Design and Implementation of a Mobile Library Management System for Improving Service Delivery

Ndukwe Oke Eke [1] **, Ibrahim Anka Salihu** [1]

[1] *Nile University of Nigeria*
Plot 681, Cadastral Zone C-OO, Research & Institution Area, Jabi Airport Bypass, Abuja FCT, 900001, Nigeria

DOI: 10.22178/pos.69-7

LCC Subject Category: Z729-875

Received 20.03.2021
Accepted 27.04.2021
Published online 30.04.2021

Corresponding Author:
Ndukwe Oke Eke
ndukwe.eke@nileuniversity.edu.ng

Abstract. A mobile library management system provides a more efficient way of managing library processes and rendering effective library services irrespective of time and place. This research work aimed to develop a Mobile Library Management System for the Nile University of Nigeria Library to overcome the challenges that hinder the librarians from managing the library processes on the go. The android mobile library management system was developed using Android Studio, HTML, CSS, PHP, and MySQLi database. Service Responsibility Table was used in eliciting and documenting the user's requirements for the library management system. The proposed Android-based mobile library system was evaluated through a survey by the librarians. The evaluation has shown that the proposed system is capable of complementing the existing library management systems.

Keywords: Library; Service Responsibility Table; Library Management System; Mobile Applications.

INTRODUCTION

Different kinds of interpretations have been used to define the term "Library". Authors [1] defined a library as an "organized collection of information sources which is made accessible to the people". According to [2] library is a "Large bank where books and other relevant materials which are meant to improve knowledge are kept and retrieved from". Libraries are service-oriented and can be grouped into six types based on the services it provides and to whom these services are rendered. These types include (1) Public libraries, (2) School libraries, (3) Academic libraries, (4) Special libraries, (5) National libraries, and (6) Virtual/Digital libraries.

As the names suggest, Public libraries are those libraries established and open to the general public; examples of such libraries include Abia State Library, Akwa Ibom State Library, Bida Library, etc. School libraries serve the Primary and Secondary pupils. Academic libraries are libraries established to serve tertiary institutions such libraries can be found in Universities, Polytechnics, and Colleges. Special libraries are neither Academic, School, Public nor National libraries; It provides specialized information resources to a specific group of persons. Special libraries include corporate, museum, military, hospital libraries, etc.

On the other hand, national libraries are the apex library in any nation; the government establishes them as the country's repository of information. Examples of a National library include "National Library of Nigeria (NLN)", "Library of Congress (LC)", "National Library of Britain (BL)", etc. In contrast, Virtual/Digital libraries can best be described as libraries without walls.

A typical library system comprises several departments, sections or units such as the Acquisition department, Cataloguing/Classification department, Reprographic and Reference services department [3]. This may vary from one library to another due to the slight differences that exist among the various types of libraries.

As library evolves and Information Technology (IT) introduced into librarianship, libraries began to incorporate IT in library management for effective and efficient service delivery. Information resources hitherto stored, documented, and access in print format are now digitized and made accessible in electronic format.

In a bid to embrace the new trend, most academic libraries, including the Nile University of Nigeria (NUN) library, have undergone a series of library transformation by automating most of their library services. However, one of the significant challenges that still face librarians today is the lack of an efficient library management system to ensure effective and timely service delivery. Library Management System (LMS) was introduced to tackle some of these difficulties. LMS provides a complete solution for libraries by integrating specialized library functions and services for library users. According to [4], Library Management System allows librarians to maintain and manage library processes by developing a computerized system capable of documenting and tracking various library transactions such as registering library users, loaning and returning library materials, and accessioning. This paper, therefore, proposes an Android-based mobile library management system for academic libraries to ensure adequate library service delivery to its teeming users.

The paper is organized as follows: Section 2 outlines the challenges associated with the manual system of library management and reviews the introduction of the mobile application. Section 3 presents the methodology and demonstrates how SRT is used to elicit and document user requirements. Section 4 explained the procedure of transforming SRT into UML diagrams. Section 5 shows the system implementation and the resulting system interface, and section 6 presents the evaluation of the MLMS App.

