Quickbook

submitted in partial fulfillment of the requirement
for the award of the Degree of

Bachelor of Technology
in
Computer Engineering

by

Malhar Bangdiwala
Sakshi Mahadik
Yashvi Mehta

under the guidance of

Prof. Abhijeet Salunke
Dr. Rita Das

Department of Computer Engineering
Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri-West, Mumbai-400058
University of Mumbai
December 2022

Certificate

This is to certify that the Project entitled “Quickbook” has been completed to our satisfaction by Mr. Malhar Bangdiwala, Ms. Sakshi Mahadik and Ms. Yashvi Mehta under the guidance of Prof. Abhijeet Salunke and Dr. Rita Das for the award of Degree of Bachelor of Technology in Computer Engineering from University of Mumbai.

Certified by

Prof. Abhijeet Salunke **Project Guide**	**Dr. Rita Das** **Co-Guide**	**Dr. Dhananjay Kalbande** **Head of Department**

Dr. Bhalchandra Chaudhari
Principal

Department of Computer Engineering
Bharatiya Vidya Bhavan’s
Sardar Patel Institute of Technology
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri(W), Mumbai-400058
University of Mumbai
December 2022

Project Approval Certificate

This is to certify that the Project entitled “Quickbook” by Mr. Malhar Bangdiwala, Ms. Sakshi Mahadik and Ms. Yashvi Mehta is found to be satisfactory and is approved for the award of Degree of Bachelor of Technology in Computer Engineering from University of Mumbai.

External Examiner	**Internal Examiner**
(signature)	**(signature)**
Name:	**Name:**
Date:	**Date:**

Seal of the Institute

Statement by the Candidates

We wish to state that the work embodied in this thesis titled “Quickbook” forms our own contribution to the work carried out under the guidance of Prof. Abhijeet Salunke and Dr. Rita Das at the Sardar Patel Institute of Technology. We declare that this written submission represents our ideas in our own words and where others’ ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission.

Name and Signature:
1. Malhar Bangdiwala - 2019130005
2. Sakshi Mahadik - 2019130034
3. Yashvi Mehta - 2019130037

Acknowledgments

It is really a pleasure to acknowledge the help and support that has gone into making this thesis. I express my sincere gratitude to the guide Prof. Abhijeet Salunke and co-guide Dr. Rita Das for their invaluable guidance.

I thank my examiners for their words of advice. I thank the HOD and staff of the Computer Engineering Department for giving me all the facilities required to carry out this research work.

I would like to thank all my family members and well-wishers for their constant encouragement for all these years, without which I could not have completed this work.

Contents

List of Figures

List of Abbreviations

API	Application Programming Interface
CSV	Comma-Separated Values
LFW	Labelled Faces in the Wild
IDE	Integrated Development Environment
ISBN	International Standard Book Number
IST	Indian Standard Time
OCR	Optical Character Recognition
UI	User Interface

Abstract

Traditional methods of managing a library manually have their disadvantages like long waiting queues and expensive scanner machines. The Quickbook application aims to overcome these drawbacks by making use of face and text recognition to issue books from the library. The main idea is that a picture of the book with the issuer will be taken from which the required data will be extracted. The account of the user will be created by the library admin by uploading a picture of the face of the user. The application consists of two types of logins - user and admin, where administrators can keep track of user transactions and history. The face recognition algorithm and book recognition algorithm gave an overall accuracy of 99.43% and 82.5% respectively. The user can also see the information related to books and this information is extracted from a third-party API. Further, easy payment and reissue options for the books on the application have also been implemented.

Chapter 1

Introduction

The role of the library can be considered a vital and important component of an educational institution, which is a main means of teaching, learning activities, etc. Students, researchers, professors, and others can explore huge resources of information obtained from libraries. The largely increasing number of library books and increasing need for books and information is difficult to manage with traditional manually operating libraries. The slow-moving process of scanning causes long queues in libraries. Automated library systems can increase efficiency and hence, universities should adopt such systems for reducing manual labor.

In our project, we created an application called Quickbook, an automated library system. The user has to take a picture holding a book and the system recognizes the face of the user and the title of the book simultaneously using face recognition and book recognition algorithms respectively. Further, easy re-issue of books and payment of overdue books is also implemented in the application. The application also displays information related to the book such as the quantity, author, genre, etc. The admin can set the rules of the library and the number of copies of a particular book. The users are notified of the return date and any overdue they have, and they can pay through the application itself which saves their time and cost of travel.

1.1 Motivation

1.2 Objectives

The objectives of our project are:

- To make the process of issuing library books easier and more efficient
- To reduce scanning costs for the library
- To reduce human error while issuing books
- To make libraries more accessible in less developed areas

1.3 Problem Statement

A major deterrent in libraries while issuing books is the long waiting lines and the slow process of scanning. Students and the public have to wait in long queues to issue books. Further, for the member, creating and possessing a library ID for issuing and returning books can be a tedious process. Adding personal details as well as carrying the library card can sometimes be inconvenient. Additionally, the hardware cost of scanners in libraries makes setting up a library more difficult in economically backward areas. Human errors while issuing books may lead to monetary losses for members. The availability of a book is not transparent for members, that is, members may not know which books are available in the library. Our solution aims to overcome all these problems by proposing an automated library system that takes the picture of the borrower as well as the book instead of scanning.

1.4 Contributions

The contribution to the project to solve the problem includes

- Face Recognition - recognizing the face of the book borrower
- Text Recognition - recognizing the ISBN of the book in order to identify the book issued
- Book search and information retrieval- feature for users to search for book information
- Keeping available stock of each book - The number of copies of every book would be known to the admin
- Payment option for overdue books - Users can pay online for the books they have returned late
- Profile for user and admin - Profile for both users and admins
- Notifications for book returns - Users will receive an email notification when the return date of a book is near.

1.4.1 Constraints and Assumptions

The Constraints and Assumptions of the project include

- Book and face are clearly visible in the picture - In order to recognize the face and the book ISBN, the picture should be clear; blurred images would affect the accuracy of the recognition
- Stable internet connection - There should be a stable internet connection while clicking the picture to recognize face and book
- Information of book will be reliant on third-party API - Since book data is not being stored and a book API is used to retrieve book information, there is a dependence on the book API
- Member data and book stock data must be provided initially by the library - The library must provide all the initial information about the books available

1.4.2 Future Work

The Future Work of the project includes

- Book recommendations to users based on past history
- Detect multiple faces and books from a single image

1.5 Layout of the Report

A brief chapter by chapter overview is presented here.
Chapter 2: A literature review of different real-time solutions for library management systems, book recognition, and face recognition is presented.
Chapter 3: The design of the proposed system is depicted here using ER diagram, a Use case diagram, a Flowchart, and a Block Diagram. Further, a detailed explanation of the algorithms used in the project will be described in this chapter.
Chapter 4: In this chapter, the implementation details are presented.
Chapter 5: The results obtained from Face recognition and Book recognition are discussed in this chapter.
Chapter 6: Conclusions and discussion on future course of research work.

