Redesign of Ipusnas (Indonesian E-Book Reader App) with User-Centered Design Method

by

Khairunnisa Nurrahmatia Salsabila

A Thesis

Submitted to the Faculty of Nanjing Xiaozhuang University
in Partial Fulfillment of the Requirements for the degree of

Bachelor of Software Engineering

School of Information Engineering
Fangshan, Nanjing
May 2023

To: Dean Zheng Hao
School of Information Engineering

This thesis, written by **Khairunnisa Nurrahmatia Salsabila**, and entitled **Redesign of Ipusnas (Indonesian E-Book Reader App) with User-Centered Design Method**, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

Dean Zheng Hao
School of Information Engineering

Date of Defense: May 24, 2023

The thesis of **Khairunnisa Nurrahmatia Salsabila** here is approved.

Dean Zheng Hao
School of Information Engineering

Dean Zhou Hong
Chief of Foreign Affairs Office

Nanjing Xiaozhuang University, 2023

ACKNOWLEDGMENTS

I would like to express my gratitude to Li Qing Laoshi for the invaluable guidance and support as the Java Android lecturer. The expertise and motivation have been instrumental in shaping my understanding of Android development and my growth in college. I am also deeply thankful to Wu Hai Yong Laoshi as the supervisor of my thesis and for providing constant encouragement throughout the process. His insightful feedback and unwavering support have been truly inspiring. I extend my appreciation to Ni Zhen Laoshi for the guidance and contributions to my academic journey. Additionally, I am immensely grateful to my parents and friends for their unconditional love, encouragement, and belief in my abilities. Their unwavering support has been a constant source of strength. I would also like to express my sincere appreciation to Mrs. Sophie Mou, the counselor of the Indonesian representatives, for her invaluable assistance and guidance that has played a crucial role in my achievements thus far.

TABLE OF CONTENTS

ACKNOWLEDGMENTS... 2
LIST OF TABLES... 4
LIST OF FIGURES... 5
ABSTRACT...6
CHAPTER 1. INTRODUCTION.. 1
1.1 Background.. 1
1.2 Problem Identification... 4
1.3 Problem Formulation... 5
1.4 The Limitation of Study..5
1.5 The Aims of Study..5
1.6 Benefits of Study...5
1.7 Methodology of Research.. 5
CHAPTER 2. LITERATURE REVIEW..8
2.1 The National Library..8
2.1.1 Open Access Service System..9
2.1.2 Closed Access Service System.. 9
2.2 Human Computer Interaction.. 10
2.3. User Experience (UX)..11
2.4 User Interface (UI)..11
2.5 Android Studio..14
2.6 Google Firebase Cloud Firestore... 15
CHAPTER 3. ANALYSIS & DESIGN... 16
3.1 System Planning... 16
3.2 Class Diagram...18
3.3 Database Design..19
3.3.1 User Table..19
3.3.2 LoginActivity..20
3.3.3 RegisterActivity.. 21
3.3.4 CategoryAddActivity..21
3.3.5 SplashActivity.. 21
3.3.5 DashboardAdminActivity..22
CHAPTER 4. RESULTS.. 23
4.1 Splash Screen..23
4.2 Firebase Integration Setup.. 25
4.3 Login/Register Function... 27

4.3.1 Login....27
Figure 4.3 checkUser to retrieve the data....27
Figure 4.3.1 loginUser to match the data and the data in real time database....27
Figure 4.3.2 validateData to match the data and the data in user input before....28
Figure 4.3.3 UI interface of login process....28
Figure 4.3.3 updateUserInfo....29
Figure 4.3.4 createUserAccount in firebase authentication....29
Figure 4.3.5 validateData.... 30
Figure 4.3.6 User interface from this Register process.... 31
4.4 Add Category....31
Figure 4.4.1 Initiate the Firebase authentication to store categories.... 31
Figure 4.4.2 validate the data are not empty....31
Figure 4.4.3 add the category to firebase....32
Figure 4.4.4 The user interface of Add Category....33
CHAPTER 5. SUMMARY/CONCLUSIONS....34
5.1 Conclusion....34
REFERENCES.... 34

LIST OF TABLES

Table 1. User table.... 19
Table 2. LoginActivity table.... 20
Table 3. RegisterActivity table.... 21
Table 4. CategoryAddActivity table.... 21
Table 5. SplashActivity table.... 21
Table 6   DashboardAdminActivity table.... 22

LIST OF FIGURES

Figure 1. Moving average rating from iPusnas app based on AppBrain on May 10, 2023.......2
Figure 1.2. iPusnas users review on Google Play Store.. 4
Figure 2. Outline of Human and Computer Interaction (Martono & Kurniawan, 2011).........11
Figure 3.1 Class diagram.. 18
Figure 3.2 User table integration with JSON...20
Figure 4. Initiation of Firebase authentication...23
Figure 4.1 Splash Screen navigates to MainActivity in if-else condition...............................23
Figure 4.1.2 User type check.. 24
Figure 4.1.3 Splash screen.. 24
Figure 4.2 Firebase dependencies setup on build:gradle file (bookapp project).................... 25
Figure 4.2.1 Initiate the app name in build:gradle (app) file... 26
Figure 4.2.2 Database project profile... 27
Figure 4.3 checkUser to retrieve the data...27
Figure 4.3.1 loginUser to match the data and the data in real time database........................... 27
Figure 4.3.2 validateData to match the data and the data in user input before........................28
Figure 4.3.3 UI interface of login process...28
Figure 4.3.3 updateUserInfo...29
Figure 4.3.4 createUserAccount in firebase authentication..29
Figure 4.3.5 validateData..30
Figure 4.3.6 User interface from this Register process.. 31
Figure 4.4.1 Initiate the Firebase authentication to store categories...................................... 31
Figure 4.4.2 validate the data are not empty...31
Figure 4.4.3 add the category to firebase..32
Figure 4.4.4 The user interface of Add Category.. 33

ABSTRACT

Recognizing the importance of literacy as a fundamental human resources right, Indonesia has launched the National Literacy Movement to accelerate human resource development and cultivate a culture of literacy. As Indonesia prepares for the Golden Generation of 2045, which marks the country's 100th year, the nation acknowledges the critical role of human resources in sustaining its progress. However, population expansion, driven by a high birth rate during the pandemic, presents both opportunities and challenges. Improper utilization of this demographic shift could lead to socio-economic issues such as poverty, poor health, unemployment, and increased crime rates. Acknowledging the lack of public interest in reading in the digital era, the government has responded by leveraging technology. With a considerable number of smartphone users and the introduction of the iPusnas digital library application, Indonesia aims to promote reading and improve access to books. However, user reviews have highlighted concerns regarding the application's usability, including difficulties in searching for specific books and issues with the sign-up process. To address these concerns and improve the user experience, the author proposes a research project focusing on the redesign of the iPusnas android mobile application to ensure that user needs and feedback influence the development process.

CHAPTER 1. INTRODUCTION

1.1 Background

Since citizens of the first state are a member of a rapidly changing global community, literacy is seen by its residents as one of the fundamental rights that must be supported by the state. According to CCSU (2016), Indonesia ranks 60th out of 61 countries in terms of literacy interest. Indonesia is one of the third countries in the world to promote literacy by launching the National Literacy Movement[1] to speed up the development of human resources with a culture of literacy that is still relatively low in Indonesia. As Indonesia prepares young talent to welcome the Golden Generation of 2045 in 2021, human resources are a crucial component in sustaining a nation. The historical momentum initiative is called the Golden Generation.

Golden Generation of 2045 is the historic momentum program to mark Indonesia's 100th year or one century. Population expansion is inescapable due to the pandemic's massive birth rate in 2020 and 2021. Indonesia will benefit from a demographic shift in 2045 when 70% of the country's population—those aged 15 to 64—are productive, compared to 30% of those aged under 14 and over 65—who are unproductive. Negative effects will result from improper use of this demographic shift, particularly socio-economic issues including poverty, poor health, unemployment, and high crime rates. The government is prepared for the lack of public interest in reading, which results in poor human resources in the digital era, based on the facts that Indonesia will have to deal with and current technology trends.

People are accustomed to technology in the contemporary digital era, particularly smartphones. With 27 million smartphone users, Indonesia is the 11th most smartphone-dependent nation in the world. Smartphone penetration has increased in Indonesia as a result of the country's expanding middle class and rising per capita income, particularly as cellphones are the primary device used to access the internet..

According to research [5], the percentage of Indonesians who use cellphones for social networking and chat is expected to quadruple from 16% in 2014 to 32% in 2015. A digital library application called iPusnas was made available on August 16, 2016, by the Indonesian National Library Agency (Perpusnas), in partnership with Aksamaraya, in response to the country's rapidly expanding smartphone market.

iPusnas is a national digital library that has a collection of 73,302 book titles and a collection of 891,397 copies of ebooks that can be downloaded and installed on the Playstore page, the AppStore for smartphones/tablets or the ipusnas.id web for computers. To register for iPusnas, we do not have to be registered as members of the National Library. Users merely need to register with their own email or Facebook accounts to be able to borrow collections at iPusnas. Users are then immediately presented with a varied selection of iPusnas reading books after that.

Based on the AppBrain website which analyzes application traffic in the Google Play Store and iOS App Store, the iPusnas application has a moving average rating of 3,677 on May 10, 2023. This rating has increased by 0.287 compared to February 2, 2020.

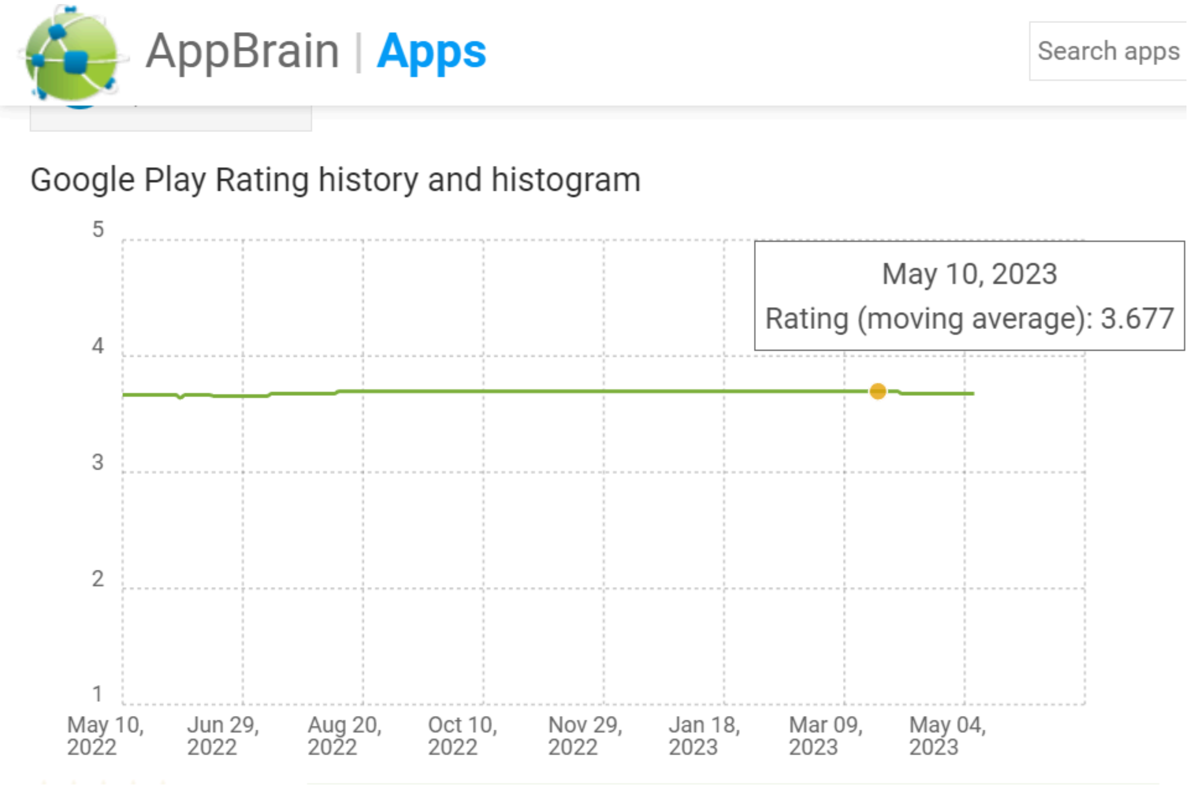


Figure 1. Moving average rating from iPusnas app based on AppBrain on May 10, 2023