Related works

Libraries collect different kinds of information resources. Such information resources can come in print and non-print formats, such as books, journals, periodicals, maps, CDs, DVDs, audiotapes, videotapes, audiovisuals, manuscripts, e-book, audiobooks and many others [1, 2]. Before the integration of Information Technology (IT) in Nigerian libraries in the early 1990s, academic libraries had operated a manual system of library management. This system is generally considered inefficient, prone to "delays, errors, as well as requires the physical and mental efforts of the library staff to manage" [5]. Furthermore, other challenges associated with the manual system of library management include:

– long and unnecessary queues at the circulation desk during user registration;

– unnecessary delays in library transactions such as issuing/returning of books;

–prone to human errors, e. g. documentation and misfiling of library materials;

– inadequate space to house the ever-increasing library collections;

– the mutilation of library materials due to limited copies;

– inability to manage library processes remotely;

– requires many workforces to carry out various library functions such as shelving and shelf-reading, cataloguing and classification of library resources.

Mobile Technology. Mobile technology is any technology with the capability that is accessible from anywhere the user is. Such technologies may include Personal Digital Assistant (PDA), notebooks, automobile industries, and mobile phones. A mobile device is a combination of hardware, operating systems, networking and software [6, 7]. With the advancement in mobile technology, many institutions such as banks and libraries, including many social media platforms like Facebook, WhatsApp, Instagram etc., have embraced the act of combining mobile technology with mobile devices to deliver effective and efficient services the user. Today, a more significant number of students and teachers use smartphones and tablet PCs to search for e-journals, eBooks, and other electronic resources. This has led to the introduction of mobile technologies in library services and the development of mobile information system compatible with mobile devices that allow library users quick access to their desired information [8].

Mobile Applications. Mobile applications, popularly referred to as "mobile apps", are a set of programs or software that runs on mobile devices such as smartphones, tablets, PCs, PDAs etc. and performs specific tasks for the user [9, 10, 11]. Mobile applications run on different platforms like Android, iPhone, windows, and Symbian phones. Some of its apps pre-installed on handheld devices, and others made available at their various app stores (Ovi Store, Play Store, and Microsoft Store).

The importance of mobile apps is visible in social networks, videos and audios, instant messaging, communication, and gaming industries. With the mobile application, people can watch trending videos from YouTube, pay for goods and services using mobile banking, book and pay for flight ticket with e-ticketing, connect with friends and family through the Facebook app, and navigate freely around the globe with the help of Google map app.

Authors [12] categorized mobile apps into three groups: Web apps, Native Apps, and Hybrid apps.

This paper aim at developing a Native App, as well as target a specific group of mobile users on the same platform (android users).

Table 1 summarises the advantages and disadvantages of the various mobile apps as discussed in this section.

Table 1 – Comparing Web, Native and Hybrid Apps

Types of Mobile Application	Advantages	Disadvantages
Web Apps	Easy to build (HTML/ CSS/JS) Easy to maintain Use any technology language Cheaper app for all platform Single app for all platform (i. e., one web)	Need to run in a browser Slower than a native app Less interactive and less intuitive It cannot be submitted to app stores Cannot interact with device utilities
Native Apps	Very fast Designed for a specific platform (with specific SDK) It can be distributed in app stores Interact with device utilities (e. g., Camera)	Single platform (i. e. Android run on android devices) Difficult programming language Very expensive and difficult to maintain
Hybrid App	Easy to develop (HTML/CSS/JS) Cheaper than a native app Single app for all platform No browser needed Access device utilities using API Easy to develop than a native app	Slower than native apps More expensive than web apps Less interactive than native apps

Android Application Development. Mobile app development involves programming a software application that runs on mobile devices to utilize network computing resources. There are various programming environments for mobile app development, such as Xcode for Apple products, Eclipse or Android Studio programming environment for Android devices, Visual Studio environment for Windows etc. [13]. This paper focuses on designing and developing an Android-based library management system application using the Android Studio programming environment, HTML, PHP, CSS, and JavaScript languages to achieve this goal.

Methodology

The system requirements were determined through direct observation of the existing system at the Nile University of Nigeria (NUN) Library and conducting interviews with the librarians and users of the NUN library to obtain direct information about the existing system and its challenges. After that, the Service Responsibility Table (SRT) is used to analyze and verify the requirements, while the Unified Modeling Language (UML) is used to model the requirements.

Eliciting and Documenting User's Requirements. Many software development projects have failed due to the inability to comprehensively capture users' requirements [14]. Eliciting user's requirements, representing and verifying the requirements can be very challenging [14]. UML has been the standard tool used by software engineers in requirements elicitation, analysis, modelling, verification and design. However, there are many criticisms that stakeholder (Non-

IT professional) finds it challenging to understand the requirements represented in UML [14]. In this research, Service Responsibility Tables (SRT) is used for requirement elicitation and verification, while UML is used for modelling. Use Case and Class diagrams will be used to model the proposed system.

