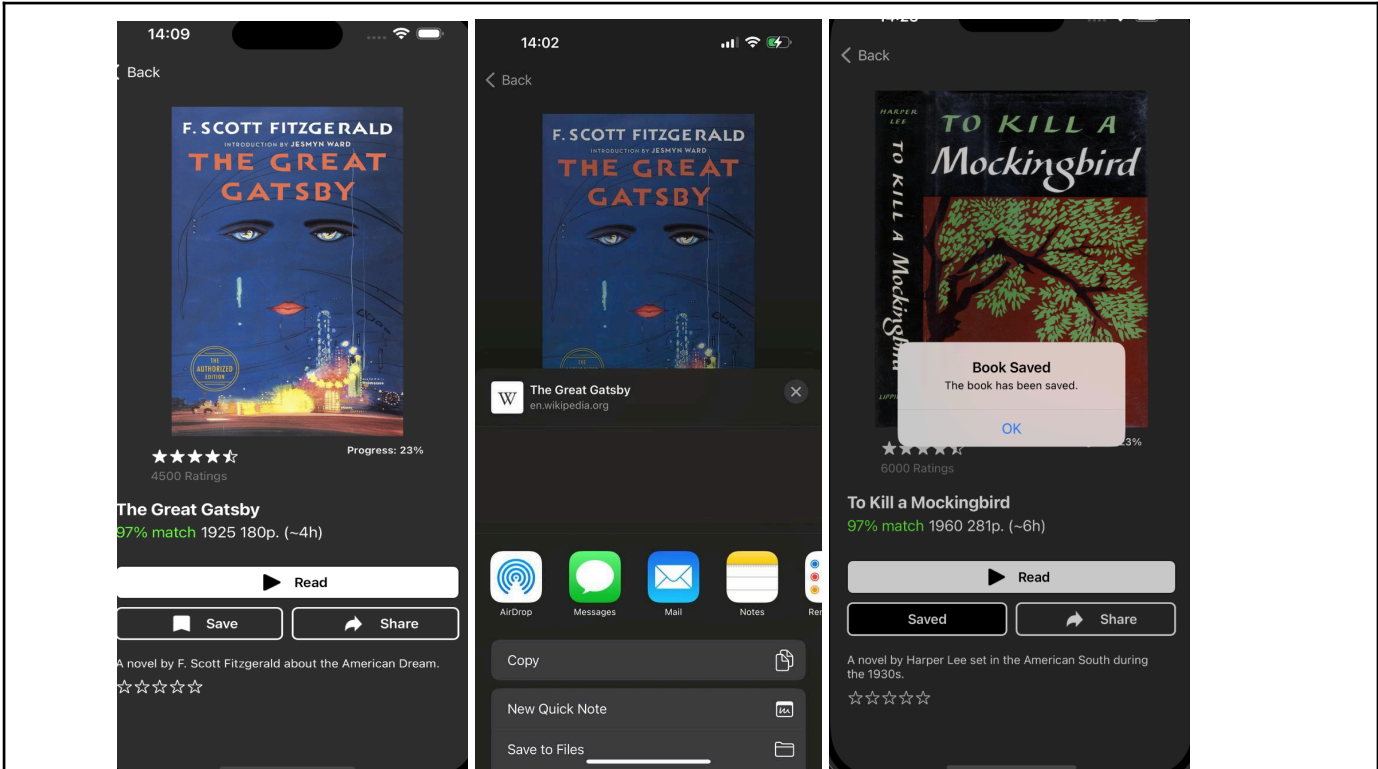


Fig #4. Book screen.

Project Execution (15%)

Design decisions:

When developing design basics, specifically colours, we have decided to use basic black and white as our brand colours. This decision was based on 2 main reasons: that lets us be agile in designing other pages and scaling our branding, but mostly, because we wanted to recreate a real book design, which is usually black and white.

Encountered issues:

During the execution of the described architecture and workflow a large amount of time was spent on the implementation of the recommendation model. Upon initiation we found that there is limited information on building an operational model that can be used for inference. Most of the code samples we came across were of poor quality and only demonstrated training and evaluation on well-cleaned datasets, with no meaningful inference capability. A suitable inference mechanism was figured out with Neural Collaborative Filtering, however it turned out that it is incapable of recommending for new users that the model was not trained on, which is the “cold-start” problem. At that point we switched to the YouTube Retrieval algorithm that is able to generate dynamic user embeddings when a user's short reading history is provided. On top of that, we found that training a recommendation model was extremely memory-intensive so we were only able to train on ~7 million interactions out of ~200 million available for 50 epochs using Nazarbayev University’s single DGX-2 NVIDIA V100 GPU (multi-GPU training was not possible with this model implementation). As a result, the recommendation accuracy is less than desirable.