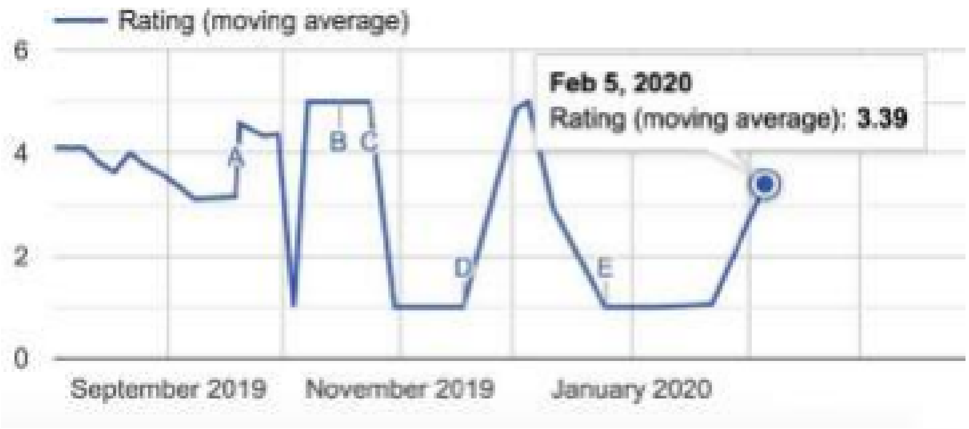
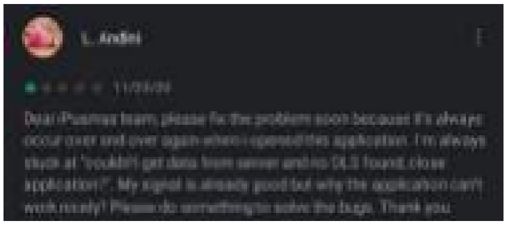


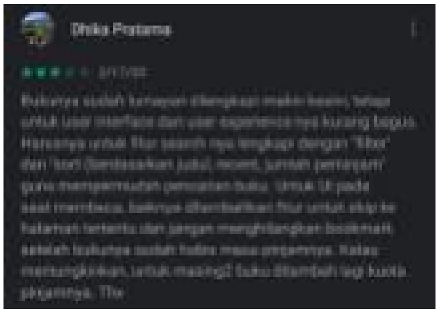
Figure 1.1 Moving average rating from iPusnas app based on AppBrain on Feb 5, 2020

Through a review of the iPusnas application on the Play Store there were also many users who criticized the appearance of the iPusnas application not meeting their needs, such as users finding it difficult to find the book they wanted because there was no sort and filter feature in the search feature, then the sign up button was not found on the sign up page , and it's hard to find the right book title.


L. Andini

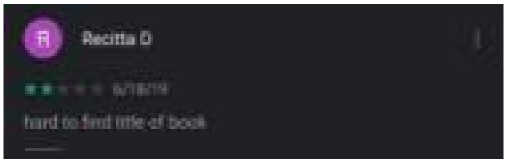
★☆☆☆☆ 11/23/20

Dear iPusnas team, please fix the problem soon because it's always occur over and over again when i opened this application. I'm always stuck at "couldn't get data from server and no DLS found, close application?". My signal is already good but why the application can't work nicely! Please do something to solve the bugs. Thank you


Dhika Pratama

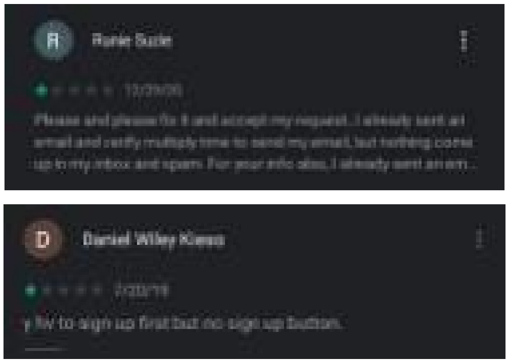
★★★☆☆ 2/17/20

Bukunya sudah lumayan dilengkapi makin keren, tetapi untuk user interface dan user experience nya kurang bagus. Harusnya untuk fitur search nya lengkapi dengan "filter" dan "sort (berdasarkan judul, recent, jumlah peminjam)" guna mempermudah pencarian buku. Untuk UI pada saat membaca, baiknya ditambahkan fitur untuk skip ke halaman tertentu dan jangan menghilangkan bookmark setelah bukunya sudah habis masa pinjamnya. Kalau memungkinkan, untuk masing2 buku ditambah lagi kuota pinjamnya. Thx


Recitta D

★★☆☆☆ 6/18/19

hard to find title of book



Ranie Suzie

12/29/20

Please and please fix it and accept my request.. I already sent an email and verify multiply time to send my email, but nothing come up to my inbox and spam. For your info also, I already sent an em...

Daniel Wiley Kiesas

2/22/19

y fw to sign up first but no sign up button.

Figure 1.2. iPusnas users review on Google Play Store

UX factors are things that need to be considered in building an application. The UX of an application is influenced by the UI because the UI is a link between application users in interacting. Building a good UI for users can use the UI and UX model designer method, namely User-Centered Design (UCD). UCD itself is a multidisciplinary design approach based on the active involvement of users to improve understanding of user needs and tasks, design iteration and evaluation. The stages contained in the user's UCD are the main components in system development because users are directly involved in system development and users have the right to provide suggestions and input that influence system development.

The problems described above, it is necessary to redesign the UI of the iPusnas application that have a major impact on system development. Therefore, the author are interested in conducting research on redesign in the form of an Android mobile based application based on User-Centered Design analysis on the iPusnas application with the title "Redesign of Ipusnas (Indonesian E-Book Reader App) with User-Centered Design Method".

1.2 Problem Identification

Based on the background previously described, the following problems can be identified in this study:

1. iPusnas application still has many shortcomings such as the difficulty to find the sign up and login button, lack of search suggestion keywords.
2. It is necessary to increase the usability value for aspects of the effectiveness and efficiency of the iPusnas application in order to better understand user needs.

1.3 Problem Formulation

Based on the identification of the problems previously described, the problem in this research can be formulated as "How to implement the design of a user interface based on user experience analysis in the iPusnas application using the user-centered design method to suit user needs?"

1.4 The Limitation of Study

The limitations of this study are as follows:
1. The object of this research is the iPusnas Mobile application which operates on Android.
2. Respondents for this research are users of the iPusnas application. Respondents to the questionnaire amounted to 50 people
3. The results of this study is to redesign and to fix the sign up button that does not appear on the app, based on the users' review on Google Play Store.

1.5 The Aims of Study

1. The aims of this research are as follows:
2. Designing the iPusnas application using the User Centered-Design (UCD) method and producing a redesign of an Android-based mobile application using Firebase Cloud Firestore

1.6 Benefits of Study

The benefits obtained in this study are as follows:

1. Proposed redesign of the iPusnas application in this study so that it can meet user needs and provide the information needed by users.

2. For the National Library team and application development companies, it can be used as a reference in updating the iPusnas application in the next version.

1.7 Methodology of Research

The method in this study was divided into 2 stages, namely the data collection method and the User Centered-Design (UCD) method as the UI and UX design method for the iPusnas application

1.7.1 Methods of Collecting Data

1. Library Studies

Literature study is studying various reference books and the results of similar previous studies which are useful for obtaining a theoretical basis for the problem to be studied [3].

2. Observation

Observation is an observation and also a sequential recording consisting of elements that appear in a phenomenon on the research object. And the results of these observations will then be reported in a systematic manner and also in accordance with the applicable rules [6].

3. Interview

An interview is a meeting of two people to exchange information and ideas through question and answer, so that the meaning of a particular topic is usually contrasted [2].

1.7.2 User-Centered Design (UCD) Method

The process of the User-Centered Design (UCD) method consists of four processes [7], namely:

1. Specify the context of use

Identify system user context. The system user context contains the user who will use the system and determines the conditions in which the user uses the system. User context data is obtained from fetching data to users. The results obtained from this stage are in the form of user characteristics, tasks to be carried out by the user and the environment that shows through what and where the user will use the system. The results at this stage are then used as system user context data.

2. Specify user and organizational requirements

Identify the needs of the system that can match the needs of the user, the data on the needs of the user is obtained from the user and the organization and has to do with the system user context data. The results obtained from this stage are in the form of data from the system according to user needs. The results at this stage are then used as data needs from the user for the system to be built.

3. Product design solutions

Designing design solutions based on data needs from users, this process is carried out in several processes, starting from designing the design concept, making the design more detailed and concrete then implementing the design. The results obtained from this stage are in the form of a system design concept design, system design in a more detailed form and implementation of system design. The results at this stage are then used as a system design solution.

4. Evaluate designs against user requirements

Testing the system design solution to assess whether the design solution can match the user's needs. The results obtained from this stage are in the form of test data. The results obtained from this stage are in the form of test data for system design solutions according to the needs of users and feedback for design improvements.

1.8 Systematics of Writing

In compiling this research report, the discussion is divided into five chapters which will be briefly described as follows:

CHAPTER I. INTRODUCTION

This chapter describes the background of the problem, problem identification, problem formulation, research limitations, research objectives, research benefits, research methodology and writing systematics.

CHAPTER II. LITERATURE REVIEW

This chapter describes the theories related to problem solving in this study.

CHAPTER III. ANALYSIS & DESIGN

This chapter describes the general description of the system analysis and database design.

CHAPTER IV. RESULTS AND DISCUSSION

This chapter contains the discussion and results of the analysis and redesign process in the iPusnas android mobile-based application using the UCD method.

CHAPTER V. CONCLUSIONS AND REFERENCES

This chapter contains conclusions and references on the results of research that has been done.

CHAPTER 2. LITERATURE REVIEW

2.1 The National Library

Based on article 4 of Law No. 43 of 2007 concerning libraries, it is stated that libraries aim to provide services to users, increase their interest in reading, and broaden their horizons and knowledge to educate the nation's life [14].

In the results of the Conference of Directors of National Libraries (CDNL) meeting in Bangkok in 1999 the National Library is an institution, primarily funded (directly or indirectly) by the state, which is responsible for comprehensive collection, bibliographical recording, preservation and provision of documentary heritage (in particular published material of all kinds) originating in or relating to that country; and may also be responsible for the efficient and effective furthering of library functions in the country through tasks such as nationally meaningful collection management, provision of infrastructure, coordination of library activities and information systems in the country concerned, international relations, and exercising leadership. Usually this responsibility is formally recognized, usually based on legislation. For the purposes of this definition, a state is defined as an independent sovereign state.

Institutions equivalent to national libraries exist in non-sovereign national entities such as in Catalonia, Quebec and Wales [9]. Article 3 of Law No. 43 2007 states that libraries function as a vehicle for education, research, preservation, information and recreation to enhance the intelligence and empowerment of the nation. The function of education is manifested by a library that is able to increase its users' love of reading. The research function is implemented by providing services for users in obtaining information as reference material for research purposes. The function of preservation is as a place to preserve library materials (library materials are a source of science, technology and culture). The information function is implemented by providing complete and quality library sources. The recreational function is implemented by providing entertainment books and recreational layout. In addition to these functions, there is also a social function, which is defined as a forum for outreach among visitors in obtaining information [14].

In the Draft Government Regulation of the Republic of Indonesia Year 2009 Concerning Library National Standards [17], it is stated that libraries based on type and ownership include:

1. National library
2. Government library
3. Provincial library
4. District/city libraries
5. District library
6. Village/kelurahan library
7. School/madrasa library
8. College library
9. Special library
10. Family library
11. Personal library

There are two kinds of service systems that are usually carried out by libraries, namely open service systems and closed service systems. Each of these systems has advantages and disadvantages.