Service Responsibility Table. Service Responsibility Table (SRT) is a lightweight analytical tool developed to supplement the Unified Modelling Language (UML) to tackle the problem of inadequate user involvement in system development [14, 15]. SRT is a two-column swim-lane diagram that specifies the responsibilities of both the service provider and the customers. The benefits of SRT over UML according to [15] can be summarised as follows: (1) it clarifies the scope and context of the process without requiring mastery of details, (2) it focuses attention on activities and responsibilities rather than on details of technology and information, (3) it identifies the job roles that are involved, (4) it brings customer responsibilities into the analysis, (5) it identifies service interactions and other steps that are not visible to customers.

Based on the business processes, SRT is used to identify and map all processes to the service providers and customers. The SRT for the system is presented in table 2.

Table 2 – Service Responsibility Table for Library Management System

No	Provider Activity	Customer Activity	Information used or generated
1	Manage Library Resources	The user identifies the book of interest	Book title, Book Author, and Book ID
2	Add User	Register with the system	Username, Password, First_Name, Last_Name, User_ID, Email, etc.
3	Provides account login details	Login with account details	Username and Password
4	Verify user login	-	-
5	Notify user of successful login	-	-
6	-	Search OPAC	OPAC
7	Access Yordam	-	-
8	Add Book	-	-
9	Modify a book	-	-
10	Delete a book	-	-

Transforming SRT to UML diagrams. Authors [14, 15] have proposed sets of heuristics to transform SRTs to use case and class diagrams and other heuristics to be used in transforming activity and sequence diagrams, respectively. However, this paper will only focus on transforming SRT to use case and Class diagrams.

The system analysis uses the use case diagram to represent the behaviour of a system under various conditions as it responds to request from the actors (users).

Based on these heuristics, the SRT in table 2 can generate a use case diagram. The following steps are considered in the transformation:

1. Admin and user are identified in the SRT (table 2), and they can be considered actors in the use case diagram.

2. The activities and responsibilities in the SRT are summarized as follows:

a. Manage library resources

b. The user identifies the book of interest

c. Add user

d. Register

e. Provides account login details

f. Login with account details

g. Verify user login

h. Notify user of successful login

i. search OPAC

j. Access Yordam

k. Add book

l. Modify a book

m. Delete/Remove a book

3. Based on the expected functionalities of the library management system, “manage library resources”, “user identifies the book of interest”, “provides account login details”, “verify user log-

in", "notify the user of successful login" is excluded from the list of use case. These are considered to be outside of the scope of the library management system. Therefore, eight use cases were identified from the SRT.

4. Link the identified use cases with the corresponding actors based on the descriptions in the SRT. The use cases "modify" and "delete" can be carried out only when "Add book" and "Register User" respectively are completed. While the use cases "Search OPAC" and "Register User" can be executed only when "Login" and "Access Yordam" is completed. This, therefore, shows that there is an "extends" among "Search OPAC", "Login", "Register User", "Add Book", "Modify", "Access Yordam", and "Delete" use cases.

The resulting use case diagram is shown in Figure 1.

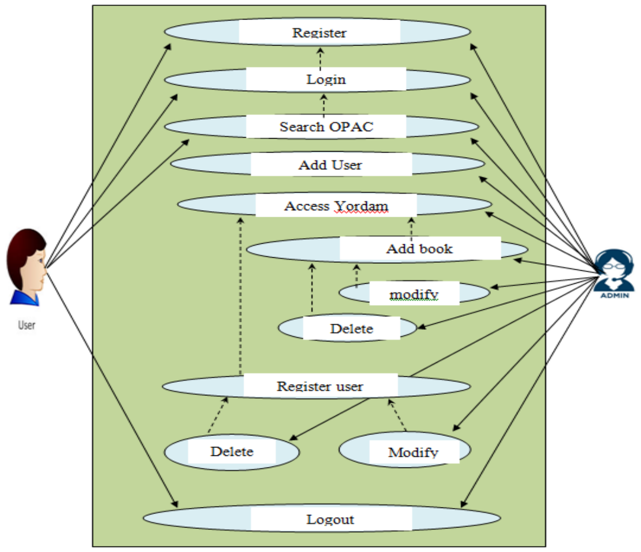


Figure1 – Use Case Diagram transformed from the SRT

Transforming SRT into a Class diagram. The class diagram is used by system analysts to "capture the contents of the domain model" [15]. Domain models are those essential things and concepts associated with a system [15]. Here, the domain models are represented as classes and the relationships among them.