Chapter 2

Literature Survey

Authors of the paper titled Face Recognition System [1] propose an automated face recognition system where they neural networks, KLT Algorithm, Viola-Jones Algorithm, Haar classifier, and PCA algorithm. Viola-Jones Algorithm and Haar classifier have been used for face detection. Neural networks further identify the detected image. The proposed method achieves better accuracy and the usage of PCA outperforms the existing work. Raktim Nath, Kaberi Kakoty, and Dibya Jyoti Bora propose face recognition using an HOG detector in their paper titled Face Detection and Recognition using Machine Learning [2]. They put forward that HOG results in more accuracy than Haar-Cascade since Haar Cascade gives more false positives. Their model was able to detect 22 faces in one frame with 1 false positive and could not detect 3 people. Even though the algorithm gives more accuracy and productiveness, it is however a very time-consuming algorithm. In the paper titled Image based book cover recognition and retrieval [3], authors - Kalyani Sukhadan, V Vijayarajan, A Krishnamoorthi, and D Geraldine Bessie Amali propose a graphical user interface powered by MATLAB that allows users to instantly check book-related data. They used the MSER algorithm to detect useful regions (text regions) in the image and compared MATLAB OCR with Tesseract OCR for accuracy. In terms of accuracy, Tesseract OCR was 86%, and MATLAB OCR was 78%. In terms of time cost, MATLAB OCR is faster than Tesseract OCR. KH Teoh, RC Ismail1, SZM Naziri, R Hussin, MNM Isa, and MSSM Basir have implemented a face recognition method using OpenCV and Haar Classifier in their paper - Face Recognition and Identification using Deep Learning Approach [4]. They describe the idea of designing and creating a facial recognition system utilizing deep learning and Python's OpenCV library. Deep learning ensures high accuracy and their final accuracy is 91.7%. They conclude that the accuracy of face recognition on images is higher than the accuracy of real-time video because of the high resolution of images. Maheshwari S et al. [5] have proposed a system wherein the library is automated by adding additional features such as separate logins for users and librarians. This would enable the generation of unique reports as well reservation of books. The system also incorporates notification features when the return date is near. The system also eliminates the need for library cards for each book. However, manual work is still involved in issuing the book (scanning the book and user id card. K.Puritat and K. Intawong [6] have proposed a library system that is more focused on recommendations. For the same, they have used the Chiangmai school library database. In their system, individual logins enable the user to get custom recommendations based on the history of the books that they have issued. To make these recommendations, they have made use of Support Vector Machines. Three sources of data have been combined for the same: Title similarity based on the Damerau–Levenshtein distance, The Dewey Decimal Classification for the book classification system, and Bibliographic Information of book similarity based on multiple information of the bibliography. S. Korade and S. Patil [7] have leveraged RFID technology for automating library processes. This involves reading the barcode of the book and the member id. The software enables the users to log in and issue the book. Fines can also be paid using the app. However, there is no effort to reduce manual interference via the app. O. Olatunji et. al. [8] have studied the issues present in the current automated library system that is being used at Kano University of Science and Technology. In this university, NewgenLib software is being used. While many of the advantages have been highlighted, some of the disadvantages that have come to light are the need for training of librarians to use the system as well as the constant need for a power supply that is required to scan and enter the data into the software. Manikandan J et. al [9], proposed a face recognition system for real-time application using CNN. This is followed by the analysis and evaluation of the system by using alternate CNN attributes or parameters to increase the accuracy and performance of the system. Initially, they used the AT&T dataset which consisted of 400 images and this model is compared with previous work. After successfully evaluating the AT&T dataset system, real-time inputs through the camera are used to evaluate the proposed system. For the AT&T dataset, the maximum precision obtained is 98.75% and for real-time inputs, the highest accuracy obtained is 98.00%. Atiqul Islam Chowdhury, Mushfika Sharmin Rahman, and Nazmus Sakib [10], implemented a system that can simultaneously detect multiple product barcodes using the Zbar Barcode Reader library. Image processing is used to identify these barcodes. 1D and 2D barcodes are detected in a single image and the system is angle invariant and provides minimum interaction by the users. The paper had the limitation of not correctly detecting barcodes because of light reflection, image contrast, and other external factors. Sahana K Adyanthaya [11], discusses various steps involved in text recognition. The application of text recognition and why it's needed is also discussed in the paper. The paper talks about preprocessing of images where images are converted to a suitable format and any kind of noise are removed and the images are processed into binary. Then, to separate each character, the segmentation process is applied using different techniques. Lastly, the paper discusses the classification process to identify the text. Ratko Grbi´c et. al [12], proposed a way to increase the efficiency of Tesseract Optical Character Recognition (OCR) on colorful images. The method consisted of creating clusters of pixels using the KNN algorithm. Then the texts in the selected cluster are classified using the Naive Bayes classifier. For text recognition, OCR is performed on the selected cluster. The results showed an improved performance in OCR by around 20%. The final accuracy achieved was 95.5%.

Chapter 3

Design

3.1 Entity-Relationship Diagram

Fig. 3.1 depicts the Entity-Relationship Diagram for our application. The entities are Admin, Users, Books, and Transaction (a weak entity). The relationships between entities are as follows:

- An admin can keep track of N users. A user is managed by N admins.
- An admin can manage N number of books.
- An admin has access to N number of transactions.
- A user can make N number of transactions.

3.2 Use Case Diagram

The use case diagram shown above shows the users (actors) involved and their interaction with the system. It shows an overview of the relationship between the use cases of the project, actors, and systems involved in our project.

3.3 Flowchart

As depicted in Figure 3.3, the user will first log into the app. Upon successful login, the user will be classified into admin and member. The member will have the option to view details of a particular book, check the currently issued books, and history of issued books, update his profile or pay the fee for a returned book. If the date of return is near, the user will also get an email notification. Similarly, the admin will have the options to view details of a particular book, edit the number of copies a book has, edit or enter new information about a user, take a picture to issue a book to a user, get the overview of the books issued in the library, verify a book return and its corresponding fee(if any) as well as set the library rules.

Figure 3.1: Entity-Relationship Diagram

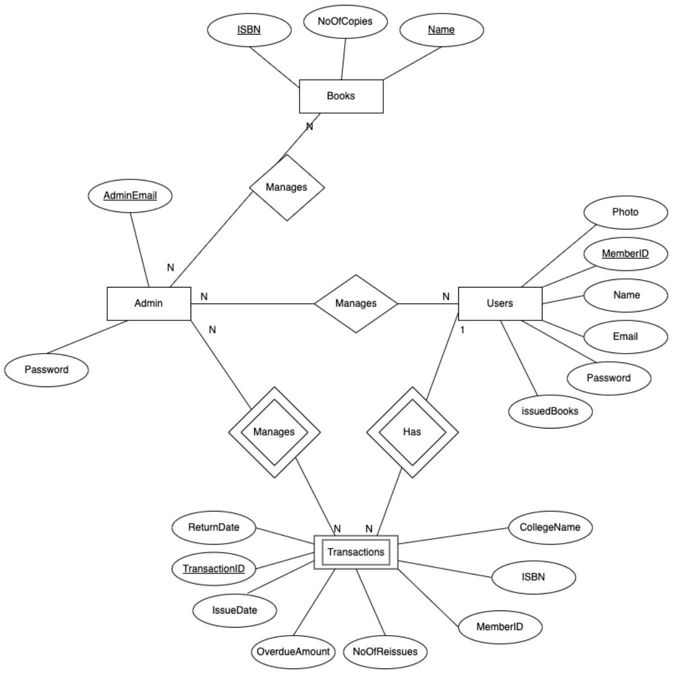


Figure 3.2: Use Case Diagram

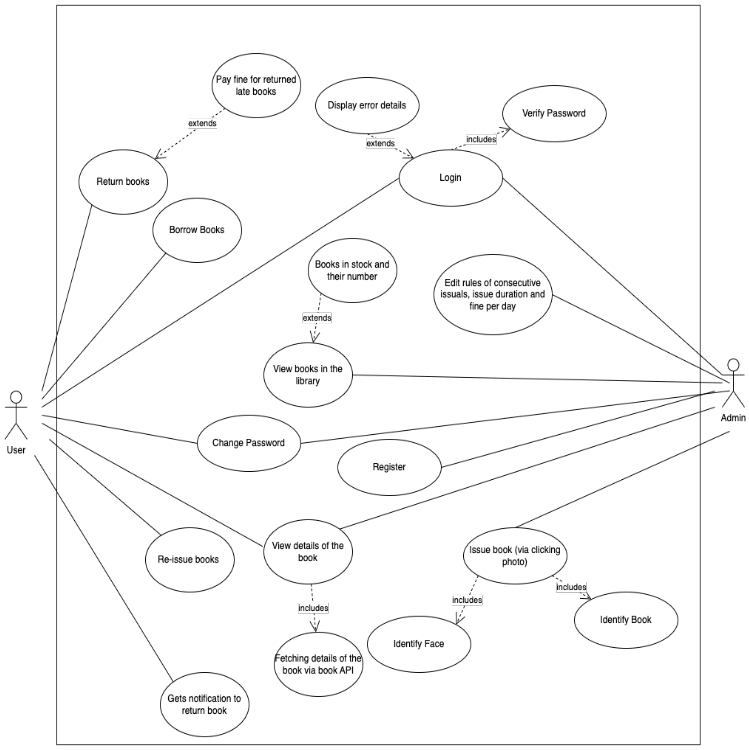
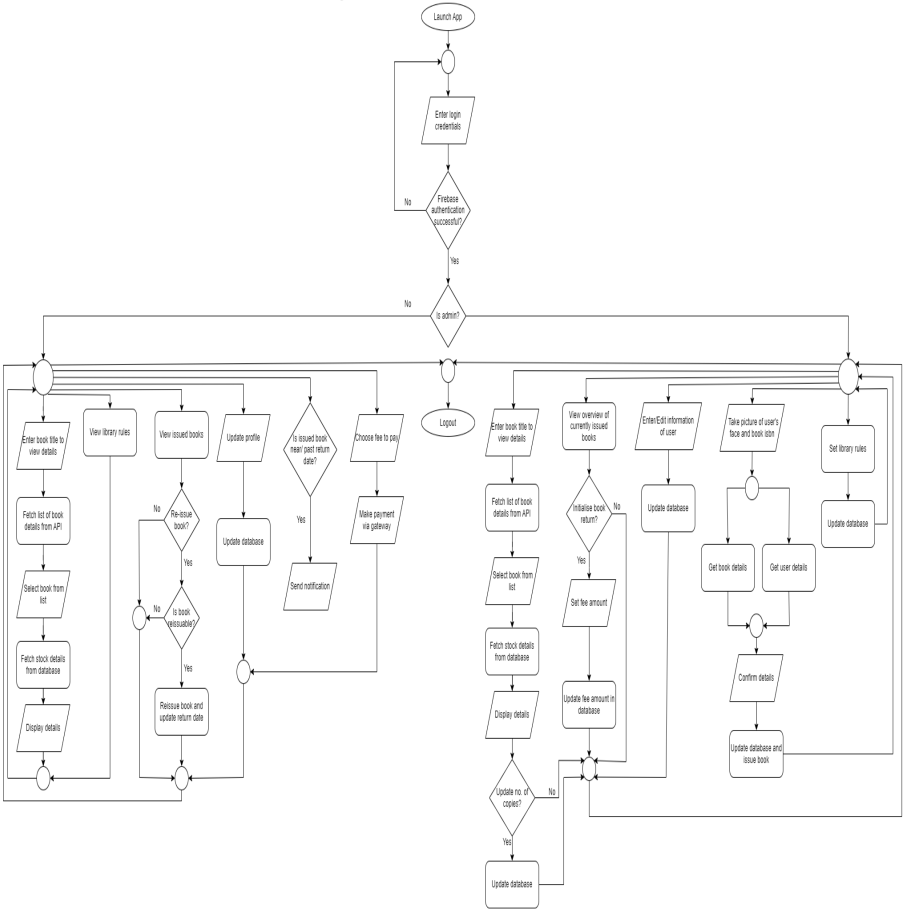