2.1.1 Open Access Service System

In the open access service system, the library gives users the freedom to enter and choose the collection they want from the shelves. The officer only notes if the collection will be borrowed and returned.

1. Pros:

a. Users are free to choose their own books
b. This freedom creates a stimulus to read
c. If the desired book is not there, you can choose another book with the same subject or topic.

2. Cons:

a. Arrangement of books on shelves becomes difficult to order.
b. It is possible that many books were lost.

2.1.2 Closed Access Service System

In contrast to the open system, visitors are not allowed to enter the collection room, but what their needs must be collected by the officer. Collection browsing/search must go through the catalog. In addition to recording borrowing and returning, the clerk also took and returned the collection to the shelf.

1. Pros:

a. The order and location of the books are maintained

b. There is no need for special officers to supervise users.

2. Cons:

a. There is no freedom to see books, you have to look through catalogs

a. Looking through the catalog is sometimes frustrating, because in the catalog there is, but the books are often not there, and you have to select again over and over again.

b. Officers must pick up and return books

c. Catalog must be complete.

2.2 Human Computer Interaction

Human and computer interaction or human computer interaction (HCI) is a relationship between humans and computers that have certain characteristics to achieve a goal by running a system masked by a user interface [15]. The main role of human and computer interaction is to produce a system that is easy to use, safe, effective and efficient [12]. There are three components in human and computer interaction, including system users, interaction models and the system to be used. To see an outline of human and computer interaction can be seen in Figure 2.1.

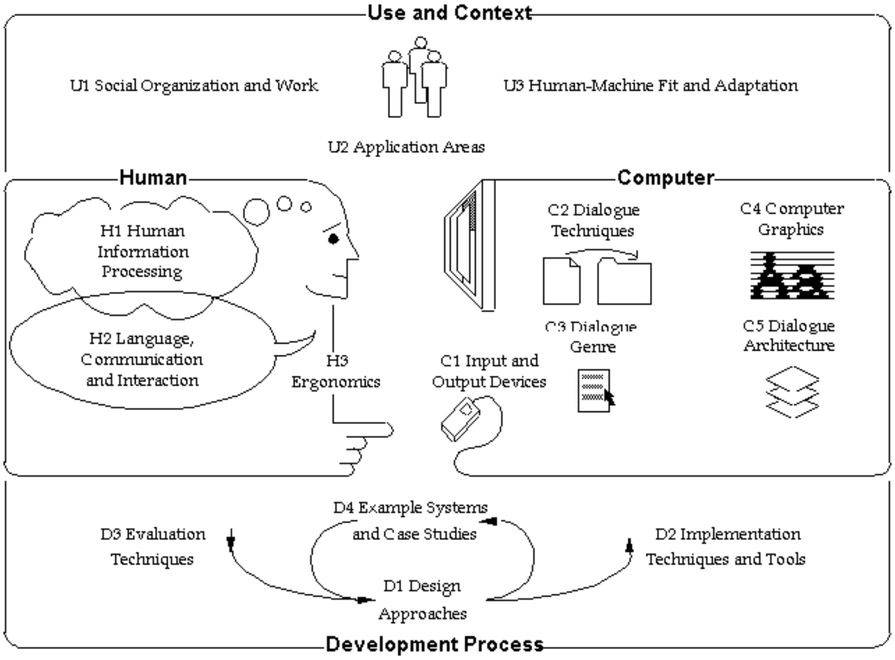

Use and Context
U1 Social Organization and Work
U3 Human-Machine Fit and Adaptation
U2 Application Areas
Human
H1 Human Information Processing
H2 Language, Communication and Interaction
H3 Ergonomics
Computer
C2 Dialogue Techniques
C4 Computer Graphics
C3 Dialogue Genre
C5 Dialogue Architecture
C1 Input and Output Devices
D4 Example Systems and Case Studies
D3 Evaluation Techniques
D2 Implementation Techniques and Tools
D1 Design Approaches
Development Process

Figure 2. Outline of Human and Computer Interaction (Martono & Kurniawan, 2011)

The explanation of HCI previously can be concluded that human and computer interaction is a relationship between humans and computers to create a system that is easy to use, effective, efficient and safe by using an interface.

2.3. User Experience (UX)

User experience (UX) is a person's perception and response resulting from the use or anticipation of a product, system or service, including all emotions, beliefs, choices, perceptions, physical and psychological responses, behaviors, and user successes that occurred before, during, and after system use [8].

2.4 User Interface (UI)

User Interface (UI) is a term used to describe the appearance of a machine or computer that interacts directly with the user. The design and arrangement of the interface display needs attention to produce a good display [11].

The goal of UI is to design an effective interface for software systems [10]. Effective means ready to use, and the results are as needed. Requirements here are the needs of its users. Users often judge the system from the interface, not from its function but from the user interface, if the design of a product's user interface is bad, then this is often a reason not to use the product. In addition, a bad interface causes users to make fatal mistakes. Design must be user-centered, meaning that users are very involved in the design process. Because of that there is an evaluation process carried out by the user against the design results.

Creating a user interface, designers need to pay attention to the principles that will be used as a reference. This principle is the rule of making a product design to suit user needs. In the book Shneiderman explains the Eight Golden Rules of Interface Design that need to be considered so that the interface design of a product that we make can be used properly and according to user needs.

In the book Shneiderman explains the Eight Golden Rules of Interface Design that need to be considered so that the interface design of a product that we make can be used properly and according to user needs.

1. Strive for consistency

Consistency in a product is needed between related pages. This is useful so that the user feels that the page to be accessed still has a connection with the product being run, thereby making the user feel comfortable in using or exploring the product that we make without having to move to another product.

2. Enable frequent users to use shortcuts

As a UI designer, you need to take into account the various types of users who will use their products, such as type of job, age, culture, language and the user's level of understanding of a product. This will be a reference for designers to design products that users can understand. As there is a certain button that is commonly used in other applications with layouts and functions that are familiar to users, this should be a reference for designers to design an application.

3. Offer informative feedback

Providing information about what the user has done or is doing requires feedback that can be seen visually and has good meaning so that the user gets information or answers from the actions taken by the user. A simple example such as the user clicking on an application menu can respond to user actions by displaying the loading process.

4. Design dialogs to yield closure

Conveys that the process carried out by the user has been completed so that the user understands without having to wait for uncertainty whether to carry out another process after carrying out a process. As a simple example, the user saves changes to personal data in the settings menu, as feedback for the actions taken by the application user can display a notification that changes to personal data have been successfully updated.

5. Offer simple error handling

The user will feel bored if the product used often experiences errors or errors and causes the user not to want to use the product again. To avoid this, the designer can make a brief guide to the process that the user will carry out.

6. Permit easy reversal of actions

With regards to UX, users may make mistakes when using an application, and fixing these errors should be easy for users to make and access. For example, in the search box in the application the user inputs the wrong word he actually wants, the user can easily replace it by deleting it or directly clicking the 'x' button in the search box which is interpreted as deleting quickly and the user will re-enter the word he wants correctly. This will make the user feel comfortable with the application he is using.

7. Support internal locus of control

Users will feel comfortable when they can own or adjust the appearance of the application they are using according to their taste. A simple example is that users can easily change the appearance of the application from light mode to dark mode or vice versa. This can increase user satisfaction with applications that greatly affect the UX of these applications.

8. Reduce short-term memory of load

When designing the appearance of a product, designers will usually focus more on the layout design of menus and buttons. This is less effective and would be better if applied to the process when the user has to provide input to the system. By implementing this, the user does not need to remember the data that must be input into the system, because the data that is input has already been provided by the system.

2.5 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for developing Android apps [16] . It is built on the IntelliJ IDEA code editor and provides additional features to improve productivity. Some notable features include:

1. Gradle-based build system: Android Studio uses a flexible build system called Gradle, which simplifies the process of building and managing Android app projects.
2. Emulator: Android Studio offers a fast and feature-rich emulator that allows developers to test their apps on virtual Android devices.
3. Unified environment: Developers can use Android Studio to develop apps for various Android devices, providing a unified platform for development.
4. Live Edit: With Live Edit, developers can update composables in emulators and physical devices in real-time, making it easier to see changes and iterate quickly.
5. Code templates and GitHub integration: Android Studio provides pre-built code templates and integration with GitHub to assist developers in building common app features and importing sample code.
6. Testing tools: Android Studio offers extensive testing tools and frameworks to facilitate efficient app testing and debugging.
7. Lint tools: Lint tools in Android Studio help identify performance, usability, version compatibility, and other issues in the codebase.
8. C++ and NDK support: Android Studio has built-in support for C++ and the Native Development Kit (NDK), enabling developers to write and optimize native code for their apps.
9. Google Cloud Platform integration: Android Studio seamlessly integrates with the Google Cloud Platform, making it easy to incorporate Google Cloud Messaging and App Engine services into Android apps.

The project structure in Android Studio consists of modules, which can be Android app modules, library modules, or Google App Engine modules. The Android project view organizes project files by modules, providing quick access to key source and build files.

2.6 Google Firebase Cloud Firestore

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps [13]. Tools are available from Firebase for monitoring analytics, reporting and resolving app errors, as well as developing marketing and product experiments.

The services provided by Firebase are the following:

1. Analytics. Google Analytics for Firebase provides free, limitless reporting on up to 500 different events. Analytics provides user activity data for iOS and Android apps, enabling better performance and app marketing decisions.
2. Authentication. Firebase Authentication makes it simple for developers to build secure authentication systems while also improving user sign-in and onboarding. With support for email and password accounts, phone auth, Google, Facebook, GitHub, Twitter login, and more, this feature provides a comprehensive identity solution.
3. Cloud messaging. Firebase Cloud Messaging (FCM) is a free, cross-platform messaging application that enables businesses to transmit and receive messages on iOS, Android, and the web with reliability. Data may be stored and synced in real time across users thanks to the Firebase Realtime Database, a NoSQL database hosted in the cloud. When an app is offline, the data is still accessible because it is continuously synced across all clients.
4. Crashlytics. Firebase Crashlytics is a real-time crash reporter that aids programmers in identifying, prioritizing, and resolving stability problems that degrade the performance of their apps. With crashlytics, developers can focus more on creating features for their apps rather than managing and troubleshooting crashes.
5. Performance. The Firebase Performance Monitoring tool offers developers insight into the iOS and Android performance characteristics of their apps, enabling them to pinpoint where and when their apps' performance may be enhanced.
6. Test lab. Firebase A cloud-based app testing infrastructure is called Test Lab. Developers may test their iOS or Android apps on a multitude of hardware combinations with a single process. In the Firebase console, they may view the outcomes, including videos, screenshots, and logs

CHAPTER 3. ANALYSIS & DESIGN

3.1 System Planning

For system planning, the author attempts to understand and identify the issues raised in iPusnas reviews on Google Play Store, then creates the program's aims and analyzes existing issues. The following phases of system planning:

1. System analysis begins with problem identification. At this point, the problem that needs to be solved comes from spreading questionnaires for 56 respondents, leading the author to conclude several datas related to the problem.
2. Through modeling, the design step tries to comprehend the problem solving discovered at the problem identification stage. After the data has been gathered, the software is designed using user flow diagrams and class diagram.
3. The implementation step is where the design that was created at the design stage is converted into a programming language that creates an application.