Transforming SRT into a UML class diagram according to [15] can be achieved by extending the SRT to accommodate the three-column based on the heuristics in [15].

1. Identify all the objects mentioned in the three columns of SRT and determine the objects about which the software system will track or maintain information. These will be considered as classes.

2. Decide the names of classes and the relationships among them.

3. Identify the attributes for each class using the contents in the third column of the SRT.

4. Fill in missing classes, attributes, and relationships by further analysis.

From the above heuristics, the SRT in table 2 can also derive a UML class diagram. The following steps were considered while transforming SRT into a UML class diagram:

1. Admin, User, and Book are identified in the SRT, and they are considered as candidates for classes.

2. Here, the identified classes ("Admin", "User", and "Book") and the relationship among them are: "add user", "verify user", "add book", "modify book", and "delete book".

3. The attribute for each class will be identified using the contents in the third column:

a) Book – book title, book author, and book ID;

b) User – username, password, user ID, and Email;

c) Admin – username and password.

4. Classes that are a type of person will be a subclass of the "person" class. The relationships between a person and its subclasses are generalized.

The resulting class diagram is shown in Figure 2 below:

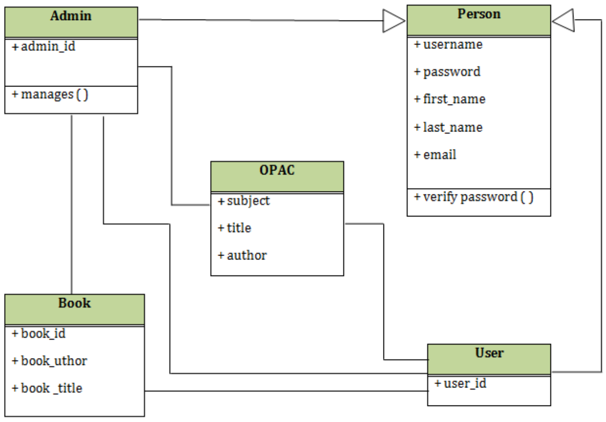


Figure 2 – Class Diagram transformed from the SRT

System Implementation. This system was designed using various tools and programming languages such as Android Studio, HTML, CSS, PHP and MySQLi. The mobile library management system was built as a responsive website then converted into a native app using the "Android Studio". The "Webview" element was used for this conversion. The procedures or steps for the conversion involve:

1. Develop a fully responsive (mobile friendly) website.

2. Create a new project in Android Studio and name your project.

3. Open activity_main.xml, create the application interface and add a web view element to it.

4. Open the "MainActivity.java" file. Declare a review variable, enable JavaScript and load the URL of the website (e. g. nilelib.com.ng).

System Evaluation. An evaluation is performed using defined criteria to understand the usability and effectiveness of any new technology [14]. The proposed system has been evaluated by the users, specifically the administrative staff of the library. The evaluation intends to find out if the proposed system has met its requirements and its usability.

Survey. A survey was conducted where the proposed MLMS was deployed to a smartphone to interact with the system. A total number of 8 respondents comprising mainly the Nile University of Nigeria library staff were selected for this study due to their direct involvement in the process. A questionnaire comprising of 10 questions was used as a means of data gathering for the survey. The questionnaire consists of closed-ended questions using the 5 – points Likert scale measurement, 1 = Strongly agree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly agree.

RESULTS AND DISCUSSION

The result derived from table 3 shows that the mean score for all the indicators is high above four (4), indicating that users rated MLSM App high in all aspect, including user-friendliness, accessibility, and compatibility with the existing system.

Standard deviation (SD) is also used to show variations in responses. SD of 0 is obtained when all the responses to the questions are the same. Low SD means most respondents are centred on the mean, and high SD means there are variations in responses.

The Standard Deviation, as shown in Table 3, is low around the mean score. This indicates that there is no variation in any of the responses. Therefore, the outcome of this survey shows that users find MLMS App effective in managing library processes.

Figure 3 indicates that 87.50% of the total respondents strongly agree that the features of the MLMS App include: Yordam, eBooks, OPAC, Support, Registration Form, Login, and Logout. At the same time, 12.50% agrees to the availability of these features on the MLMS App.