Pay fine for returned late books
extends
Display error details
Verify Password
extends
includes
Return books
Login
Borrow Books
Books in stock and their number
Edit rules of consecutive issuals, issue duration and fine per day
extends
View books in the library
Admin
User
Change Password
Register
View details of the book
Issue book (via clicking photo)
Re-issue books
includes
includes
includes
Identify Book
Identify Face
Fetching details of the book via book API
Gets notification to return book

Figure 3.3: Flowchart


Launch App
Enter login credentials
Firebase authentication successful?
No
Yes
Is admin?
No
Enter book title to view details
Fetch list of book details from API
Select book from list
Fetch stock details from database
Display details
View library rules
View issued books
Re-issue book?
No
Yes
Is book reissuable?
No
Yes
Reissue book and update return date
Update profile
Update database
Is issued book near/ past return date?
Yes
Send notification
Choose fee to pay
Make payment via gateway
Logout
Enter book title to view details
Fetch list of book details from API
Select book from list
Fetch stock details from database
Display details
Update no. of copies?
No
Yes
Update database
View overview of currently issued books
Initialise book return?
No
Yes
Set fee amount
Update fee amount in database
Enter/Edit information of user
Update database
Take picture of user's face and book isbn
Get book details
Get user details
Confirm details
Update database and issue book
Set library rules
Update database

3.4 Proposed System

Fig. 3.4 shows the proposed block diagram of the application. This diagram highlights all the major modules of the application. Firstly, the login module, enables both the users and the admin to log into the app. The view book details module enables the users and admins to view all the information of a specific book. The issue book module comprises two machine learning modules: the face recognition module and the book recognition module. Only the admin is allowed to issue a book by clicking the image of the user along with the book. The view history module enables the user to check all the past records of the books issued by him/her. The re-issue module allows the user to reissue a book if the required criteria are met. The rules module consists of setting the rules (Issue duration, Number of consecutive issues, Penalty for late returns) by the admins and viewing the same by the members. The update book information module lets the admin update the number of copies of a specific book that the library has. The return book module also contains the payments module (the user has to pay a fine in case the return of a book is late) and the notifications module (notifications are sent to the user when the return date is near). Only admin can access the register user module. Here, the admin has to add all the details of the registered user along with one photo. The UI module is from a development standpoint, where the entire UI of the app will be generalized and refined to give the user a rich aesthetic and user experience.

3.4.1 Algorithm Details

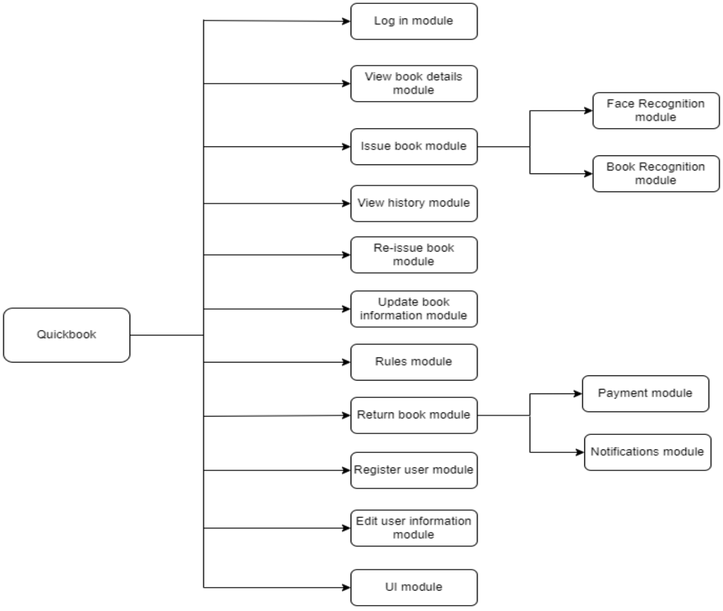
For the Book Recognition module, the Tesseract OCR engine is being leveraged. Since the code is being written in Python, Pytesseract[13] is being used. The image is first stored in .tiff format. Then, the image is converted to greyscale and is zoomed in for better results. Haar classifier[14] is used to detect a book in the image. Further, the image is then rotated by 90 degrees until the image text can be converted into a string. Next, regex is used to extract the ISBN number from the obtained string. To obtain book information, this ISBN is passed to the Google Books API.

For the Face Recognition module, the dlib[15] library is being used. Since the code is being written in Python, its python wrapper module: face_recognition[16] has been used. First, an image of the user is first broken down into 128 data points. This is the reference point. This data, along with the user name is stored in separate CSV files. When the image has to be recognized, these two CSV files are first read. The image is then scanned for faces using the hog model. The new image is then broken down into similar 128 data points and then compare to the existing faces. The difference is calculated and the closest face is given as the output, provided the distance is under a pre-defined limit.

These two modules are bundled in a Flask app that has been hosted on an Azure VM. The APIs for the two modules are called from the Android app using Retrofit[17]. The login module has been implemented using Java. Firebase's authentication module[18] has been used similarly. The payments module has been implemented using the Razorpay API[19]. The Glide[20] framework is used for image caching(book images) for a smoother user experience while scrolling. The notification module has been implemented as a python script on the Azure Virtual

Machine. Cron job scheduler has been used to run the script on a daily basis. The rest of the modules have been implemented using Java by building their respective logic using the Android Studio IDE.

Figure 3.4: Block Diagram



Chapter 4

Implementation

4.1 Details of work done

The entire application, as defined by the scope has been completed. A Flask app to expose the APIs for the two modules: book recognition and face recognition, as well as for searching a book has also been created. There are three APIs in total. One API is to register a new user's face on the platform. The face is stored as numerical data of 128 data points in a CSV file. The second API is to identify the face from an image, provided that the face has already been registered previously. This API also identifies the book's ISBN number from the image. Details regarding the face and book are then returned. The third API is to return all the data related to a book with the help of Google Books API [21]. The integration with the external API has also been completed. This Flask app in turn has been hosted on an Azure Virtual Machine on a public IP Address. Thus, the APIs are available for everyone to use.

Further, the entirety of the Android App has been completed. First, the user would encounter a splash screen and the login page. Upon logging, two flows are possible based on the type of user: admin and member.

If the user is an admin, the following will first be visible: a page where a book can be issued to a member, by clicking the picture. Once the image has been taken, the results are displayed on the screen. These results are editable to account for prediction errors if any. The book can then be issued by clicking on the "Issue Book" button. There is also a page for registering a new member. The admin can click an image of the user and enter the corresponding member ID and email ID. This information will be stored in the database. The face will be stored as numerical data as described above. Then, there is the settings page where the admin can edit certain rules of the library: Duration of a book issued, Number of consecutive issues allowed, and the fine imposed per day in case of a late return by a member. On this page, the admin can also change the account password. From here, the admin can log out from the app as well. A "Search books" page has been created where the admin can search for any book by its name. Google Books API is leveraged for the same. The results that are given by the Google Books API are then cleaned and relevant information is passed to the app. Here, the admin will now be able to see a list of all books matching the search title. On clicking on any book, the admin will be able to see basic details such as title, authors, publication information, description, and the cover page of the book. An editable field for storing the number of copies available in the library has also been provisioned. Finally, there is an ”overview” page, where the admin can check all the currently issued books. The admin can wish to accept the return of any book by a member from here. A search bar to search for currently issued books by a certain member has been enabled. If the admin, accepts a return, a fee is calculated for a late return. This is an editable field (to account for damages to the book etc). Once submitted, the fees will be reflected on the member's side.

If the user is a member, an overview of all the books issued by him will be visible. If applicable, an option to reissue the book will also be present. Similarly, the option to pay the fee for a specific book might also be present. The payment system has been implemented by Razorpay. A reminder email, 2 days prior to the return date will be sent to the member. This has been enabled by using the Cron scheduler on the virtual machine. These reminders would be sent at 12:30 PM IST. A settings page has been created where the member can change the current password. Here, the rules, as mentioned above are also visible. However, these rules are not editable. A search page similar to the admin's section has also been implemented. The only difference is that the member can only view the number of copies available in the library, and not edit it.