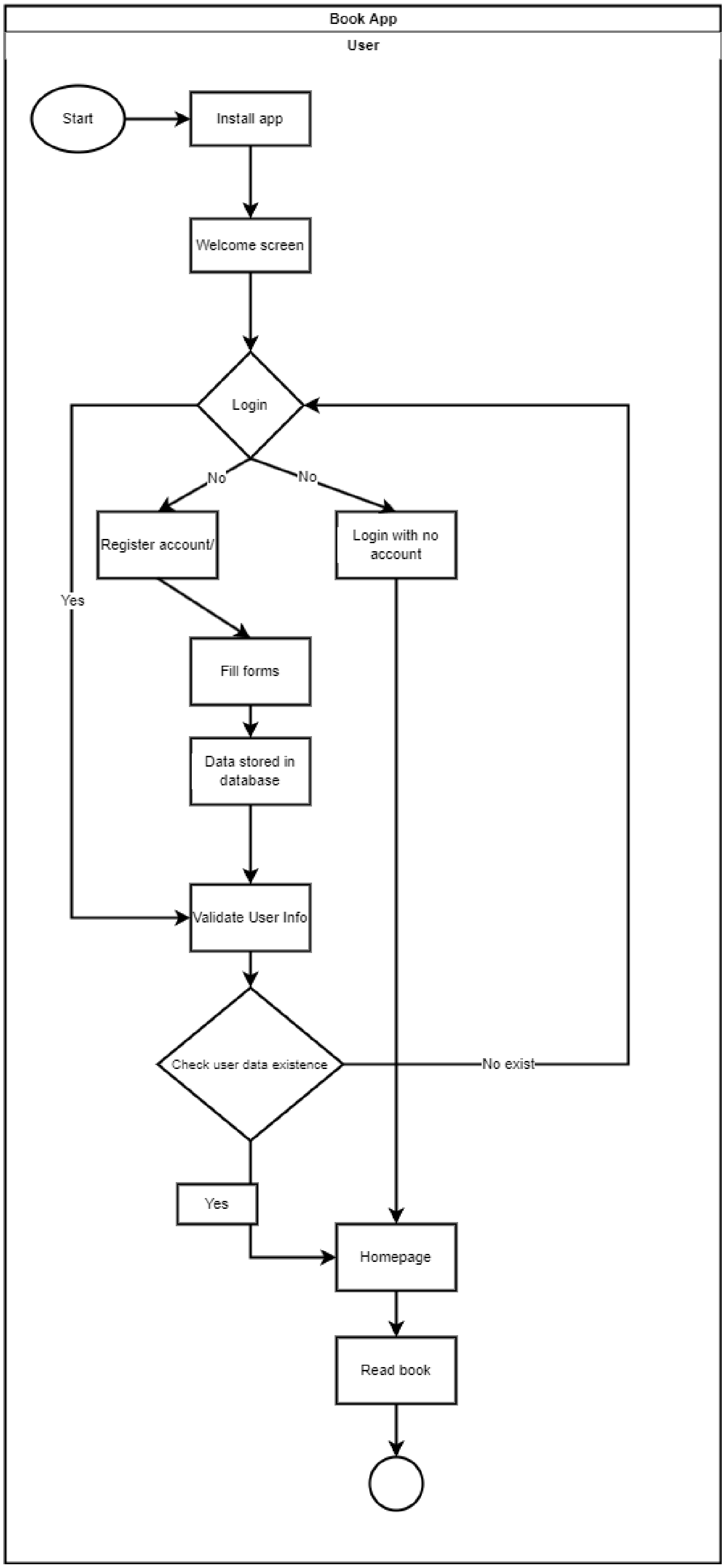


Figure 3. User flow diagram

The user flow diagram starts with installing and opening the application that leads to the welcome screen, and faces three different login methods such as login with an account that already exists before, login with no account, and register first then login.

If the users already have their account before, the next flow is Validate User Data to check user data in the database. If the user wants to login without an account, the next flow is Homepage. If the user does not have their account before, the user needs to fill the form first to store the data in the database, then validate the user data to check the data in the database. If the user data exists, then continue to the Homepage and Read Book. If the user data does not exist, the flow will be back to the Login process.

3.2 Class Diagram

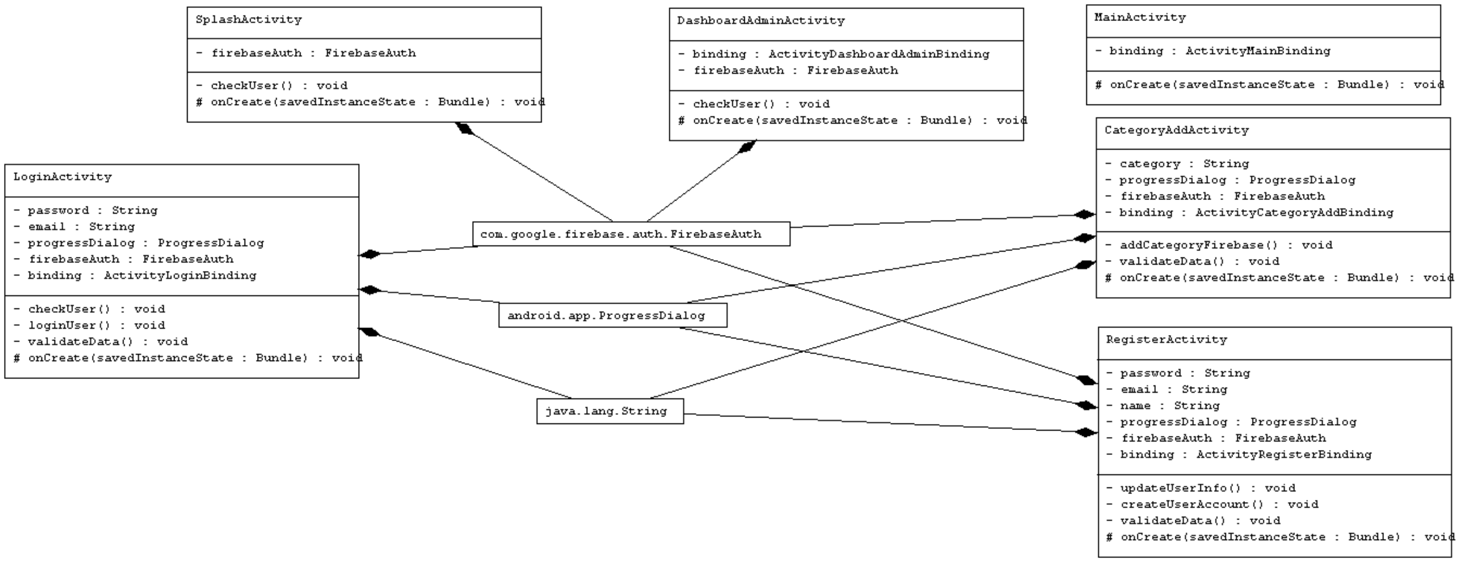

SplashActivity
- firebaseAuth : FirebaseAuth
- checkUser() : void
# onCreate(savedInstanceState : Bundle) : void
DashboardAdminActivity
- binding : ActivityDashboardAdminBinding
- firebaseAuth : FirebaseAuth
- checkUser() : void
# onCreate(savedInstanceState : Bundle) : void
MainActivity
- binding : ActivityMainBinding
# onCreate(savedInstanceState : Bundle) : void
CategoryAddActivity
- category : String
- progressDialog : ProgressDialog
- firebaseAuth : FirebaseAuth
- binding : ActivityCategoryAddBinding
- addCategoryFirebase() : void
- validateData() : void
# onCreate(savedInstanceState : Bundle) : void
LoginActivity
- password : String
- email : String
- progressDialog : ProgressDialog
- firebaseAuth : FirebaseAuth
- binding : ActivityLoginBinding
- checkUser() : void
- loginUser() : void
- validateData() : void
# onCreate(savedInstanceState : Bundle) : void
com.google.firebase.auth.FirebaseAuth
android.app.ProgressDialog
java.lang.String
RegisterActivity
- password : String
- email : String
- name : String
- progressDialog : ProgressDialog
- firebaseAuth : FirebaseAuth
- binding : ActivityRegisterBinding
- updateUserInfo() : void
- createUserAccount() : void
- validateData() : void
# onCreate(savedInstanceState : Bundle) : void

Figure 3.1 Class diagram

In the Unified Modeling Language (UML), a class diagram is a form of static structural diagram that symbolizes the structure of a system by displaying the classes, their attributes, operations (or methods), and the relationships between objects [18]. According to [18], the relationship between Login Activity, CategoryAddActivity, and RegisterActivity class is a composition relationship with java.lang.String class. This type is the special type of aggregation where parts cannot stand by themselves and will be destroyed when the whole is destroyed. In these three classes, the java.lang.String class is important because String is needed in every class.

In the diagram above, there are two other composition relationships. The first one is android.app.ProgressDialog with LoginActivity, RegisterActivity, and CategoryAddActivity. The second one is com.google.firebase.auth.FirebaseAuth with LoginActivity, SplashActivity, DashboardMainActivity, CategoryAddActivity, and RegisterActivity. This two groups' composition relationship depends on each other.

3.3 Database Design

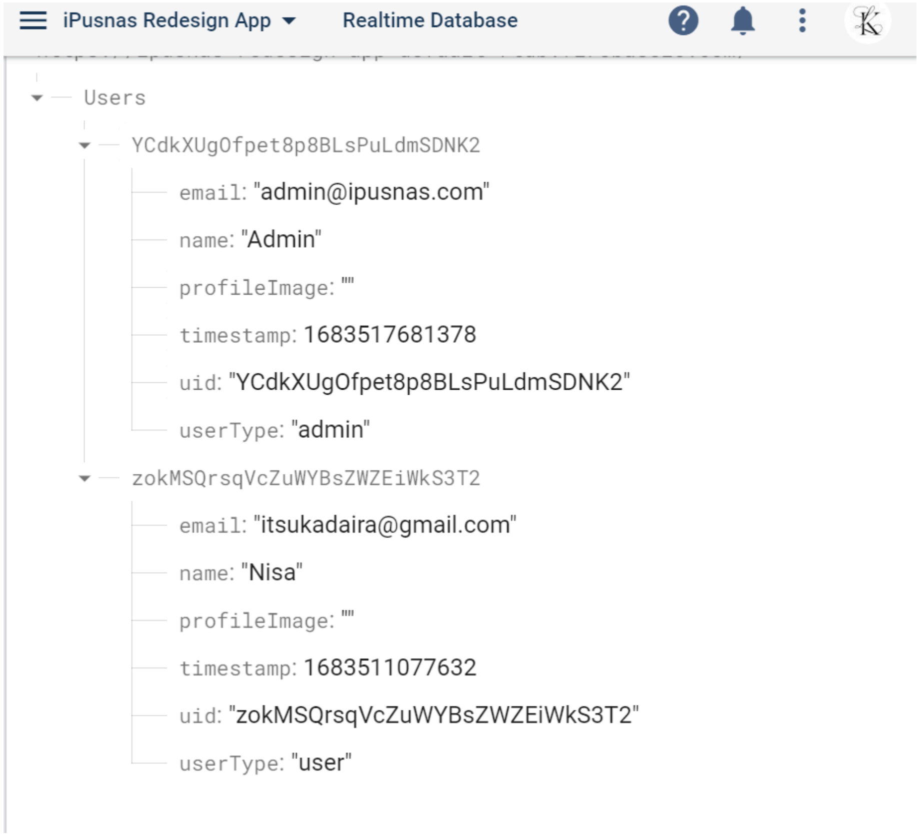
To specify how your data should be organized and when data can be read from or written to, the Realtime Database offers an expression-based rules language called Firebase Realtime Database. Developers can choose who has access to what data and how they can access it when Firebase Authentication is integrated.

Since the Realtime Database is a NoSQL database, it offers different capabilities and optimizations than a relational database. Data is stored as JSON and synchronized in realtime to every connected client. Only operations that may be finished fast are permitted by the Realtime Database API.

