PROJECT PROPOSAL PROTOTYPE

Bachelor’s capstone project

ITPU IT PARK UNIVERSITY

DEVELOPING A MOBILE CATALOGUE FOR HOME LIBRARIES

Proposal for the capstone project

PROPOSAL DETAILS:

Author(s)	Yevheniia Chubar
Submission date	September 25, 2025
Reviewer	Pavel Bui, PhD EPAM, senior software developer
Resolution	Approve
Approval date	October 1, 2025

PROGRAM DETAILS:

Program code and title	60610600—Software engineering
Field of education	Software engineering
Program type	University major (UP)
Qualification level	Bachelor's (I)
Form of studies	Distance education
Program duration	3 years

GENERAL DESCRIPTION

Mobile application for filing and cataloguing books in the home library.

PROJECT BACKGROUND

Research rationale

If one's home library includes close to or more than 50 books, it becomes difficult to track where to find those books (especially, if the owner often lends or borrows them). While I understand that some older readers might prefer manual catalogues, I assume that most readers will be interested in electronic catalogues with a quick and easy access that can be updated at need in any location.

Competitor solutions

The closest competitor solutions I found in Google Play Store are:

Competitor	Pros	Cons
Handy Library (Bookshare Co., Ltd)	– Separate tab for lent/borrowed books – Search for book trough shelves – Import, export, and back-up features	– Application language is set automatically as the smartphone language – No nested shelves – Book profile is not user-friendly
My Library (Julien Keith)	– Import, export, and back-up features	– Interface is not user-friendly – No shelves – Application language is set automatically as the smartphone language
My library (Gribdall)	None	– Requires registration – Interface is not user-friendly – No back-ups
Bookshelf (SquidBit)	– Beautiful interface – Import, export, and back-up features – Allows setting the application language	– Interface is not user-friendly – No separate tab for lent/borrowed books – No nested shelves – No search for book trough shelves – Extensive profile for wish list items

Stakeholders and intended users

I assume that an intended user of my application can be described like this:

- has a home library or frequently lends or borrows books,
- has a smartphone and is used to using mobile applications,
- is frustrated by the clutter on their bookshelves and the fact that books get lost because they are borrowed and never returned,
- needs to get access to their home library catalogue in any location,
- prioritizes reliability, accuracy, and ease of use.

Market potential

Statistics (Anderson, 2024) says that an average person reads close to 5.48 books per year, in some countries this parameter going as high as 17 and even 24 books per year. Statistical data for USA (12 to 17 books read on average per year) proves that 80% readers prefer physical books to digital ones (including Gen-Z readers). If the USA data correlates to the data in other countries, I assume that an average reader in the world has a home library of physical books and needs a book catalogue for it that will include physical books, eBooks, and audiobooks.

At the same time, I found only four mobile applications for filing and cataloguing books in Google Play Store, and none of those applications fully cover the functionality I suggest.

PROJECT REALIZATION

Features and functionalities

1. Requires no registration.
2. Import, export, and back-up features.
3. Allows setting the application language.
4. **Books**, **Shelves**, and **Lend/Borrow** tabs + wish list.
5. **Books** support sorting and filtering (by parameters from the book profile).
6. **Shelves** support reordering, nesting, and searching for books on shelves.
7. Book profile has three tabs with different parameters (required and optional).
8. Shelves and genres are user-defined.

Technology stack

1. Client side:
   - Programming language: Kotlin
   - Framework: Jetpack Compose
   - Database: SQLite via Room Persistence Library
2. Data storage and backups:
   - Local database: SQLite
   - Cloud backup: Firebase Realtime Database
3. Development tools:
   - IDE: Android Studio
   - Version control: GitHub
   - Testing: JUnit for unit testing, Espresso for UI testing
4. Design & user experience
   - UI guidelines: Material Design 3