4.2 Screenshots

Fig.4.1 is the splash screen. This screen is displayed momentarily when the app is first opened. Fig.4.2 is the login page. The user will enter the email id and password to successfully log into the app. Fig.4.3 is the upload page. Here, the librarian has the option to either click a photo or upload a photo from the gallery. The face recognition and book recognition algorithms will be run on this image. The output of which will be displayed as per Fig.4.4. Fig.4.5 shows the page where the admin can add a new member into the system. This page is visible after the admin clicks the image of the new user. Fig.4.6 shows the settings page for the admin. Here, the admin can change the rules such as issual duration and fine amount. At the bottom half, the admin can change their password. A logout button is also present. Fig.4.7 shows the overview of all the currently issued books to the admin. The admin can choose to accept the return of any of these books as well. Fig.4.8 shows the search book page. This page is common to both the user and the admin. Here, upon searching the book name, the result from the Google Books API is displayed. Fig.4.9 is the detail of the book that the admin searched for. Along with the details, the admin can edit the number of copies that are currently present in the library. Fig.4.10 is the same page, but as seen by a user. Here, the only difference is that the number of copies cannot be edited. Fig.4.11 shows the page where the user can see his history of issued books. If the book can be re-issued, the button for reissual will be displayed. Upon clicking it, the return date will be modified. Similarly, if a pending fine is there, the pay fine button will be present. Upon clicking that, Fig.4.12 will be visible. The razorpay UI is present in which the payment can be done. Fig.4.13 is the user settings page. It is similar to the admin settings page, except that the rules are not editable. Fig.4.14 shows the email notification, as visible in the Gmail app.

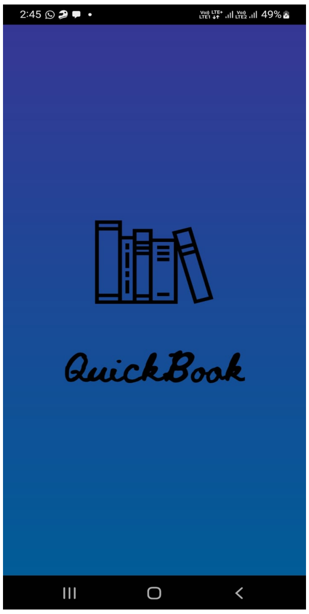
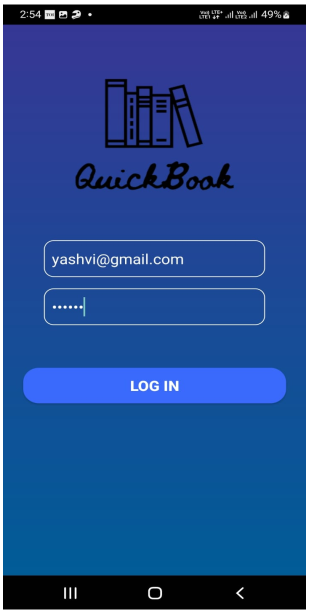


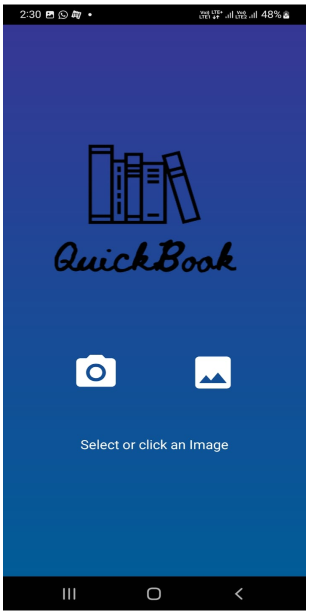
Figure 4.1: Splash Screen Page



yashvi@gmail.com

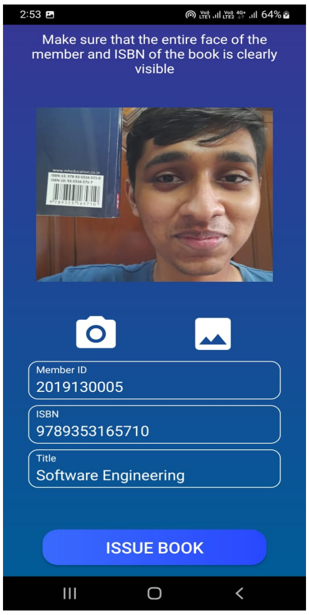
••••••

Figure 4.2: Log in Page



Select or click an Image

Figure 4.3: Upload Page-1



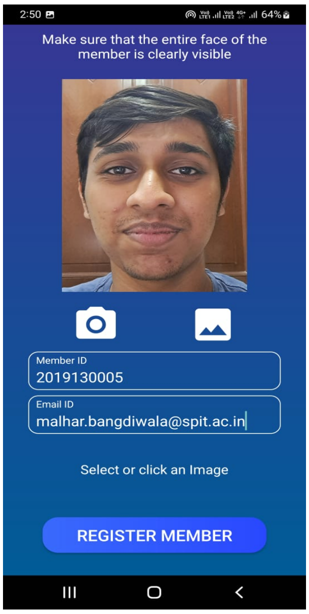
Make sure that the entire face of the member and ISBN of the book is clearly visible

Member ID
2019130005

ISBN
9789353165710

Title
Software Engineering

Figure 4.4: Upload Page-2



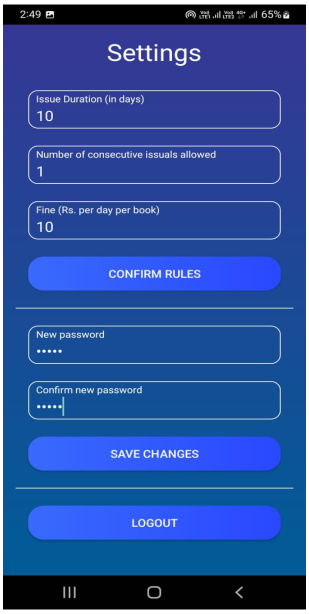
Make sure that the entire face of the member is clearly visible

Member ID: 2019130005

Email ID: malhar.bangdiwala@spit.ac.in

Select or click an Image

Figure 4.5: Add User Page



Settings

Issue Duration (in days): 10

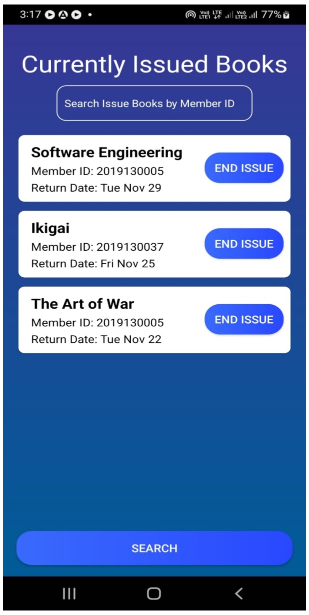
Number of consecutive issuals allowed: 1

Fine (Rs. per day per book): 10

New password: •••••

Confirm new password: •••••

Figure 4.6: Admin Settings Page



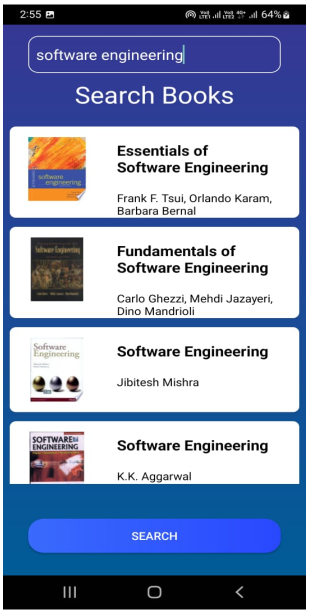
Currently Issued Books

Software Engineering
Member ID: 2019130005
Return Date: Tue Nov 29

Ikigai
Member ID: 2019130037
Return Date: Fri Nov 25

The Art of War
Member ID: 2019130005
Return Date: Tue Nov 22

Figure 4.7: Admin Issued Books Overview Page



software engineering

Search Books

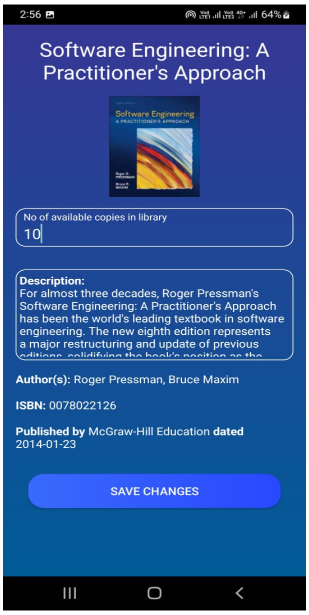
Essentials of Software Engineering
Frank F. Tsui, Orlando Karam, Barbara Bernal

Fundamentals of Software Engineering
Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli

Software Engineering
Jibitesh Mishra

Software Engineering
K.K. Aggarwal

Figure 4.8: Search Book Page



Software Engineering: A Practitioner's Approach

No of available copies in library
10

Description:
For almost three decades, Roger Pressman's Software Engineering: A Practitioner's Approach has been the world's leading textbook in software engineering. The new eighth edition represents a major restructuring and update of previous editions, solidifying the book's position as the