3.3.1 User Table

Table 1. User table

Field name	Data type
email	text
password	text
name	text
phone	phone
UID	integer
userType	text



```
Users
  YCdkXUgOfpet8p8BLsPuLdmSDNK2
    email: "admin@ipusnas.com"
    name: "Admin"
    profileImage: ""
    timestamp: 1683517681378
    uid: "YCdkXUgOfpet8p8BLsPuLdmSDNK2"
    userType: "admin"
  zokMSQrsqVcZuWYBsZWZEiWkS3T2
    email: "itsukadaira@gmail.com"
    name: "Nisa"
    profileImage: ""
    timestamp: 1683511077632
    uid: "zokMSQrsqVcZuWYBsZWZEiWkS3T2"
    userType: "user"
```

Figure 3.2 User table integration with JSON

3.3.2 LoginActivity

Table 2. LoginActivity table

Field name	Data type
password	text
email	text
progressDialog	ProgressDialog
firebaseAuth	FirebaseAuth
binding	ActivityLoginBinding

3.3.3 RegisterActivity

Table 3. RegisterActivity table

Field name	Data type
password	text
email	text
name	text
progressDialog	ProgressDialog
firebaseAuth	FirebaseAuth
binding	ActivityLoginBinding

3.3.4 CategoryAddActivity

Table 4. CategoryAddActivity table

Field name	Data type
category	text
progressDialog	ProgressDialog
firebaseAuth	FirebaseAuth
binding	ActivityCategoryAddBinding

3.3.5 SplashActivity

Table 5. SplashActivity table

Field name	Data type
firebaseAuth	FirebaseAuth

3.3.5 DashboardAdminActivity

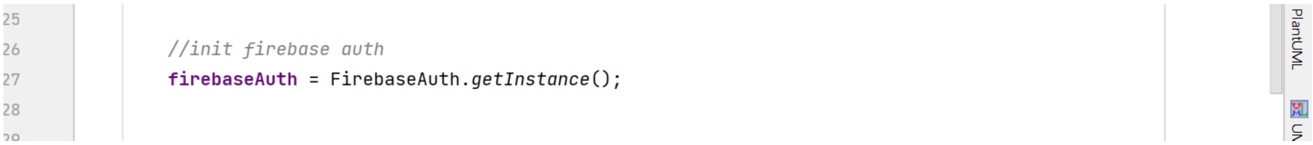
Table 6 DashboardAdminActivity table

Field name	Data type
firebaseAuth	FirebaseAuth
binding	ActivityDashboardAdminAddBinding

CHAPTER 4. RESULTS

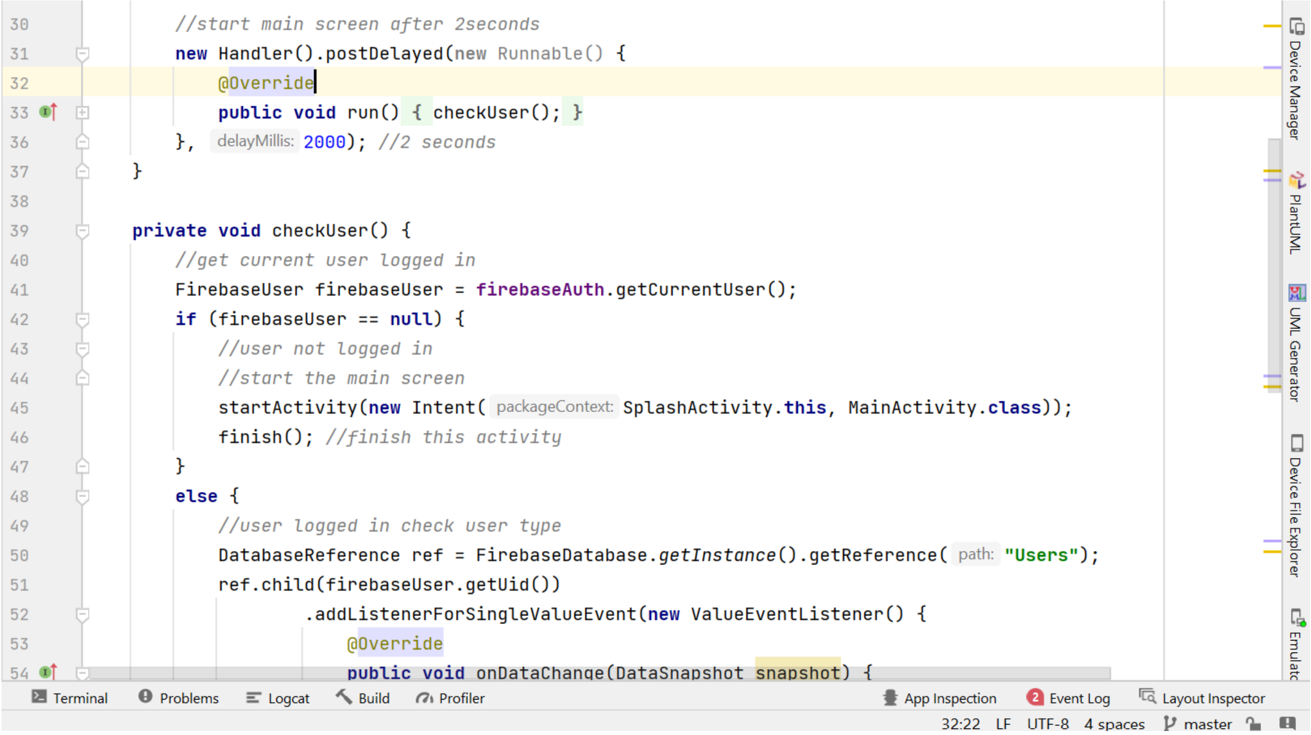
4.1 Splash Screen

In this function, Splash Screen is used as the transition while opening the app as 2000ms (2 seconds) are consumed for running the application. The Splash Screen also navigates to MainActivity while doing the checkUser() void to check whether the user type is admin or user. This process also integrated with Firebase Authentication.



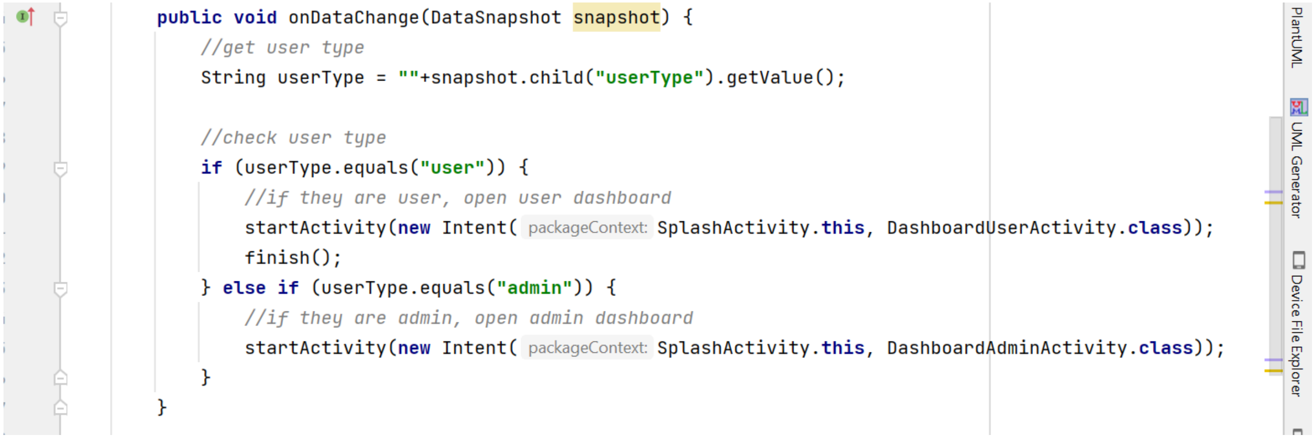
```
        //init firebase auth
        firebaseAuth = FirebaseAuth.getInstance();
```

Figure 4. Initiation of Firebase authentication



```
        //start main screen after 2seconds
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() { checkUser(); }
        }, delayMillis: 2000); //2 seconds
    }

    private void checkUser() {
        //get current user logged in
        FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();
        if (firebaseUser == null) {
            //user not logged in
            //start the main screen
            startActivity(new Intent( packageContext: SplashActivity.this, MainActivity.class));
            finish(); //finish this activity
        }
        else {
            //user logged in check user type
            DatabaseReference ref = FirebaseDatabase.getInstance().getReference( path: "Users");
            ref.child(firebaseUser.getUid())
                    .addListenerForSingleValueEvent(new ValueEventListener() {
                        @Override
                        public void onDataChange(DataSnapshot snapshot) {
```

Figure 4.1 Splash Screen navigates to MainActivity in if-else condition

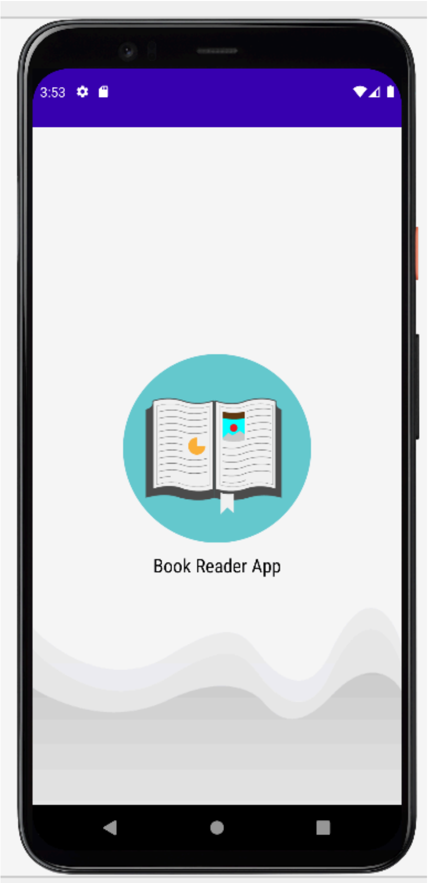


```
public void onDataChange(DataSnapshot snapshot) {
    //get user type
    String userType = ""+snapshot.child("userType").getValue();

    //check user type
    if (userType.equals("user")) {
        //if they are user, open user dashboard
        startActivity(new Intent( packageContext: SplashActivity.this, DashboardUserActivity.class));
        finish();
    } else if (userType.equals("admin")) {
        //if they are admin, open admin dashboard
        startActivity(new Intent( packageContext: SplashActivity.this, DashboardAdminActivity.class));
    }
}
```

Figure 4.1.2 User type check

In figure 4.2, there is a check process to check whether the user is Admin or User. If the user type is Admin, then Splash Screen navigates to DashboardAdminActivity. If the user type is User, then Splash Screen navigates to DashboardUserActivity.

Figure 4.1.3 Splash screen



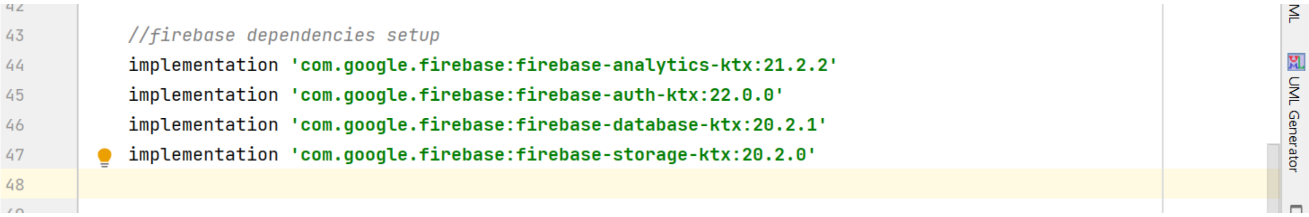
4.2 Firebase Integration Setup

To setup Firebase and integrate to the Android project, there are some prerequisites based on Google Firebase documentation:

1. Install or update Android Studio to its latest version.
2. Make sure that your project meets these requirements:
   a. Targets API level 19 (KitKat) or higher
   b. Uses Android 4.4 or higher
   c. Uses Jetpack (AndroidX), which includes meeting these version requirements:
      i. com.android.tools.build:gradle v3.2.1 or later
      ii. compileSdkVersion 28 or later
3. Set up a physical device or use an emulator to run your app.
   Note that Firebase SDKs with a dependency on Google Play services require the device or emulator to have Google Play services installed.