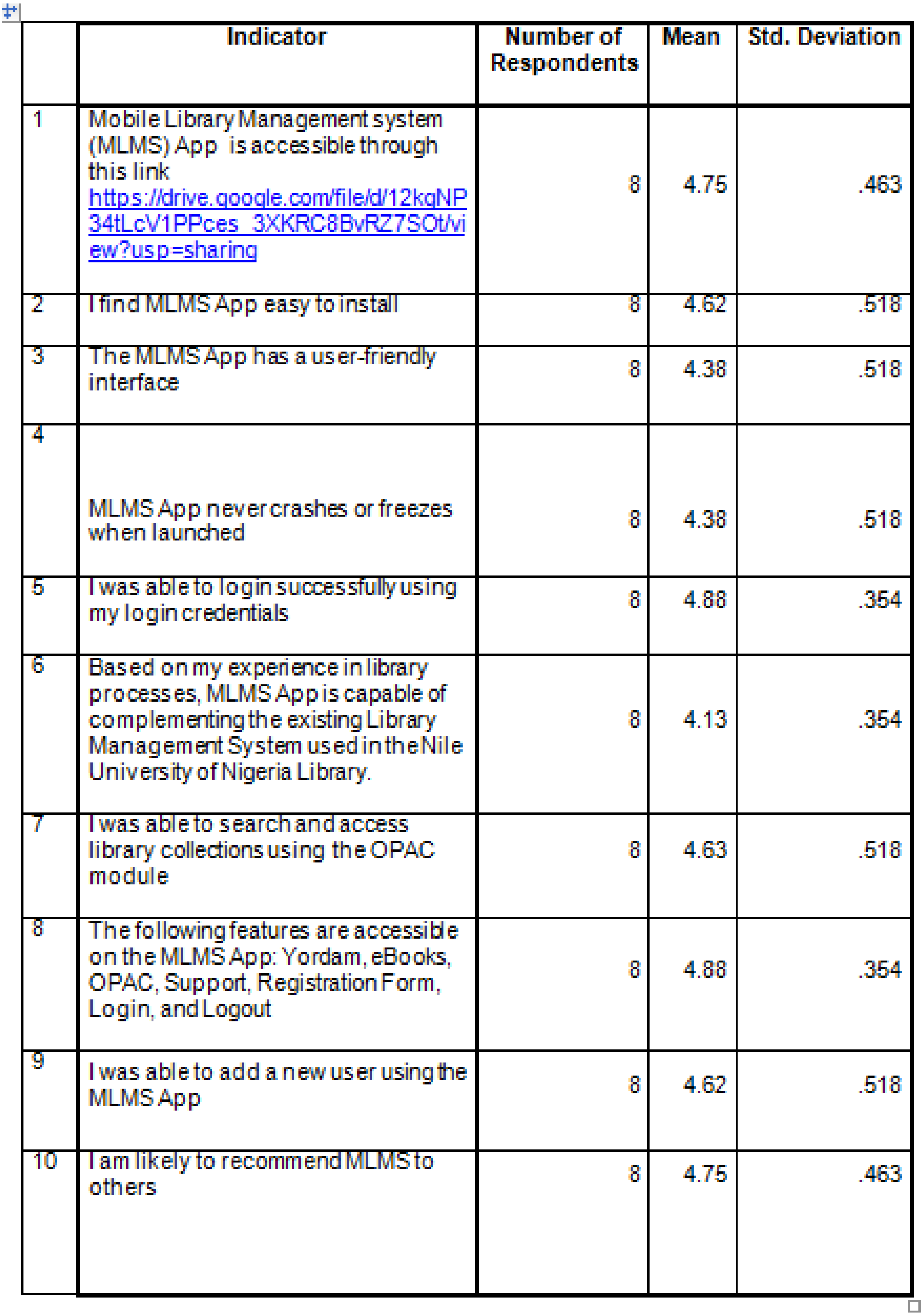
Figure 4 shows that 87.50% of respondents Strongly Agree that MLMS App can complement the existing LMS, while 12.50% Agree the App can complement the existing system.

Figure 5 illustrates that 5 out of the eight respondents Agree that MLMS App has a user-friendly interface. While the remaining three respondents Strongly Agree with the user-friendliness of the MLMS App.

CONCLUSION

The idea of having a MLMS makes the task of the library staff more accessible and less stressful. The developed mobile library management system can complement the existing system (Yordam) to ensure flexible and efficient library management. The system may be improved upon to incorporate Barcode technology for easy capturing of books and other library materials.