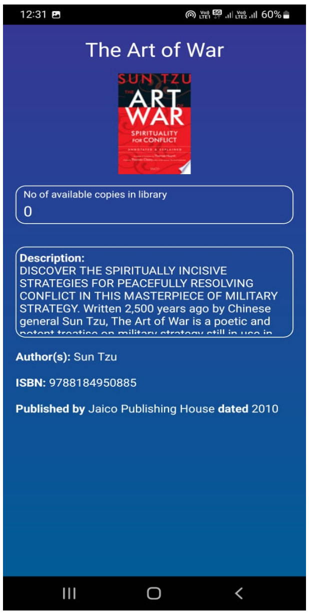
Author(s): Roger Pressman, Bruce Maxim

ISBN: 0078022126

Published by McGraw-Hill Education **dated** 2014-01-23

SAVE CHANGES

Figure 4.9: Admin view for Book Information Page



The Art of War

No of available copies in library
0

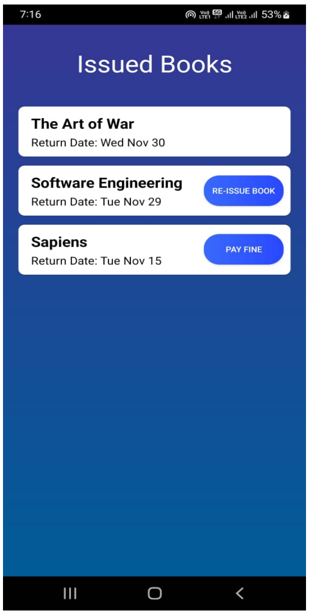
Description:
DISCOVER THE SPIRITUALLY INCISIVE STRATEGIES FOR PEACEFULLY RESOLVING CONFLICT IN THIS MASTERPIECE OF MILITARY STRATEGY. Written 2,500 years ago by Chinese general Sun Tzu, The Art of War is a poetic and potent treatise on military strategy still in use in

Author(s): Sun Tzu

ISBN: 9788184950885

Published by Jaico Publishing House **dated** 2010

Figure 4.10: User view for Book Information Page



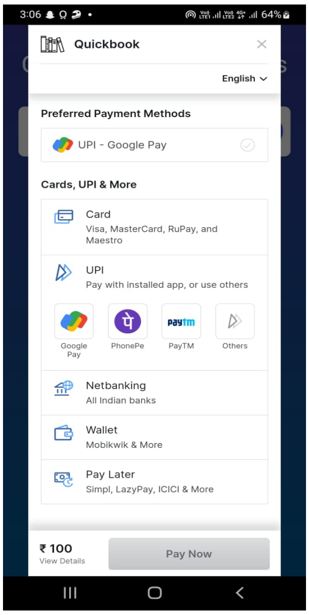
Issued Books

The Art of War
Return Date: Wed Nov 30

Software Engineering
Return Date: Tue Nov 29
RE-ISSUE BOOK

Sapiens
Return Date: Tue Nov 15
PAY FINE

Figure 4.11: User view for Issued Books Page



Quickbook

English

Preferred Payment Methods

UPI - Google Pay

Cards, UPI & More

Card
Visa, MasterCard, RuPay, and Maestro

UPI
Pay with installed app, or use others

Google Pay
PhonePe
PayTM
Others

Netbanking
All Indian banks

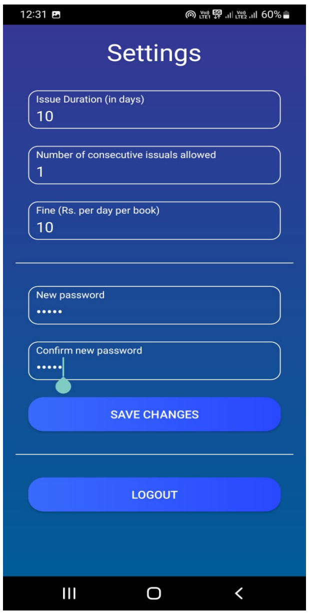
Wallet
Mobikwik & More

Pay Later
Simpl, LazyPay, ICICI & More

₹ 100
View Details

Pay Now

Figure 4.12: Payment via Razorpay Page



Settings

Issue Duration (in days): 10

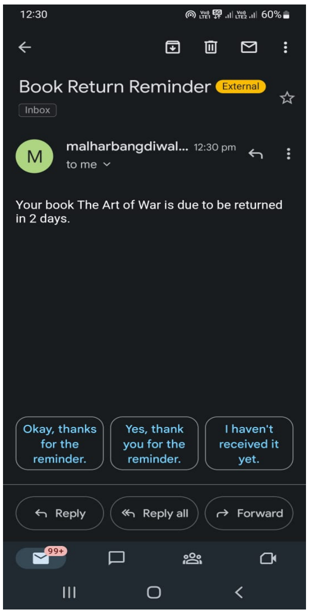
Number of consecutive issuals allowed: 1

Fine (Rs. per day per book): 10

New password: •••••

Confirm new password: •••••

Figure 4.13: User settings Page



Book Return Reminder

malharbangdiwal... 12:30 pm
to me

Your book The Art of War is due to be returned in 2 days.

12:30
60%
External
Inbox
M
Okay, thanks for the reminder.
Yes, thank you for the reminder.
I haven't received it yet.
Reply
Reply all
Forward
99+

Figure 4.14: Email Reminder Notification on Gmail

Chapter 5

Results and Discussion

5.1 Face Recognition

The face recognition algorithm has been tested on the Labelled Faces in the Wild (LFW)[22] dataset. After cleaning, this dataset contained images of over 800 famous personalities from different angles and expressions. 1 photo of each celebrity was used for training purposes. After training, the remaining photos were used as the testing dataset. More than 2100 images were utilised for the purpose. The algorithm was able to detect a face in all the images. An overall accuracy of 99.43% was obtained. This has been captured in Fig.5.1 below.

5.2 Book Recognition

The book recognition algorithm has been tested on a manually curated dataset of about 80 book images that contained the ISBNs in it. These images were clicked in different lighting and different angles. In this, 66 ISBNs were correctly identified, 8 were incorrectly identified and 6 were not identified at all. This meant an overall accuracy of 82.5%. This has been depicted in Fig.5.2 below.

Figure 5.1: Accuracy graph for Face Recognition Algorithm

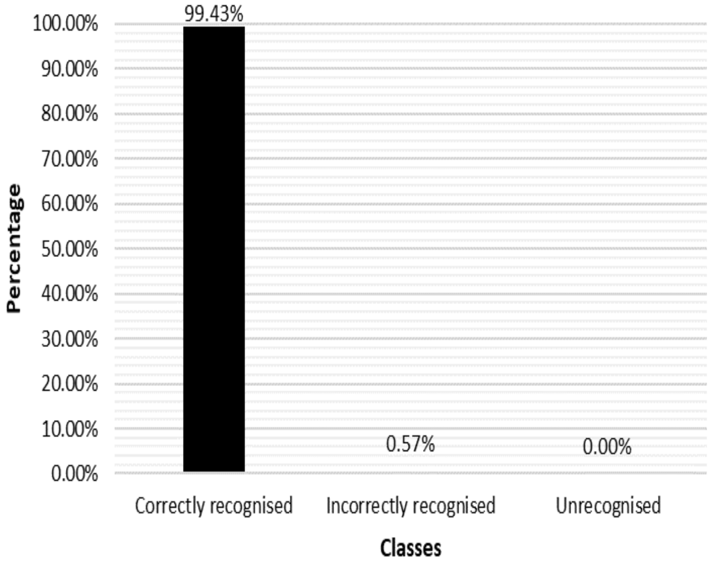
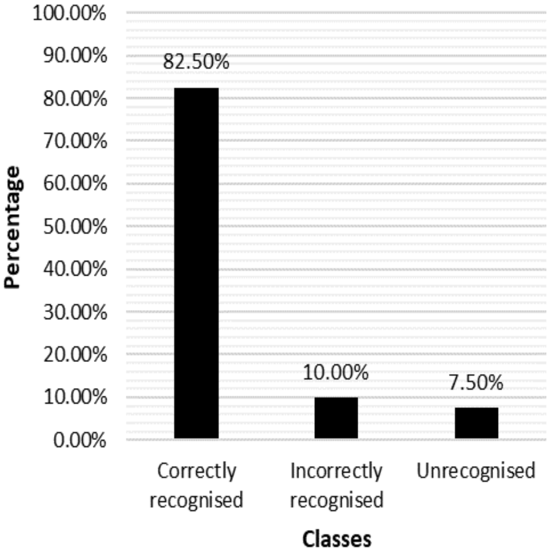


Figure 5.2: Accuracy graph for Book Recognition Algorithm



Chapter 6

Conclusion

The issues with the traditional library and current library method are addressed with our proposed system. Using our app, the librarian can issue a book to the registered user by clicking a single picture which saves time and prevents long queues. The main objective of creating an automated library project, that is , to utilise face and text recognition, was achieved. Basic features such as issue, return, re-issue, fine were added in the application as well. Further, additional functionalities such as search the book present in the library along with its number of copies were implemented which is additional feature to the basic library characteristic.

Chapter 7

Future Scope

Although the app successfully recognizes faces and books and implements all the functionalities of a typical library, the functionality can be extended to include features like the recommendation of books to users. The app should be able to recommend books to users based on their search history and their list of issued books. Further, the app should also suggest books to users based on their semester requirements. Moreover, currently, the app only recognizes a single book and face in one picture. The future scope also includes recognizing multiple faces and books from a single picture so the process becomes faster and more convenient.