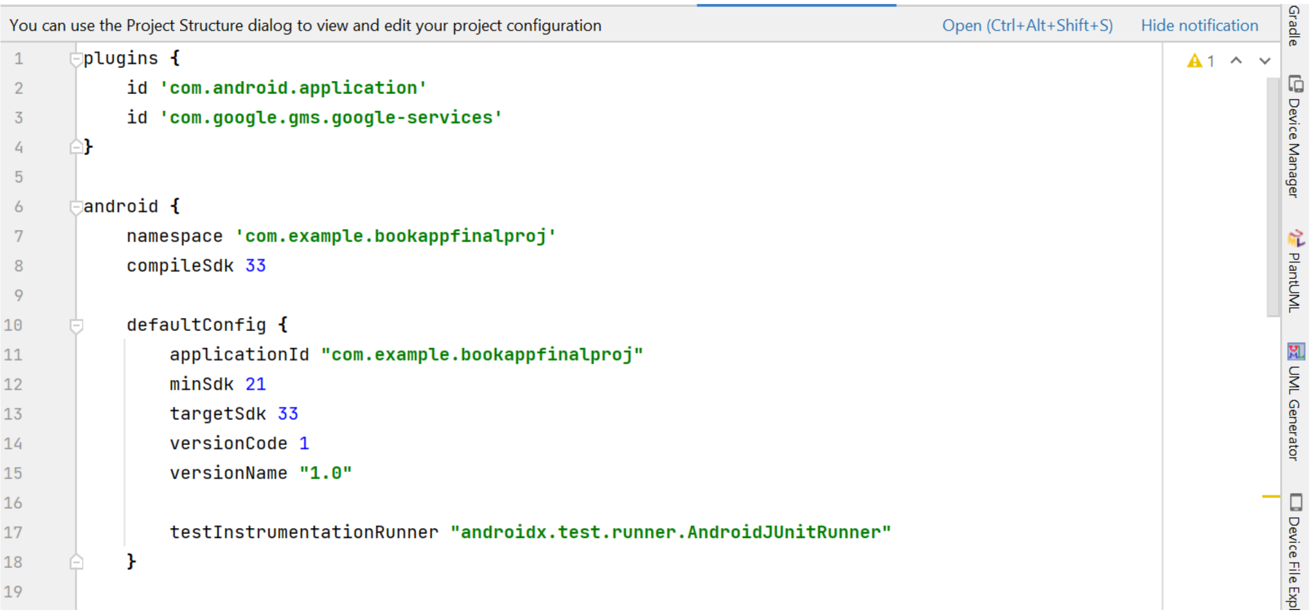
- Sign into Firebase using your Google account.

After prerequisites, the author initiates the implementation dependencies on the Android gradle file.



```
//firebase dependencies setup
implementation 'com.google.firebase:firebase-analytics-ktx:21.2.2'
implementation 'com.google.firebase:firebase-auth-ktx:22.0.0'
implementation 'com.google.firebase:firebase-database-ktx:20.2.1'
implementation 'com.google.firebase:firebase-storage-ktx:20.2.0'
```

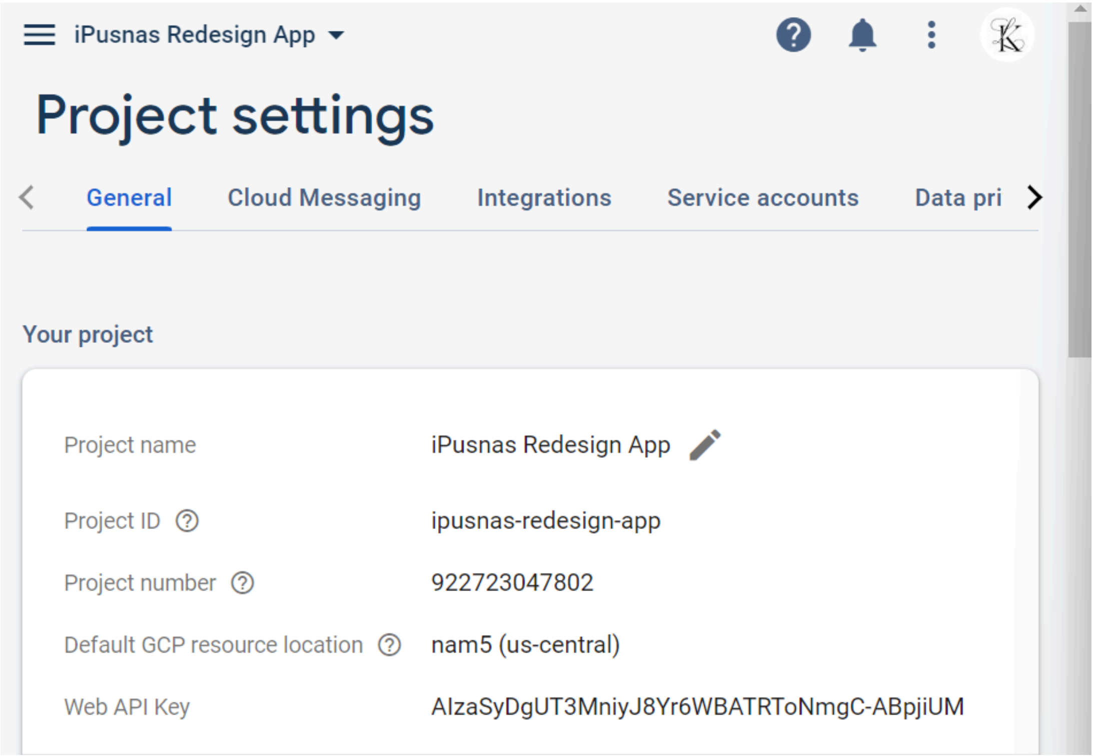
Figure 4.2 Firebase dependencies setup on build:gradle file (bookapp project)



You can use the Project Structure dialog to view and edit your project configuration

```
plugins {
    id 'com.android.application'
    id 'com.google.gms.google-services'
}

android {
    namespace 'com.example.bookappfinalproj'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.bookappfinalproj"
        minSdk 21
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

```

Figure 4.2.1 Initiate the app name in build:gradle (app) file



Project name iPusnas Redesign App

Project ID ipusnas-redesign-app

Project number 922723047802

Default GCP resource location nam5 (us-central)

Web API Key AIzaSyDgUT3MniyJ8Yr6WBATRToNmgC-ABpjiUM

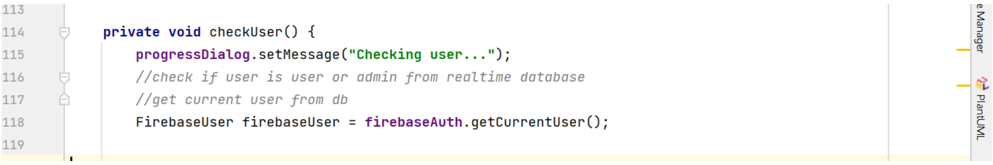
Figure 4.2.2 Database project profile

4.3 Login/Register Function

According to the questionnaire data before, the main problem of using the iPusnas app is the login process, including the application abruptly quits itself during the login process, the login button is difficult to find, and the users are unfamiliar with the login button.

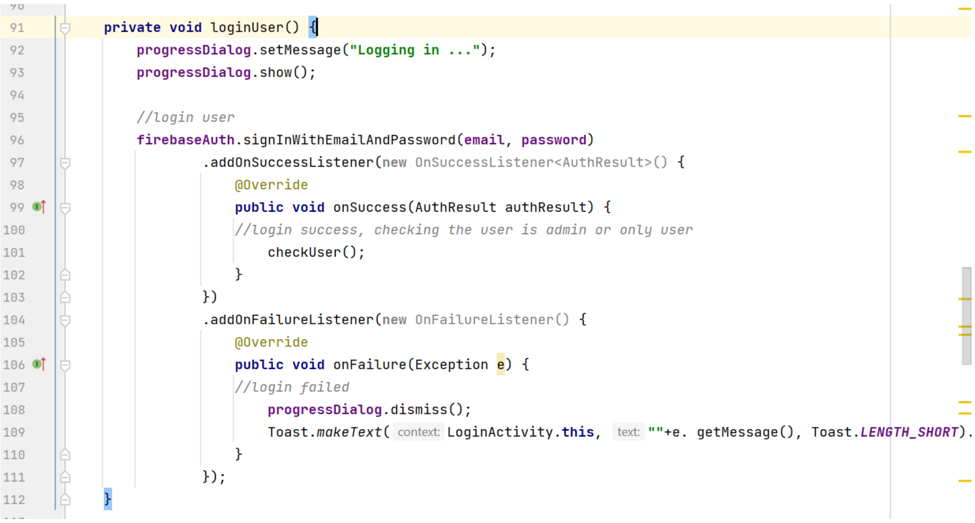
4.3.1 Login

In this function, other than the Login with No Account function, the Login page will come up first then let users input their email and password. Inside the LoginActivity, there is checkUser void that will be used to retrieve the data in project, validateData to match the password and email into the data they input before, and loginUser to match it with the data in the real-time database.



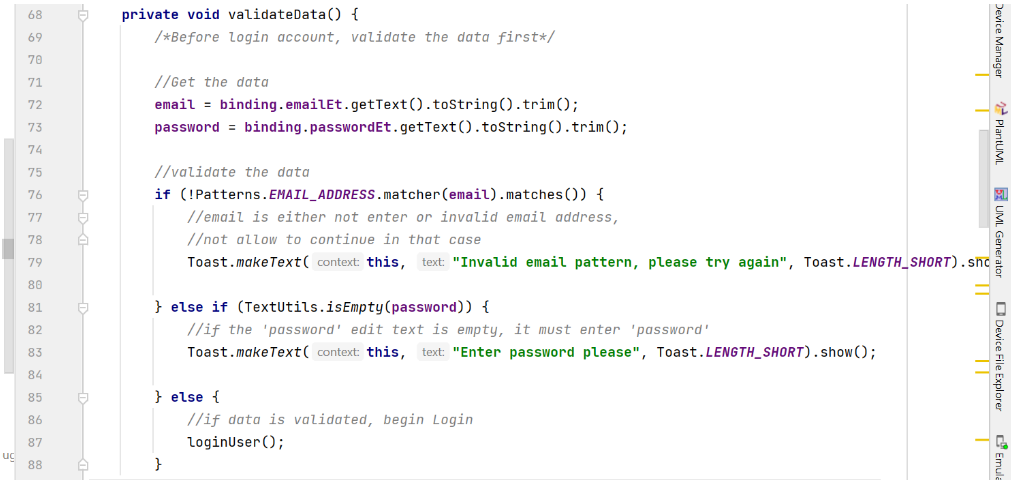
```
    private void checkUser() {
        progressDialog.setMessage("Checking user...");
        //check if user is user or admin from realtime database
        //get current user from db
        FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();

```

Figure 4.3 checkUser to retrieve the data



```
    private void loginUser() {
        progressDialog.setMessage("Logging in ...");
        progressDialog.show();

        //login user
        firebaseAuth.signInWithEmailAndPassword(email, password)
                .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
                    @Override
                    public void onSuccess(AuthResult authResult) {
                    //login success, checking the user is admin or only user
                        checkUser();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(Exception e) {
                    //login failed
                        progressDialog.dismiss();
                        Toast.makeText( context: LoginActivity.this,  text: ""+e. getMessage(), Toast.LENGTH_SHORT).
                    }
                });
    }
```

Figure 4.3.1 loginUser to match the data and the data in real time database



```
private void validateData() {
    /*Before login account, validate the data first*/

    //Get the data
    email = binding.emailEt.getText().toString().trim();
    password = binding.passwordEt.getText().toString().trim();

    //validate the data
    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        //email is either not enter or invalid email address,
        //not allow to continue in that case
        Toast.makeText( context: this,  text: "Invalid email pattern, please try again", Toast.LENGTH_SHORT).sho

    } else if (TextUtils.isEmpty(password)) {
        //if the 'password' edit text is empty, it must enter 'password'
        Toast.makeText( context: this,  text: "Enter password please", Toast.LENGTH_SHORT).show();

    } else {
        //if data is validated, begin Login
        loginUser();
    }
```