Table 3 – Descriptive Statistics for the scale of usability and effectiveness of the MLMS App



	Indicator	Number of Respondents	Mean	Std. Deviation
1	Mobile Library Management system (MLMS) App is accessible through this link https://drive.google.com/file/d/12kqNP34tLcV1PPces_3XKRC8BvRZ7SOt/view?usp=sharing	8	4.75	.463
2	I find MLMS App easy to install	8	4.62	.518
3	The MLMS App has a user-friendly interface	8	4.38	.518
4	MLMS App never crashes or freezes when launched	8	4.38	.518
5	I was able to login successfully using my login credentials	8	4.88	.354
6	Based on my experience in library processes, MLMS App is capable of complementing the existing Library Management System used in the Nile University of Nigeria Library.	8	4.13	.354
7	I was able to search and access library collections using the OPAC module	8	4.63	.518
8	The following features are accessible on the MLMS App: Yordam, eBooks, OPAC, Support, Registration Form, Login, and Logout	8	4.88	.354
9	I was able to add a new user using the MLMS App	8	4.62	.518
10	I am likely to recommend MLMS to others	8	4.75	.463

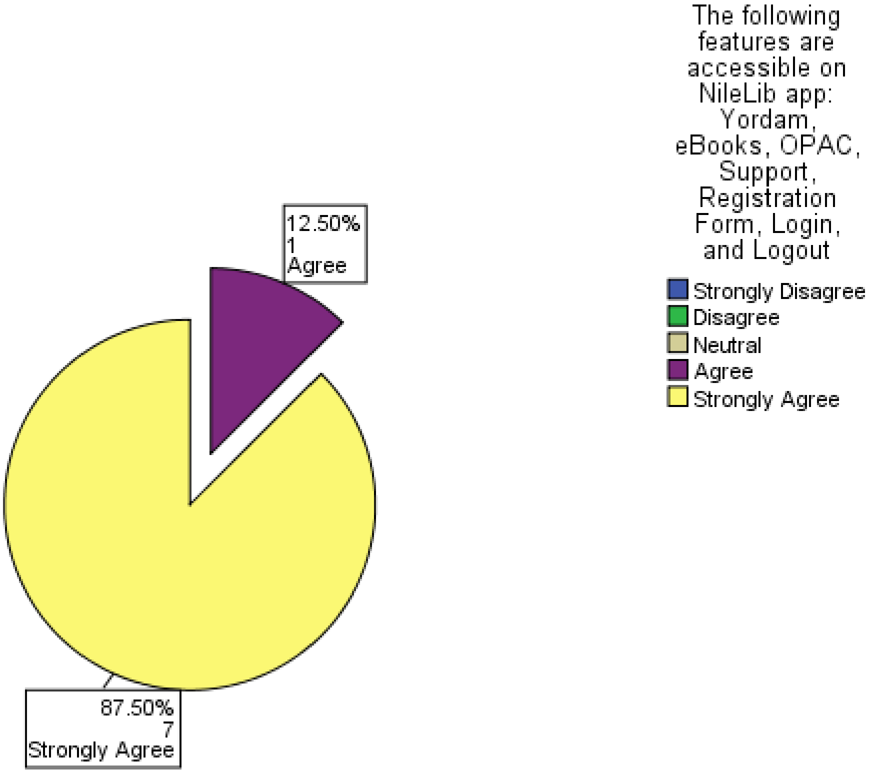


Figure 3 – Features accessible on the MLMS App

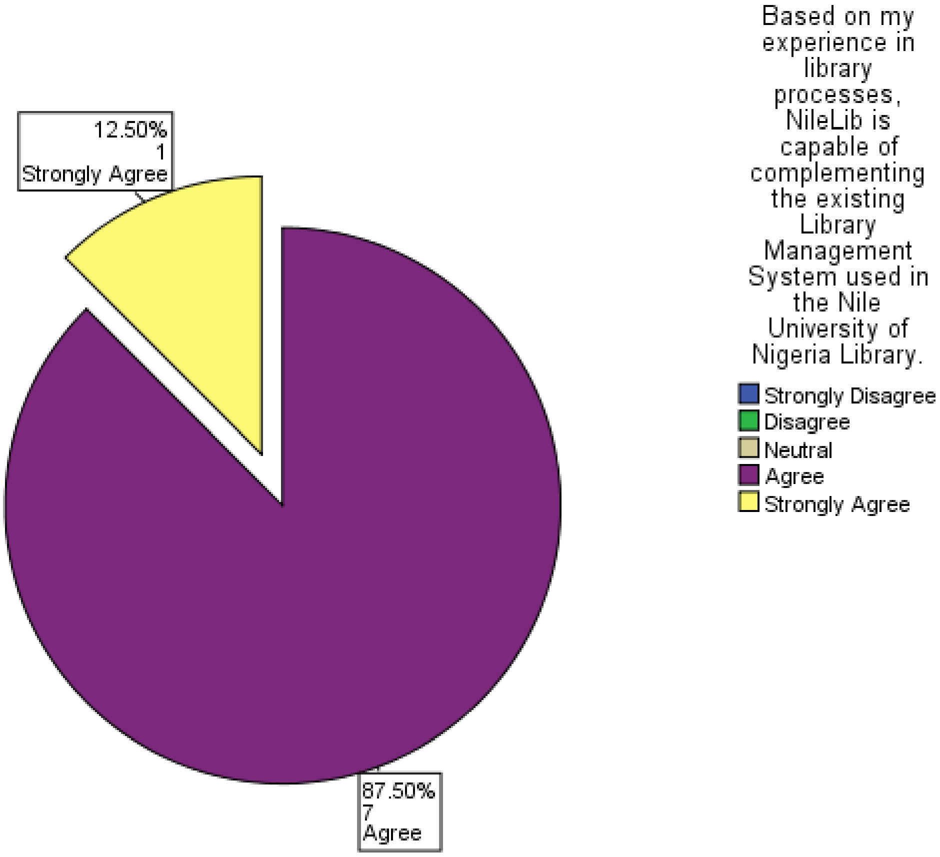


Figure 4 – MLMS is capable of complementing the existing Library Management System