Chapter 8

Research Publication

The paper corresponding to this project, titled ‘Automated Library System Management’ has been accepted and will be presented at the 4th International Conference of Emerging Technologies (INCET). The paper will be published in IEEE thereafter.

Bibliography

[1] Shivam Singh, Prof. S. Graceline Jasmine, 2019, Face Recognition System, International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 Vol. 8 Issue 05, May-2019

[2] Nath, Raktim & Kakoty, Kaberi & Bora, Dibya & Welipitiya, Udari. (2021). Face Detection and Recognition Using Machine Learning. 43. 194-197.

[3] Sukhadan, K., Vijayarajan, V., Krishnamoorthi, A., & Geraldine Bessie Amali, D. (2017). Image based book cover recognition and retrieval. In IOP Conference Series: Materials Science and Engineering (Vol. 263, p. 042088). IOP Publishing. https://doi.org/10.1088/1757-899x/263/4/042088

[4] Teoh, K., Ismail, R., Naziri, S., Hussin, R., Isa, M., & Basir, M. (2021). Face Recognition and Identification using Deep Learning Approach. In Journal of Physics: Conference Series (Vol. 1755, Issue 1, p. 012006). IOP Publishing. https://doi.org/10.1088/1742-6596/1755/1/012006

[5] Maheshwari S, Dhanalakshmi N N, Venkatesha D, Santhosh Kumar B V, Alfred D'Souza, Kavya A P, 2018, Library Automation, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCESC – 2018 (Volume 6 – Issue 13),

[6] K. Puritat and K. Intawong, ”Development of an Open Source Automated Library System with Book Recommedation System for Small Libraries,” 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), 2020, pp. 128-132, doi: 10.1109/ECTIDAMTNCON48261.2020.9090753.

[7] Kore, Sharada & Patil, Sonal. (2022). LIBRARY AUTOMATION SYSTEM USING RFID. 9. 707-711.

[8] Olatunji, Olabisi & Dr, Farouq, & Mr, Idris,. (2020). Challenges to the Implementation of Open Source Integrated Library Management Software in Academic Libraries: A Case of Kano University of Science and Technology Library.

[9] B, Pranav & J, Manikandan. (2020). Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks. Procedia Computer Science. 171. 1651-1659. 10.1016/j.procs.2020.04.177.

[10] Islam Chowdhury, Atiqul & Rahman, Mushfika & Sakib, Nazmus. (2019). A Study on Multiple Barcode Detection from an Image in Business System. International Journal of Computer Applications. 181. 30-37. 10.5120/ijca2019918340.

[11] Sahana K Adyanthaya, 2020, Text Recognition from Images: A Study, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCCDS – 2020 (Volume 8 – Issue 13),

[12] M. Brisinello, R. Grbić, D. Stefanovič and R. Pečkai-Kovač, "Optical Character Recognition on images with colorful background," 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2018, pp. 1-6, doi: 10.1109/ICCE-Berlin.2018.8576202.

[13] "pytesseract" pypi.org https://pypi.org/project/pytesseract/ (accessed: October 7,2022)

[14] "OpenCV: Cascade" docs.opencv.org https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html (accessed: October 7,2022)

[15] "dlib C++ Library" dlib.net http://dlib.net/ (accessed: October 7,2022)

[16] "face-recognition" pypi.org https://pypi.org/project/face-recognition/ (accessed: October 7,2022)

[17] "Retrofit" square.github.io https://square.github.io/retrofit/ (accessed: October 7,2022)

[18] "Firebase Authentication" firebase.google.com https://firebase.google.com/docs/auth (accessed: October 7,2022)

[19] "Payment Method — Razorpay" razorpay.com https://razorpay.com/docs/payments/payment-button/ (accessed: October 7,2022)

[20] "Glide v4: Fast and efficient image loading for Android" bumptech.github.io https://bumptech.github.io/glide/ (accessed: November 29,2022)

[21] "Google Books APIs" developers.google.com https://developers.google.com/books (accessed: October 7,2022)

[22] "LFW Face Database : Main" vis-www.cs.umass.edu. http://vis-www.cs.umass.edu/lfw/lfw.tgz (accessed: October 7,2022)

Automated Library Management System

Malhar Bangdiwala
Computer Engineering
Sardar Patel Institute of Technology
Mumbai, India
malhar.bangdiwala@spit.ac.in

Sakshi Mahadik
Computer Engineering
Sardar Patel Institute of Technology
Mumbai, India
sakshi.mahadik@spit.ac.in

Yashvi Mehta
Computer Engineering
Sardar Patel Institute of Technology
Mumbai, India
yashvi.mehta@spit.ac.in

Abhijeet Salunke
Computer Engineering
Sardar Patel Institute of Technology
Mumbai, India
abhijeet_salunke@spit.ac.in

Dr. Rita Das
Applied Science and Humanities
Sardar Patel Institute of Technology
Mumbai, India
rita_das@spit.ac.in

***Abstract*—Traditional methods of managing a library have their disadvantages like long waiting queues and expensive scanner machines. This paper aims to overcome these drawbacks by introducing a new application that makes use of face and text recognition to issue books from the library. The main idea is that a picture of the book with the issuer will be taken from which the required data will be extracted. The account of the user will be created by the library admin by uploading a picture of the face of the user. The application consists of two types of logins - user and admin, where administrators can keep track of user transactions and history. The face recognition algorithm and book recognition algorithm gave an overall accuracy of 99.43% and 82.5% respectively. The user can also see the information related to books and this information is extracted from a third-party API. Further, easy payment and reissue options for the books on the application have also been implemented.**

I. Introduction

The role of the library can be considered a vital and important component of an educational institution, which is a main means of teaching, learning activities, etc. Students, researchers, professors, and others can explore huge resources of information obtained from libraries. The largely increasing number of library books and increasing need for books and information is difficult to manage with traditional manually operating libraries. The slow-moving process of scanning causes long queues in libraries. Automated library systems can increase efficiency and hence, universities should adopt such systems for reducing manual labor.

A major deterrent in libraries while issuing books is the long waiting lines and the slow process of scanning. Students and the public have to wait in long queues to issue books. Further, for the member, creating and possessing a library ID for issuing and returning books can be a tedious process. Adding personal details as well as carrying the library card can sometimes be inconvenient. Additionally, the hardware cost of scanners in libraries makes setting up a library more difficult in economically backward areas. Human errors while issuing books may lead to monetary losses for members. The availability of a book is not transparent for members, that is, members may not know which books are available in the library. The proposed solution aims to overcome all these problems by proposing an automated library system that takes the picture of the borrower as well as the book instead of scanning.

In this paper, an application to automate the library process has been created. A picture containing the member has to be taken, along with the book. The system then extracts the member id from the face and the title of the book from the ISBN number simultaneously by using face recognition and book recognition algorithms respectively. Further, easy re-issue of books and payment of overdue books can also be done. The application also consists of information related to the book such as the quantity of the book which is available in the library, the author, and the ISBN. The admin has access to the history of all members and their transactions. The members are notified of the return date and any overdue they have, and they can pay through the application itself which saves their time and cost of travel.