Figure 4.3.2 validateData to match the data and the data in user input before

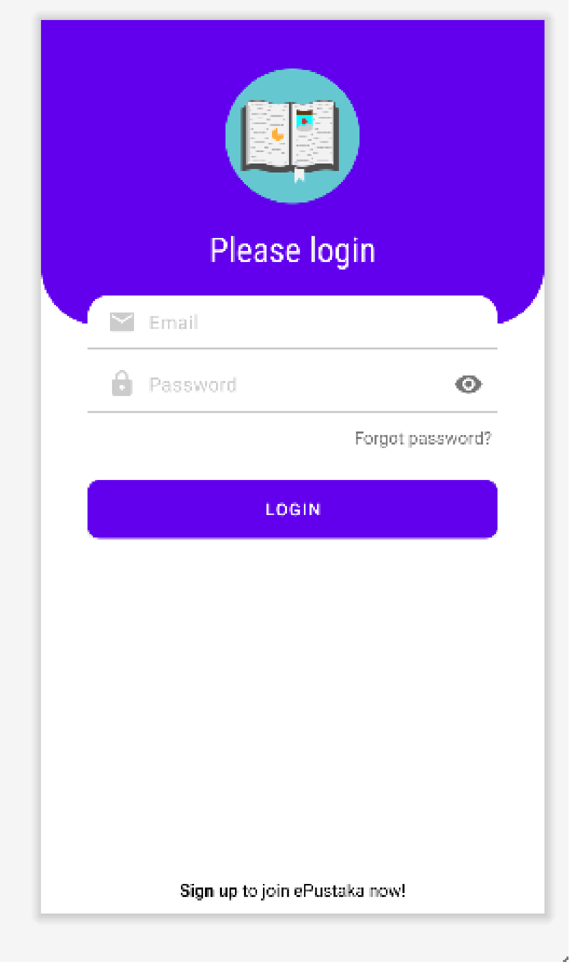
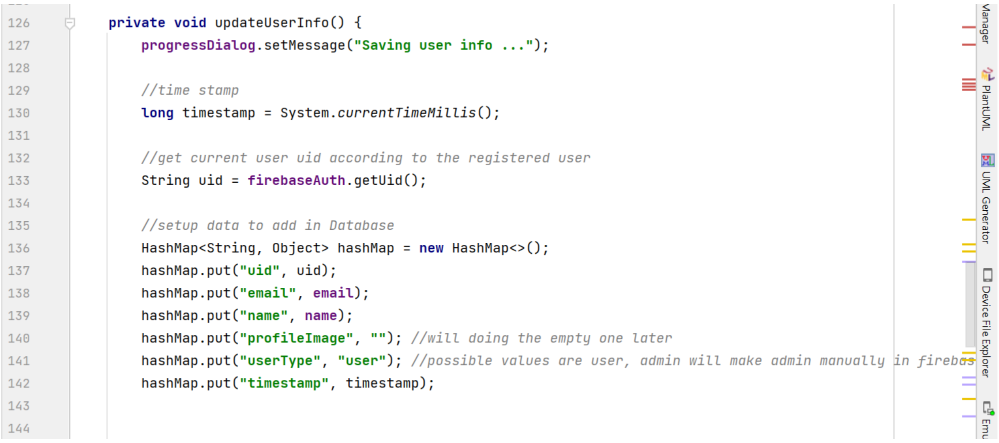


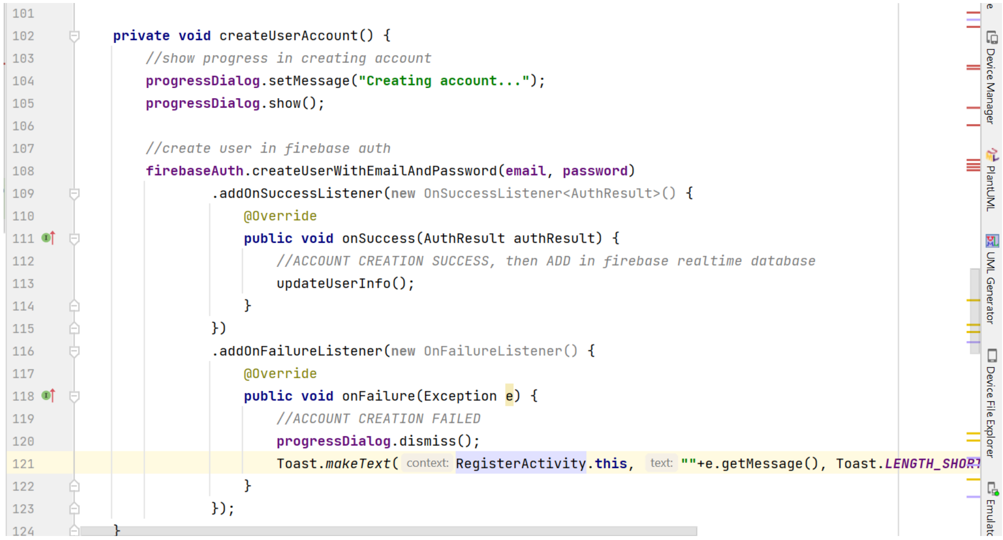
Figure 4.3.3 UI interface of login process

Below the Login process textfield, there is a Sign up button to click. This will ease users to create new accounts. Register account is storing the data with a hashmap between String and Object. The Register process starts with updateUserInfo, which creates a new instance filled with HashMap email, uid, name, userType, and timestamp from the project.



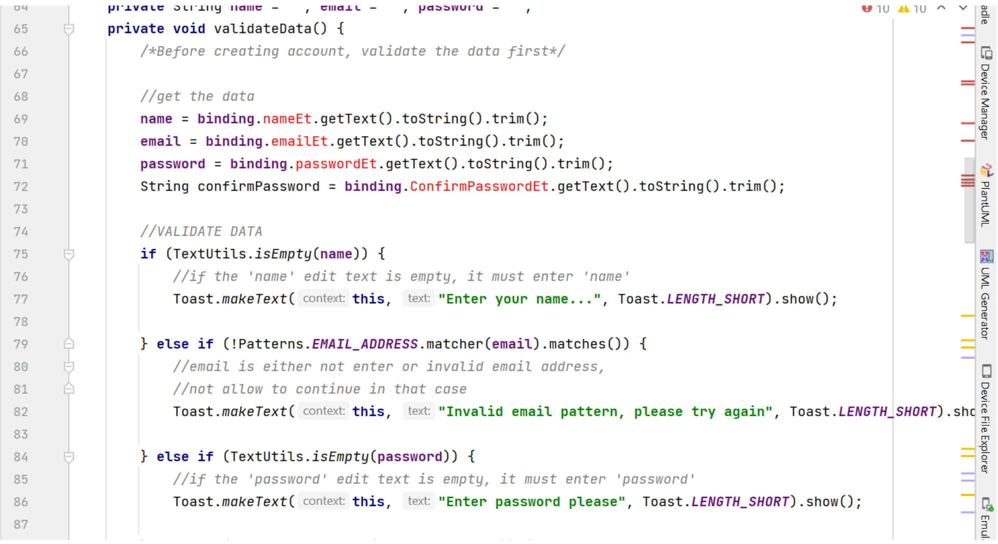
```
private void updateUserInfo() {
    progressDialog.setMessage("Saving user info ...");

    //time stamp
    long timestamp = System.currentTimeMillis();

    //get current user uid according to the registered user
    String uid = firebaseAuth.getUid();

    //setup data to add in Database
    HashMap<String, Object> hashMap = new HashMap<>();
    hashMap.put("uid", uid);
    hashMap.put("email", email);
    hashMap.put("name", name);
    hashMap.put("profileImage", ""); //will doing the empty one later
    hashMap.put("userType", "user"); //possible values are user, admin will make admin manually in firebas
    hashMap.put("timestamp", timestamp);
```

Figure 4.3.3 updateUserInfo



```
private void createUserAccount() {
    //show progress in creating account
    progressDialog.setMessage("Creating account...");
    progressDialog.show();

    //create user in firebase auth
    firebaseAuth.createUserWithEmailAndPassword(email, password)
            .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
                @Override
                public void onSuccess(AuthResult authResult) {
                    //ACCOUNT CREATION SUCCESS, then ADD in firebase realtime database
                    updateUserInfo();
                }
            })
            .addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(Exception e) {
                    //ACCOUNT CREATION FAILED
                    progressDialog.dismiss();
                    Toast.makeText( context: RegisterActivity.this,  text: ""+e.getMessage(), Toast.LENGTH_SHOR
                }
            });
}
```

Figure 4.3.4 createUserAccount in firebase authentication



```java
private String name = "", email = "", password = "";
private void validateData() {
    /*Before creating account, validate the data first*/

    //get the data
    name = binding.nameEt.getText().toString().trim();
    email = binding.emailEt.getText().toString().trim();
    password = binding.passwordEt.getText().toString().trim();
    String confirmPassword = binding.ConfirmPasswordEt.getText().toString().trim();

    //VALIDATE DATA
    if (TextUtils.isEmpty(name)) {
        //if the 'name' edit text is empty, it must enter 'name'
        Toast.makeText( context: this,  text: "Enter your name...", Toast.LENGTH_SHORT).show();

    } else if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        //email is either not enter or invalid email address,
        //not allow to continue in that case
        Toast.makeText( context: this,  text: "Invalid email pattern, please try again", Toast.LENGTH_SHORT).shc

    } else if (TextUtils.isEmpty(password)) {
        //if the 'password' edit text is empty, it must enter 'password'
        Toast.makeText( context: this,  text: "Enter password please", Toast.LENGTH_SHORT).show();
```

Figure 4.3.5 validateData

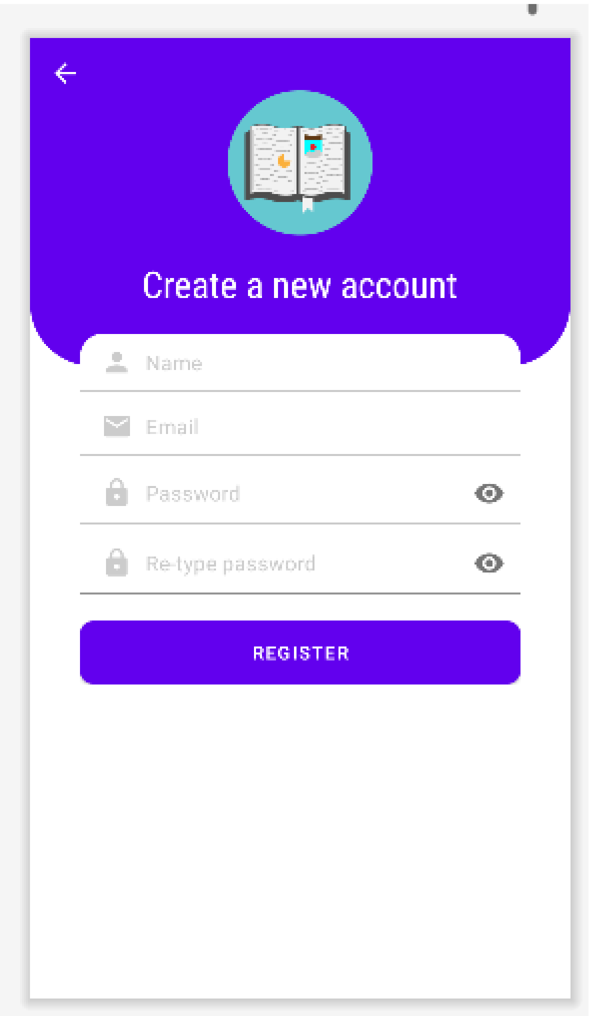
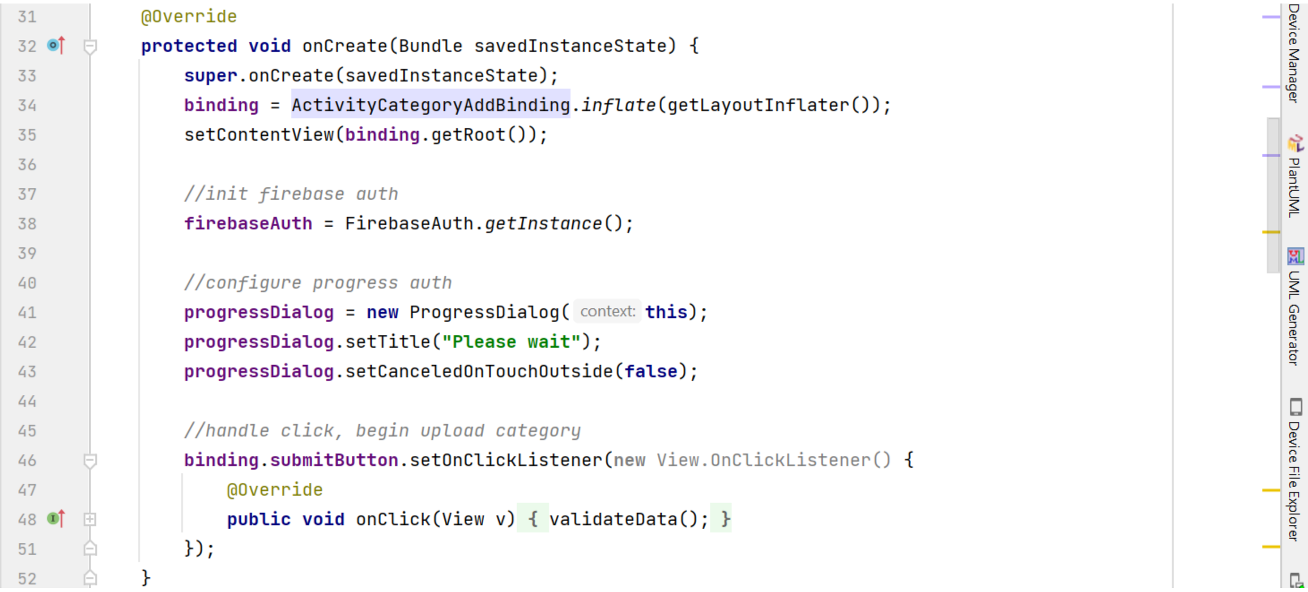