Evaluation (20%)

In evaluating our application, we focused on its ability to address the challenges outlined in the introduction. We believe that our app will provide qualitative and reliable literature, free from commercial or personal biases. By utilising recommendation algorithms inspired by other platforms, we ensure that users can easily discover books that align with their preferences and goals. This not only makes the process of finding the right book more comfortable but also encourages more people to engage with literature, leading to a more self-aware, competent, and culturally reinforced readership.

To evaluate the user experience on its usability/ effectiveness, we used a questionnaire designed to gather detailed feedback from users. The questionnaire is structured to capture both quantitative metrics and qualitative insights, providing an understanding of how users perceive and engage with our platform.

The questionnaire begins by collecting demographic information to segment users based on factors such as age, reading preferences, and technological proficiency. This segmentation helps us tailor our analysis and identify specific user groups that may have unique needs or preferences. Next, we delve into the user's initial impressions and expectations before using the application. This includes questions about their previous experiences with book recommendation platforms, their motivations for using our app, and their initial thoughts on its features and design, as well as the relevance of book recommendations, and overall satisfaction with the user experience.

After using the application for a sufficient period, we will follow up with questions about the user's overall experience, including their satisfaction with the books recommended, whether they discovered new authors or genres, and how likely they are to continue using the app in the future.

In addition to these structured questions, we also included open-ended prompts to encourage users to share their thoughts, suggestions, and any additional insights they wish to provide. This qualitative feedback is invaluable in understanding the emotional and subjective aspects of the user experience, helping us uncover hidden pain points and opportunities for innovation.

The analysis of the questionnaire revealed valuable insights about BamBook users at the university.

- **Users:** Mainly students aged 16-25 with diverse reading habits and preferences.
- **Reading habits:** Vary from frequent to rare, with preference for bedtime and leisure reading.
- **Genres:** Fiction and Sci-Fi/Fantasy are popular, but there's a demand for variety (Mystery, Biography etc.).
- **Engagement features:** There's interest in reading challenges and personalised recommendations based on reading habits.
- **New book discovery streams before:** Users utilise various platforms including Amazon, BookMate, Litres, MyBook, and even TikTok.

Overall, the analysis highlights the need for BamBook to be shaped to a diverse range of user preferences by offering personalised recommendations and engaging features.

Conclusion and possible future work (5%)

Conclusion:



Developing BamBook, our objective was to provide a user-friendly platform within a mobile application for book readers to give them the opportunity to explore even more books within their personal taste. With a recommendation system captured in the user-friendly mobile application we provided such a product for users, thereby achieving our main goal. During the development process we faced a lot of challenges such as choosing a model, training it on a large dataset, optimization for the iOS platform, lack of experience and time for such a complex project. Nevertheless, we managed to do our best and by iterative approach to our senior project, achieved our original goal which is to create a usable mobile book recommendation application system.

Future work:

Here are the possible features that we could focus to enhance the functionality of the app:

- **Model improvement:** Continuously improve and train the model according to the latest standards to provide users with the suitable recommendations.
- **Security:** Ensuring the safety and privacy of the data stored. Conducting regular audits, enforcing strict access control policies and staying updated with the latest security trends.
- **Multi-platform availability:** Currently the application is adapted only for IOS. Developing Android and web-based versions will help to reach a broader audience and provide seamless access across different devices.
- **Optimization:** Optimise application so that user gets a smoother and enjoyable experience in using the application. For example, make sure that coverings of all books load timely.
- **Gamification:** Introducing gamification elements such as achievement badges, challenges, and rewards for active users to explore new genres, authors, and reading goals.
- **Expansion of content:** Continuously update and increase the database with the book information, including niche genres, international literature, and emerging authors, to cater to diverse reading interests.
- **Hypothesis testing and feedback:** Develop new product ideas and constantly conduct user testing and getting feedback from the audience, in order to keep our app user friendly and relevant for our audience.

References (5%)

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, page 191–198, New York, NY, USA. Association for Computing Machinery.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, WWW '17, page 173–182, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Iliia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha



Smelyanskiy. 2019. Deep learning recommendation model for personalization and recommendation systems. CoRR, abs/1906.00091.

Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behaviour chains. In Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018, pages 86–94. ACM.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 2605–2610. Association for Computational Linguistics.

Appendix