II. Literature Survey

Authors of the paper titled Face Recognition System [1] propose an automated face recognition system where they neural networks, KLT Algorithm, Viola-Jones Algorithm, Haar classifier, and PCA algorithm. Viola-Jones Algorithm and Haar classifier have been used for face detection. Neural networks further identify the detected image. The proposed method achieves better accuracy and the usage of PCA outperforms the existing work. Raktim Nath, Kaberi Kakoty, and Dibya Jyoti Bora propose face recognition using an HOG detector in their paper titled Face Detection and Recognition using Machine Learning [2]. They put forward that HOG results in more accuracy than Haar-Cascade since Haar Cascade gives more false positives. Their model was able to detect 22 faces in one frame with 1 false positive and could not detect 3 people. Even though the algorithm gives more accuracy and productiveness, it is however a very time-consuming algorithm. In the paper titled Image based book cover recognition and retrieval [3], authors - Kalyani Sukhadan, V Vijayarajan, A Krishnamoorthi, and D Geraldine Bessie Amali propose a graphical user interface powered by MATLAB that allows users to instantly check book-related data. They used the MSER algorithm to detect useful regions (text regions) in the image and compared MATLAB OCR with Tesseract OCR for accuracy. In terms of accuracy, Tesseract OCR was 86%, and MATLAB OCR was 78%. In terms of time cost, MATLAB OCR is faster than Tesseract OCR. KH Teoh, RC Ismail1, SZM Naziri, R Hussin, MNM Isa, and MSSM Basir have implemented a face recognition method using OpenCV and Haar Classifier in their paper - Face Recognition and Identification using Deep Learning Approach [4]. They describe the idea of designing and creating a facial recognition system utilizing deep learning and Python's OpenCV library. Deep learning ensures high accuracy and their final accuracy is 91.7%. They conclude that the accuracy of face recognition on images is higher than the accuracy of real-time video because of the high resolution of images. Maheshwari S et al. [5] have proposed a system wherein the library is automated by adding additional features such as separate logins for users and librarians. This would enable the generation of unique reports as well reservation of books. The system also incorporates notification features when the return date is near. The system also eliminates the need for library cards for each book. However, manual work is still involved in issuing the book (scanning the book and user id card. K.Puritat and K. Intawong [6] have proposed a library system that is more focused on recommendations. For the same, they have used the Chiangmai school library database. In their system, individual logins enable the user to get custom recommendations based on the history of the books that they have issued. To make these recommendations, they have made use of Support Vector Machines. Three sources of data have been combined for the same: Title similarity based on the Damerau–Levenshtein distance, The Dewey Decimal Classification for the book classification system, and Bibliographic Information of book similarity based on multiple information of the bibliography. S. Korade and S. Patil [7] have leveraged RFID technology for automating library processes. This involves reading the barcode of the book and the member id. The software enables the users to log in and issue the book. Fines can also be paid using the app. However, there is no effort to reduce manual interference via the app. O. Olatunji et. al. [8] have studied the issues present in the current automated library system that is being used at Kano University of Science and Technology. In this university, NewgenLib software is being used. While many of the advantages have been highlighted, some of the disadvantages that have come to light are the need for training of librarians to use the system as well as the constant need for a power supply that is required to scan and enter the data into the software. Manikandan J et. al [9], proposed a face recognition system for real-time application using CNN. This is followed by the analysis and evaluation of the system by using alternate CNN attributes or parameters to increase the accuracy and performance of the system. Initially, they used the ATT dataset which consisted of 400 images and this model is compared with previous work. After successfully evaluating the ATT dataset system, real-time inputs through the camera are used to evaluate the proposed system. For the ATT dataset, the maximum precision obtained is 98.75% and for real-time inputs, the highest accuracy obtained is 98.00%. Atiqul Islam Chowdhury, Mushfika Sharmin Rahman, and Nazmus Sakib [10], implemented a system that can simultaneously detect multiple product barcodes using the Zbar Barcode Reader library. Image processing is used to identify these barcodes. 1D and 2D barcodes are detected in a single image and the system is angle invariant and provides minimum interaction by the users. The paper had the limitation of not correctly detecting barcodes because of light reflection, image contrast, and other external factors. Sahana K Adyanthaya [11], discusses various steps involved in text recognition. The application of text recognition and why it's needed is also discussed in the paper. The paper talks about preprocessing of images where images are converted to a suitable format and any kind of noise are removed and the images are processed into binary. Then, to separate each character, the segmentation process is applied using different techniques. Lastly, the paper discusses the classification process to identify the text. Ratko Grbi ́c et. al [12], proposed a way to increase the efficiency of Tesseract Optical Character Recognition (OCR) on colorful images. The method consisted of creating clusters of pixels using the KNN algorithm. Then the texts in the selected cluster are classified using the Naive Bayes classifier. For text recognition, OCR is performed on the selected cluster. The results showed an improved performance in OCR by around 20%. The final accuracy achieved was 95.5%.

III. Design and Methodology

A. Algorithm Details

For the Book Recognition module, the Tesseract OCR engine is being leveraged. Since the code is being written in Python, Pytesseract [13] is being used. The image is first stored in .tiff format. Then, the image is converted to greyscale and is zoomed in for better results. Haar classifier [14] is used to detect a book in the image. Further, the image is then rotated by 90 degrees until the image text can be converted into a string. Next, regex is used to extract the ISBN number from the obtained string. To obtain information about the book, this ISBN is passed to the Google Books API [15].

For the Face Recognition module, the dlib [16] library is being used. Since the code is being written in Python, its python wrapper module: face_recognition [17] has been used. First, an image of the user is first broken down into 128 data points. This is the reference point. This data, along with the name of the user is stored in separate CSV files. When the image has to be recognized, these two CSV files are first read. The image is then scanned for faces using the hog model. The new image is then broken down into similar 128 data points and then compare to the existing faces. The difference is calculated and the closest face is given as the output, provided the distance is under a pre-defined limit.

These two modules are bundled in a Flask app that has been hosted on an Azure VM. The APIs for the two modules are called from the Android app using Retrofit [18]. The login module has been implemented using Java. Firebase's authentication module [19] has been used for the same. The payments module has been implemented using the Razorpay API [20]. The notification module has been implemented as a python script on the Azure Virtual Machine. Cron job scheduler has been used to run the script on a daily basis. The rest of the modules have been implemented using Java by building their respective logic using the Android Studio IDE.

B. Proposed System

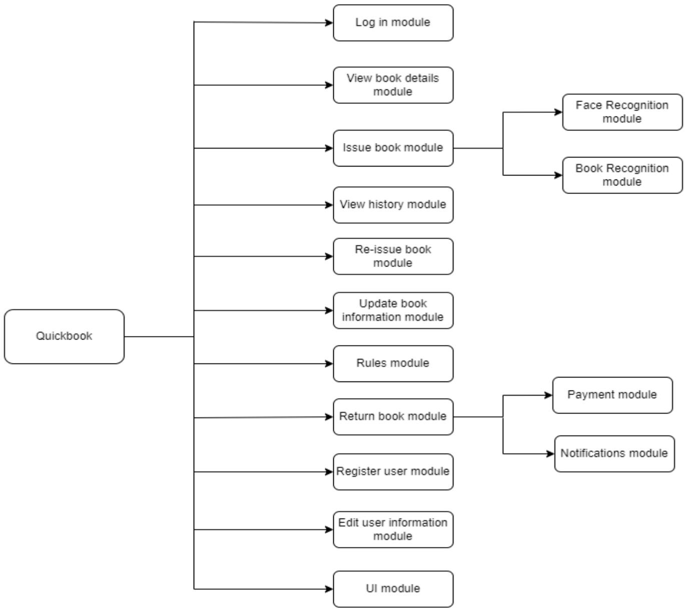


Figure 1. Block Diagram

Fig.1 shows the proposed block diagram of the application. This diagram highlights all the major modules of the application. A Flask app to expose the APIs for the two modules: book recognition and face recognition, as well as for searching a book has been created. There are three APIs in total. One API is to register a new user's face on the platform. The face is stored as numerical data of 128 data points in a CSV file. The second API is to identify the face from an image, provided that the face has already been registered previously. This API also identifies the book's ISBN number from the image. Details regarding the face and book are then returned. The third API is to return all the data related to a book with the help of Google Books API. This Flask app in turn has been hosted on an Azure Virtual Machine on a public IP Address. Thus, the APIs are available for everyone to use.

In the android app, the user would first encounter a splash screen and the login page. Upon logging in, two flows are possible based on the type of user: admin and member.

If the user is an admin, the following will first be visible: a page where a book can be issued to a member, by clicking the picture. Once the image has been taken, the results are displayed on the screen. These results are editable to account for prediction errors if any. The book can then be issued by clicking on the "Issue Book" button. There is also a page for registering a new member. The admin can click an image of the user and enter the corresponding member ID and email ID. This information will be stored in the database. The face will be stored as numerical data as described above. Then, there is the settings page where the admin can edit certain rules of the library: Duration of a book issued, Number of consecutive issues allowed, and the fine imposed per day in case of a late return by a member. On this page, the admin can also change the account password. From here, the admin can log out from the app as well. A "Search books" page has been created where the admin can search for any book by its name. Google Books API is leveraged for the same. The results that are given by the Google Books API are then cleaned and relevant information is passed to the app. Here, the admin will now be able to see a list of all books matching the search title. On clicking on any book, the admin will be able to see basic details such as title, authors, publication information, description, and the cover page of the book. An editable field for storing the number of copies available in the library has also been provisioned. Finally, there is an "overview" page, where the admin can check all the currently issued books. The admin can wish to accept the return of any book by a member from here. A search bar to search for currently issued books by a certain member has been enabled. If the admin, accepts a return, a fee is calculated for a late return. This is an editable field (to account for damages to the book etc). Once submitted, the fees will be reflected on the member's side.

If the user is a member, an overview of all the books issued by him will be visible. If applicable, an option to reissue the book will also be present. Similarly, the option to pay the fee for a specific book might also be present. The payment system has been implemented by Razorpay. A reminder email, 2 days prior to the return date will be sent to the member. This has been enabled by using the Cron scheduler on the virtual machine. These reminders would be sent at 12:30 PM IST. A settings page has been created where the member can change the current password. Here, the rules, as mentioned above are also visible. However, these rules are not editable. A search page similar to the admin's section has also been implemented. The only difference is that the member can only view the number of copies available in the library, and not edit it.

There are however a few things that have been assumed while creating the said application. The main assumption being that the face and book should be clearly visible in the picture. Secondly, a stable internet connection is needed while recognising face and book. Further, since the book data is not being stored in the database and depends on a third-party

API, there is a dependence on it. Lastly, information such as the member ID and the book stock data should be provided by the library initially.

