Development Document

I. Pseudocode

1. Prompt user to show their ID using speech recognition.  
2. Connect to the database using provided credentials.  
3. Capture the user's login details via barcode scanning.  
4. Query the database to retrieve the borrowed amount and status for the user.  
5. Check if the user was borrowing an eBook or a physical book.  
6. If it’s an eBook, update the database to clear the borrowed book, status, and time borrowed.  
7. If it’s a physical book, follow the same steps to clear the borrowed book information.  
8. Log the time of return in the logbook.  
9. Commit all changes to the database.  
10. Close the database connection.  
11. Handle any database errors or other exceptions gracefully.  
12. Thank the user for returning the book and prompt for any further action.

II. Block of Code

import os  
import mysql.connector  
import Borrow\_Book  
  
def return\_book():  
 db\_config = {  
 'host': 'localhost',  
 'user': 'root',  
 'password': '',  
 'database': 'studentqr'  
 }  
   
 Borrow\_Book.speak("Show your ID")  
   
 try:  
 conn = mysql.connector.connect(\*\*db\_config)  
 cursor = conn.cursor()  
   
 user\_name, barcode = Borrow\_Book.login\_and\_capture(db\_config)  
   
 if user\_name:  
 cursor.execute("SELECT borrowedamount FROM qr WHERE name = %s", (user\_name,))  
 borrowed\_amount = cursor.fetchone()  
   
 cursor.execute("SELECT status FROM qr WHERE name = %s", (user\_name,))  
 ebook\_status = cursor.fetchone()  
   
 if ebook\_status and ebook\_status[0] == "Borrowing an Ebook":  
 update\_query = """  
 UPDATE qr  
 SET borrowedbook = '', timeborrowed = '', status = '', borrowedamount = ''  
 WHERE name = %s  
 """  
 cursor.execute(update\_query, (user\_name,))  
 else:  
 update\_query = """  
 UPDATE qr  
 SET borrowedbook = '', timeborrowed = '', status = '', borrowedamount = ''  
 WHERE name = %s  
 """  
 cursor.execute(update\_query, (user\_name,))  
   
 log\_query = """  
 UPDATE logbook  
 SET Time\_returned = %s  
 WHERE STUD\_ID = %s AND Time\_returned = ''  
 """  
 cursor.execute(log\_query, (Borrow\_Book.date\_and\_time(), barcode))  
   
 conn.commit()  
 Borrow\_Book.speak(f"Thank you, {user\_name}, your book has been returned.")  
   
 except mysql.connector.Error as db\_error:  
 print(f"Database error: {db\_error}")  
 Borrow\_Book.speak("There was an error with the database.")  
   
 except Exception as e:  
 print(f"An unexpected error occurred: {e}")  
 Borrow\_Book.speak("An unexpected error occurred.")  
   
 finally:  
 if cursor:  
 cursor.close()  
 if conn:  
 conn.close()  
   
if \_\_name\_\_ == '\_\_main\_\_':  
 os.system('cls' if os.name == 'nt' else 'clear')  
 return\_book()

III. Code Explanation

1. Database Configuration: The `db\_config` dictionary contains the credentials needed to connect to the MySQL database. This is used to establish the database connection.  
2. Speech Interaction: The `Borrow\_Book.speak()` function is used to prompt the user to show their library ID, making the interaction vocal.  
3. Login and Barcode Capture: The function `Borrow\_Book.login\_and\_capture()` captures the user’s name and their barcode by scanning their ID using a barcode reader.  
4. Database Queries: The code queries the `qr` table to fetch the borrowed amount and the status (eBook or physical book) of the user. Depending on the result, the code determines how to update the database.  
5. Return Process: If the user was borrowing an eBook, the status, borrowed book, and other related information are cleared. The same steps are taken for physical books.  
6. Logbook Update: The `log\_query` updates the `logbook` table with the time the book was returned. This ensures that return transactions are logged.  
7. Database Commit:After all queries are executed, `conn.commit()` is used to save the changes in the database.  
8. Error Handling: The code has two `except` blocks: one for MySQL-specific errors (`mysql.connector.Error`) and a general block for catching unexpected errors. This ensures that the user receives feedback if something goes wrong.  
9. Final Cleanup: Whether an error occurs or not, the `finally` block ensures that the cursor and connection to the database are closed properly.