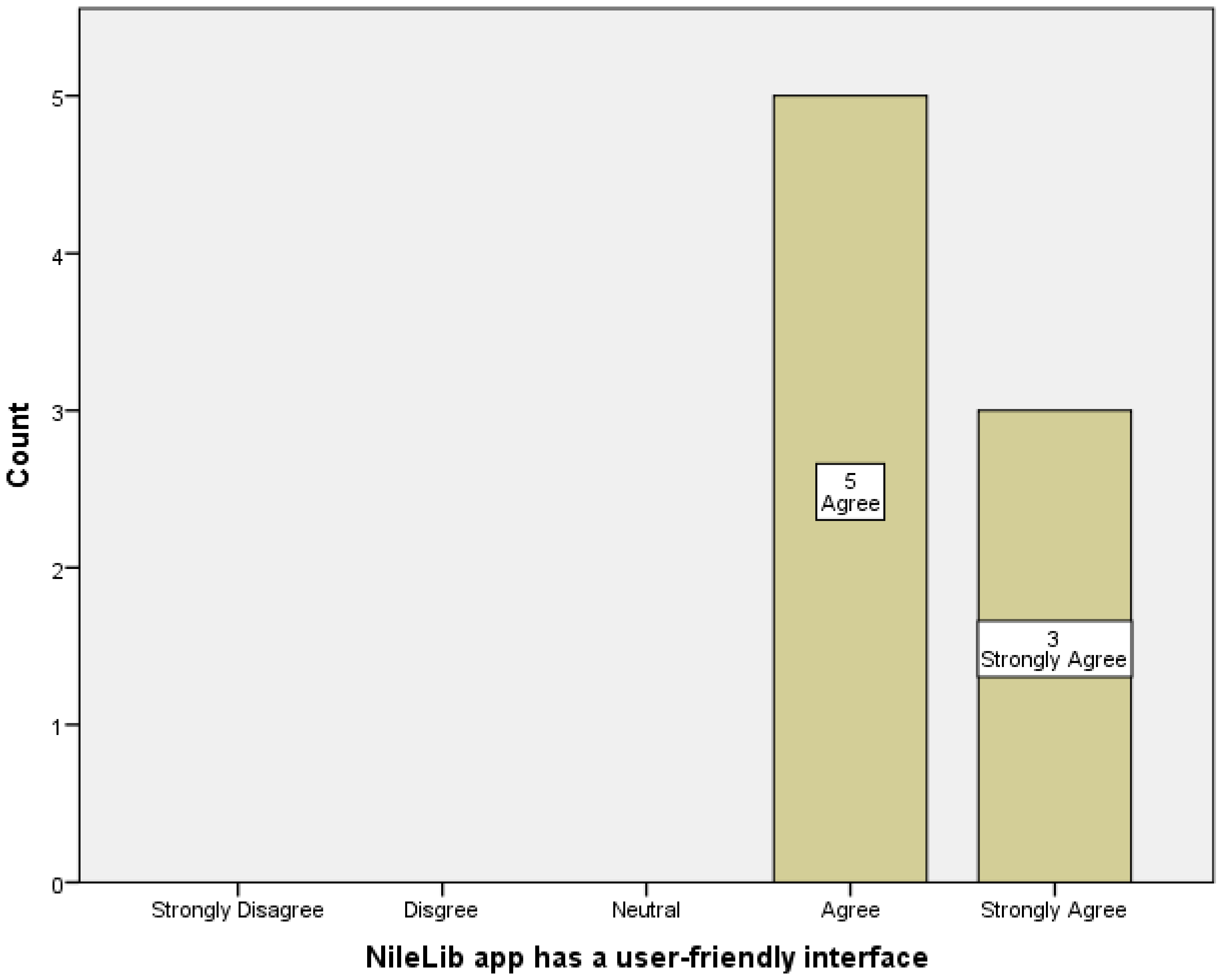


Figure 5 – MLMS App has a User-friendly interface

REFERENCES

1. Masiye, S. (2019). Design and Development of an Electronic Library Management System for Mufulira Skills Training Institute. *The International Journal of Multi-Disciplinary Research, 1189, 1-51*

2. Adeniran, T., Zubairu, J., & Onidare, S. (2018). Design and Implementation of a Secured Automated Library Using RFID. *International Journal of Information Processing and Communication, 6*(2), 334–346.

3. Rubin, R., Rubin, R., &Alire, C. (2020). *Fundamentals of Library and Information Science*. Chicago: ALA Neal-Shuman.

4. Samuel, A., Godfred, A., & He, X. (2018). Design and Implementation of Library Management System. *International Journal of Computer Applications, 182*(13), 18–25. doi: 10.5120/ijca2018917766

5. Uwaifo, S. (2010). Computer anxiety as predictor of librarians' perceived ease of use of automated library systems in Nigerian University Libraries. *Information Technologist, 6*(2), doi: 10.4314/ict.v6i2.52698

6. Nalluri, S., & Gaddam, B. (2016). Mobile Library Services and Technologies: A Study. *International Journal of Research in Library Science, 2*(2), 59–66.

7. Salihu, I. A., Ibrahim, R., & Usman, A. (2018). A Static-dynamic Approach for UI Model Generation for Mobile Applications. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. doi: 10.1109/icrito.2018.8748410

8. Fourie, I. (2011). Personal information and reference management. *Library Hi Tech, 29*(2), 387–393. doi: 10.1108/07378831111138242

