```
import java.util.ArrayList;
import java.util.List;

abstract class Book {
    protected String title;
    protected String author;
    protected int availableCopies;

    public Book(String title, String author, int availableCopies) {
        this.title = title;
        this.author = author;
        this.availableCopies = availableCopies;
    }

    public abstract void displayInfo();

    public boolean checkAvailability() {
        return availableCopies > 0;
    }

    public void borrowBook() throws Exception {
        if (availableCopies > 0) {
            availableCopies--;
        } else {
            throw new Exception("Book is not available.");
        }
    }

    public void returnBook() {
        availableCopies++;
    }

    public String getTitle() {
        return title;
    }

    public int getAvailableCopies() {
        return availableCopies;
    }
}

class Fiction extends Book {
    private String genre;

    public Fiction(String title, String author, int availableCopies, String genre) {
        super(title, author, availableCopies);
        this.genre = genre;
    }

    @Override
    public void displayInfo() {
        System.out.println("Fiction Book: " + title + " by " + author + ", Genre: " + genre + ",
Available copies: " + availableCopies);
    }

    public String getGenre() {
        return genre;
    }
}

class NonFiction extends Book {
    private String subject;

    public NonFiction(String title, String author, int availableCopies, String subject) {
        super(title, author, availableCopies);
        this.subject = subject;
    }

    @Override
    public void displayInfo() {
        System.out.println("Non-Fiction Book: " + title + " by " + author + ", Subject: " + subject + ",
Available copies: " + availableCopies);
    }

    public String getSubject() {
        return subject;
    }
}

class Reference extends Book {

    public Reference(String title, String author, int availableCopies) {
        super(title, author, availableCopies);
    }

    @Override
    public void displayInfo() {
        System.out.println("Reference Book: " + title + " by " + author + ", Available copies: " +
availableCopies);
    }
}

class Library {
    private List<Book> books;

    public Library() {
        books = new ArrayList<>();
    }

    public void addBook(Book book) throws Exception {
        for (Book b : books) {
            if (b.getTitle().equals(book.getTitle())) {
                throw new Exception("Book already exists in the library.");
            }
        }
        books.add(book);
    }

    public void removeBook(String title) throws Exception {
        Book bookToRemove = null;
        for (Book b : books) {
            if (b.getTitle().equals(title)) {
                bookToRemove = b;
                break;
            }
        }
        if (bookToRemove != null) {
            books.remove(bookToRemove);
        } else {
            throw new Exception("Book not found in the library.");
        }
    }

    public Book searchBook(String title) {
        for (Book b : books) {
            if (b.getTitle().equals(title)) {
                return b;
            }
        }
        return null;
    }

    public void displayBooks() {
        for (Book b : books) {
            b.displayInfo();
        }
    }
}

class User {
    private String name;
    private String libraryCardNumber;
    private List<Book> borrowedBooks;
    private static final int MAX_BORROW_LIMIT = 3;

    public User(String name, String libraryCardNumber) {
        this.name = name;
        this.libraryCardNumber = libraryCardNumber;
        this.borrowedBooks = new ArrayList<>();
    }

    public String getName() {
        return name;
    }

    public String getLibraryCardNumber() {
        return libraryCardNumber;
    }

    public void borrowBook(Book book) throws Exception {
        if (borrowedBooks.size() >= MAX_BORROW_LIMIT) {
            throw new Exception("Borrow limit reached. You cannot borrow more than " +
MAX_BORROW_LIMIT + " books.");
        }

        if (book.checkAvailability()) {
            book.borrowBook();
            borrowedBooks.add(book);
            System.out.println(name + " borrowed: " + book.getTitle());
        } else {
            throw new Exception("Book is not available.");
        }
    }

    public void returnBook(Book book) {
        if (borrowedBooks.contains(book)) {
            borrowedBooks.remove(book);
            book.returnBook();
            System.out.println(name + " returned: " + book.getTitle());
        } else {
            System.out.println(name + " did not borrow the book: " + book.getTitle());
        }
    }

    public void displayBorrowedBooks() {
        if (borrowedBooks.isEmpty()) {
            System.out.println(name + " has not borrowed any books.");
            return;
        }
        System.out.println(name + "'s Borrowed Books:");
        for (Book b : borrowedBooks) {
            System.out.println(b.getTitle());
        }
    }
}

public class LibraryManagementSystem {
    public static void main(String[] args) {
        try {
            Library library = new Library();
            User user = new User("Alice", "U001");

            Fiction fictionBook = new Fiction("The Great Gatsby", "F. Scott Fitzgerald", 2, "Classic");
            NonFiction nonFictionBook = new NonFiction("Sapiens", "Yuval Noah Harari", 1,
"History");
            Reference referenceBook = new Reference("Oxford Dictionary", "Oxford", 3);

            library.addBook(fictionBook);
            library.addBook(nonFictionBook);
            library.addBook(referenceBook);

            library.displayBooks();

            user.borrowBook(fictionBook);
            user.borrowBook(nonFictionBook);
            user.displayBorrowedBooks();

            user.returnBook(fictionBook);
            user.displayBorrowedBooks();

            library.removeBook("Sapiens");
            library.displayBooks();

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```