C. Results

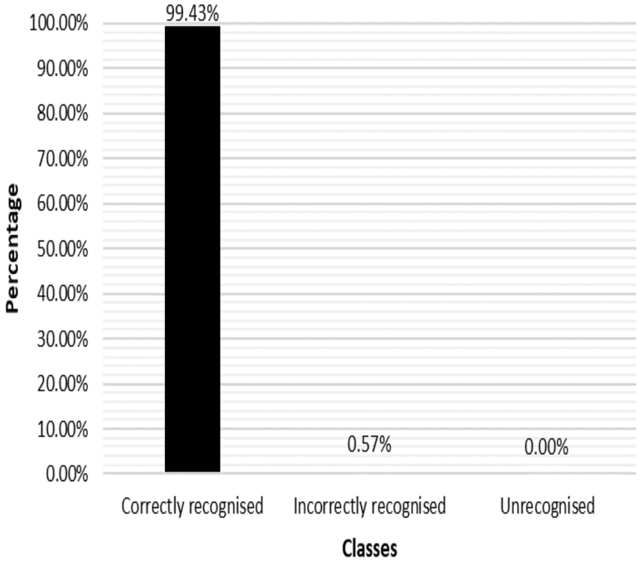


Figure 2. Accuracy graph for Face Recognition Algorithm

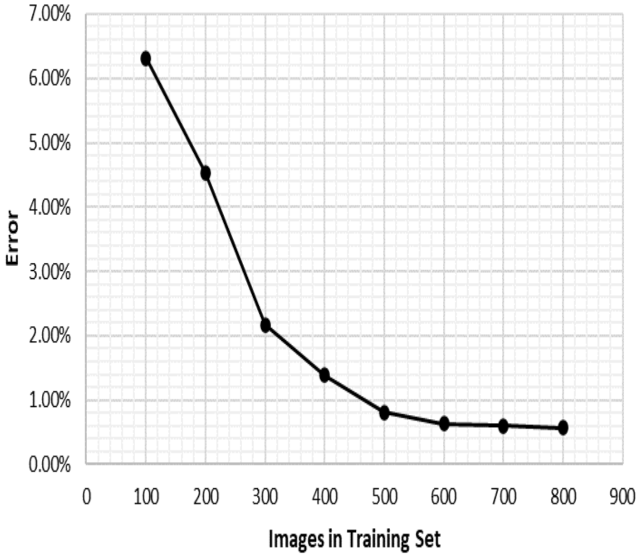


Figure 3. Accuracy graph for Face Recognition Algorithm

1) Face Recognition: The face recognition algorithm has been tested on the Labelled Faces in the Wild (LFW) [21] dataset. After cleaning, this dataset contained images of over 800 famous personalities from different angles and expressions. 1 photo of each celebrity was used for training purposes. After training, the remaining photos were used as the testing dataset. More than 2100 images were used for this purpose. The algorithm was able to detect a face in all the images. An overall accuracy of 99.43% was obtained. This has been captured in Fig.2 above. Fig.3 depicts the error rate with varying lengths of training set. When there were only 100 images in the training set, an error rate of 6.31% was obtained. The error rate dropped to 0.57% when 800 images were used.

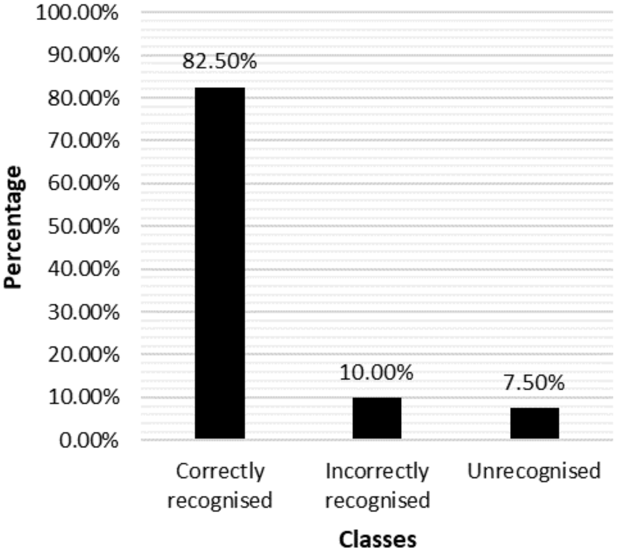


Figure 4. Accuracy graph for Book Recognition Algorithm

2) Book Recognition: The book recognition algorithm has been tested on a manually curated dataset of about 80 book images that contained the ISBN numbers in it. These images were clicked in different lighting and different angles. In this, 66 ISBN numbers were correctly identified, 8 were incorrectly identified and 6 were not identified at all. This meant an overall accuracy of 82.5%. This has been depicted in Fig.4 above.

IV. Conclusion & Future Work

Although the app successfully recognizes faces and books and implements all the functionalities of a typical library, the functionality can be extended to include features like the recommendation of books to users. We want the app to be more interactive and suggest books to users as per their needs. Further, currently, the app only recognizes a single book and face in one picture. The future scope also includes recognising multiple faces and books from a single picture so that the process becomes faster and more convenient.

References

[1] Shivam Singh, Prof. S. Graceline Jasmine, 2019, Face Recognition System, International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 Vol. 8 Issue 05, May-2019

[2] Nath, Raktim & Kakoty, Kaberi & Bora, Dibya & Welipitiya, Udari. (2021). Face Detection and Recognition Using Machine Learning. 43. 194-197.

[3] Sukhadan, K., Vijayarajan, V., Krishnamoorthi, A., & Geraldine Bessie Amali, D. (2017). Image based book cover recognition and retrieval. In IOP Conference Series: Materials Science and Engineering (Vol. 263, p. 042088). IOP Publishing. https://doi.org/10.1088/1757-899x/263/4/042088

[4] Teoh, K., Ismail, R., Naziri, S., Hussin, R., Isa, M., & Basir, M. (2021). Face Recognition and Identification using Deep Learning Approach. In Journal of Physics: Conference Series (Vol. 1755, Issue 1, p. 012006). IOP Publishing. https://doi.org/10.1088/1742-6596/1755/1/012006

[5] Maheshwari S, Dhanalakshmi N N, Venkatesha D, Santhosh Kumar B V, Alfred D'Souza, Kavya A P, 2018, Library Automation, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCESC – 2018 (Volume 6 – Issue 13),

[6] K. Puritat and K. Intawong, "Development of an Open Source Automated Library System with Book Recommedation System for Small Libraries," 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), 2020, pp. 128-132, doi: 10.1109/ECTIDAMT-NCON48261.2020.9090753.

[7] Kore, Sharada & Patil, Sonal. (2022). LIBRARY AUTOMATION SYSTEM USING RFID. 9. 707-711.

[8] Olatunji, Olabisi & Dr, Farouq, & Mr, Idris,. (2020). Challenges to the Implementation of Open Source Integrated Library Management Software in Academic Libraries: A Case of Kano University of Science and Technology Library.

[9] B, Pranav & J, Manikandan. (2020). Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks. Procedia Computer Science. 171. 1651-1659. 10.1016/j.procs.2020.04.177.

[10] Islam Chowdhury, Atiqul & Rahman, Mushfika & Sakib, Nazmus. (2019). A Study on Multiple Barcode Detection from an Image in Business System. International Journal of Computer Applications. 181. 30-37. 10.5120/ijca2019918340.

[11] Sahana K Adyanthaya, 2020, Text Recognition from Images: A Study, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCCDS – 2020 (Volume 8 – Issue 13),

[12] M. Brisinello, R. Grbić, D. Stefanovič and R. Pečkai-Kovač, "Optical Character Recognition on images with colorful background," 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2018, pp. 1-6, doi: 10.1109/ICCE-Berlin.2018.8576202.

[13] "pytesseract" pypi.org https://pypi.org/project/pytesseract/ (accessed: October 7,2022)

[14] "OpenCV: Cascade Classifier" docs.opencv.org https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html (accessed: October 7,2022)

[15] "Google Books APIs | Google Developers" developers.google.com https://developers.google.com/books (accessed: October 7,2022)

[16] "dlib C++ Library" dlib.net http://dlib.net/ (accessed: October 7,2022)

[17] "face-recognition" pypi.org https://pypi.org/project/face-recognition/ (accessed: October 7,2022)

[18] "Retrofit" square.github.io https://square.github.io/retrofit/ (accessed: October 7,2022)

[19] "Firebase Authentication" firebase.google.com https://firebase.google.com/docs/auth (accessed: October 7,2022)

[20] "Payment Method | Razorpay Docs" razorpay.com https://razorpay.com/docs/payments/payment-button/ (accessed: October 7,2022)

[21] "LFW Face Database : Main" vis-www.cs.umass.edu. http://vis-www.cs.umass.edu/lfw/lfw.tgz (accessed: October 7,2022)

REPORT OF THE PLAGIARISM CHECK

THIS REPORT CERTIFIES THAT THE ATTACHED WORK

Report Paper Plagiarism

WAS CHECKED WITH THE PLAGIARISM PREVENTION SERVICE MY.PLAGRAMME.COM

AND HAS:

SIMILARITY

2%

RISK OF THE PLAGIARISM

2%

PARAPHRASE

0%

IMPROPER CITATIONS

0%

File name: Research Paper.docx
File checked: 2022-12-07
Report generated: 2022-12-07

REPORT OF THE PLAGIARISM CHECK

THIS REPORT CERTIFIES THAT THE ATTACHED WORK

Blackbook

WAS CHECKED WITH THE PLAGIARISM PREVENTION SERVICE
MY.PLAGRAMME.COM AND HAS:

SIMILARITY

3%

RISK OF THE PLAGIARISM

1%

PARAPHRASE

0%

IMPROPER CITATIONS

0%

File name: Blackbook.docx
File checked: 2022-12-07
Report generated: 2022-12-07