9. Islam, R., & Mazumder, T. (2010). Mobile application and its global impact. *International Journal of Engineering & Technology, 10*(6), 72–78.

10. Minelli, R., & Lanza, M. (2013). Software Analytics for Mobile Applications--Insights &amp; Lessons Learned. *2013 17th European Conference on Software Maintenance and Reengineering*. doi: 10.1109/csmr.2013.24

11. Usman, A., Ibrahim, N., & Salihu, I. A. (2020). TEGDroid: Test Case Generation Approach for Android Apps Considering Context and GUI Events. *International Journal on Advanced Science, Engineering and Information Technology, 10*(1), 16. doi: 10.18517/ijaseit.10.1.10194

12. Serrano, N., Hernantes, J., & Gallardo, G. (2013). Mobile Web Apps. *IEEE Software, 30*(5), 22–27. doi: 10.1109/ms.2013.111

13. Wasserman, A. I. (2010). Software engineering issues for mobile application development. *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research - FoSER '10*. doi: 10.1145/1882362.1882443

14. Salihu, I., & Selamat, A. (2016). Improved Service Responsibility Table of Users' Requirements for E-Service System. *2nd National Research and Innovation Conference (NRICon2016) at Kuching, Sarawak, Malaysia*. Retrieved from https://www.researchgate.net/publication/316823442_Improved_Service_Responsibility_Table_of_Users'_Requirements_for_E-Service_Systems

15. Tan, X., Alter, S., & Siau, K. (2011). Using service responsibility tables to supplement UML in analyzing e-service systems. *Decision Support Systems, 51*(3), 350–360. doi: 10.1016/j.dss.2011.01.001