Figure 4.3.6 User interface from this Register process

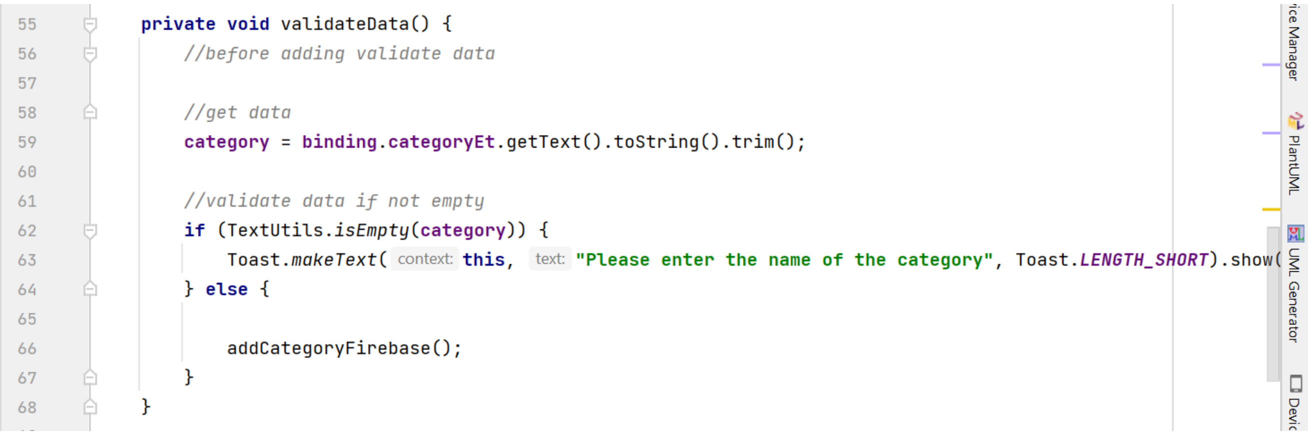
4.4 Add Category

Category is one of the ways to make something sorted. In this redesign of iPusnas application, add categories can be done by Admin users. The author retrieved the category from android project then stored in the database with validateData.



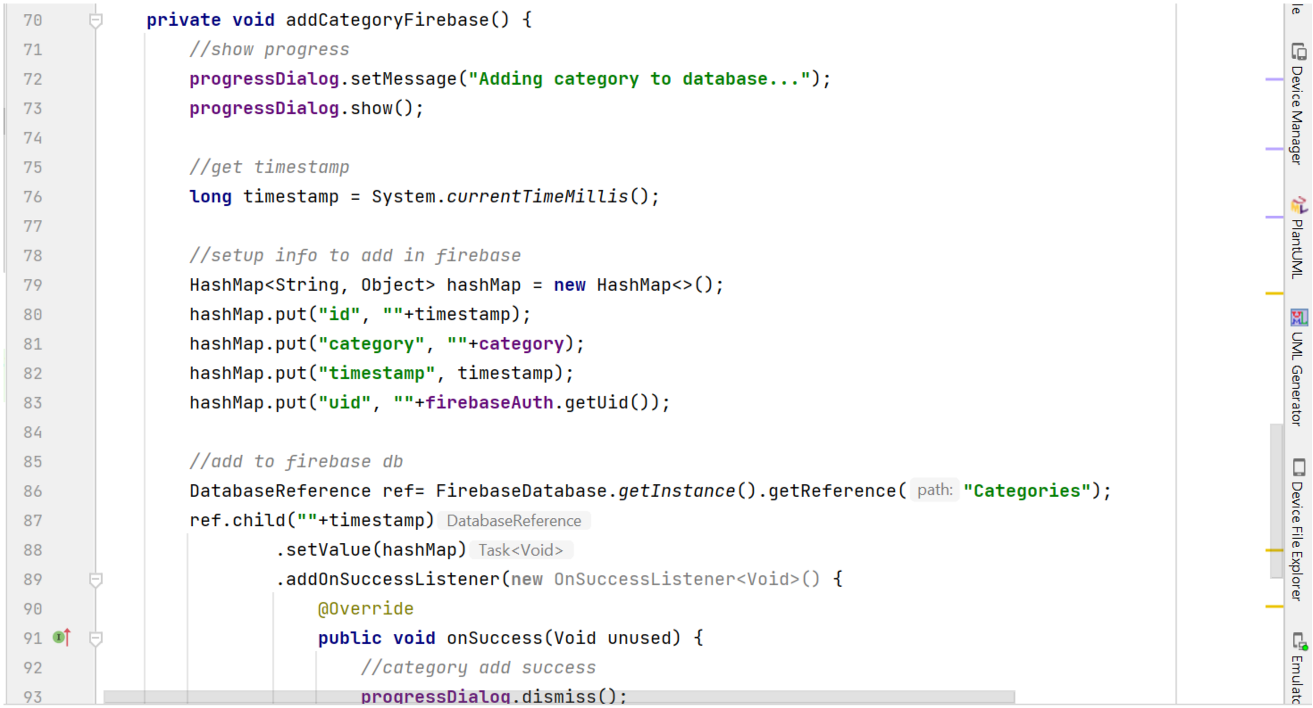
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityCategoryAddBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    //init firebase auth
    firebaseAuth = FirebaseAuth.getInstance();

    //configure progress auth
    progressDialog = new ProgressDialog( context: this);
    progressDialog.setTitle("Please wait");
    progressDialog.setCanceledOnTouchOutside(false);

    //handle click, begin upload category
    binding.submitButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { validateData(); }
    });
}
```

Figure 4.4.1 Initiate the Firebase authentication to store categories



```
private void validateData() {
    //before adding validate data

    //get data
    category = binding.categoryEt.getText().toString().trim();

    //validate data if not empty
    if (TextUtils.isEmpty(category)) {
        Toast.makeText( context: this,  text: "Please enter the name of the category", Toast.LENGTH_SHORT).show(
    } else {

        addCategoryFirebase();
    }
}
```

Figure 4.4.2 validate the data are not empty



```
private void addCategoryFirebase() {
    //show progress
    progressDialog.setMessage("Adding category to database...");
    progressDialog.show();

    //get timestamp
    long timestamp = System.currentTimeMillis();

    //setup info to add in firebase
    HashMap<String, Object> hashMap = new HashMap<>();
    hashMap.put("id", ""+timestamp);
    hashMap.put("category", ""+category);
    hashMap.put("timestamp", timestamp);
    hashMap.put("uid", ""+firebaseAuth.getUid());

    //add to firebase db
    DatabaseReference ref= FirebaseDatabase.getInstance().getReference( path: "Categories");
    ref.child(""+timestamp) DatabaseReference
            .setValue(hashMap) Task<Void>
            .addOnSuccessListener(new OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void unused) {
                    //category add success
                    progressDialog.dismiss();
```

Figure 4.4.3 add the category to firebase

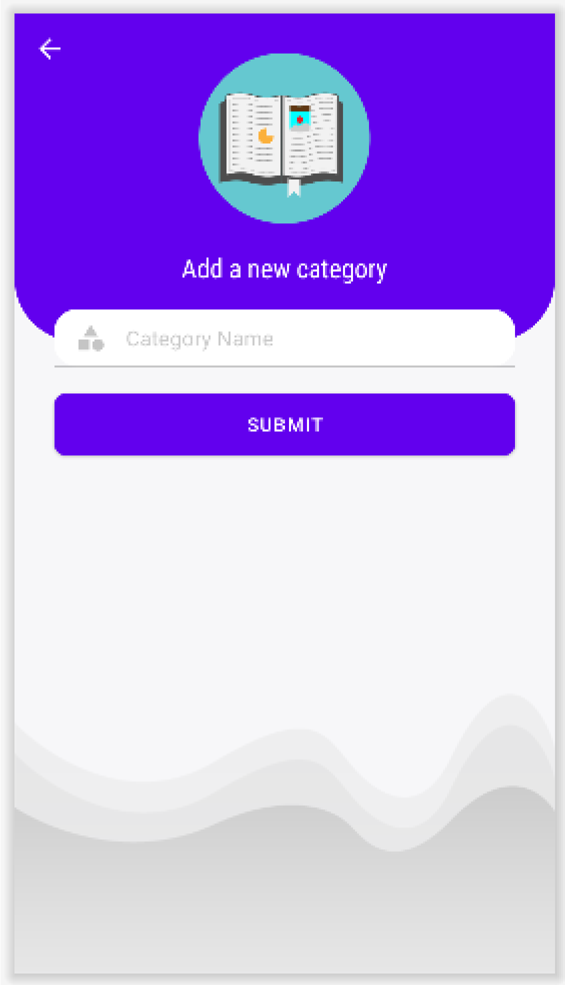


Figure 4.4.4 The user interface of Add Category

CHAPTER 5. SUMMARY/CONCLUSIONS

The redesign of iPusnas book app is a mobile app that allows users to experience the new version of iPusnas. This application is designed to provide an easy and convenient way for users to log in to the application with three different methods, register first then login, login, and login without account. The application typically includes features such as sorting the book by the categories and adding the categories for Admin users.

5.1 Conclusion

Based on the discussion from the previous chapters regarding the report final project entitled "Redesign of Ipusnas (Indonesian E-Book Reader App) with User-Centered Design Method" the author can draw several conclusions, such as following:

a. This redesign book application can meet the user needs and problem, which is the Login process. Users can login with three different methods.

b. This redesign book application can reduce the risk of loss of transaction data, because the data is stored in real time databases.

REFERENCES

[1] S. Masitoh, "Blended Learning Berwawasan Literasi Digital Suatu Upaya Meningkatkan Kualitas Pembelajaran dan Membangun Generasi Emas 2045," ICECRS UNIDA, vol. 1, no. 3, 30 March 2018.

[2] Sugiyono, Metodologi penelitian kuantitatif kualitatif dan R&D, Alphabeta , 2013.

[3] J. Sarwono, Metode penelitian kuantitatif & kualitatif, Graha Ilmu, 2006.

[4] K. V. P. W. S. T. C. Ji-Ye Mao, "The State of User-Centered Design Practice," Communication of the ACM, vol. 48, no. 3, pp. 105-109, 2005.

[5] H. J. W. W. Z. Chuanxiong Guo, "Smart-Phone Attacks and Defenses," ACM Sigcomm, 2014.

[6] H. M. M. H. Nawawi, The Instument of Social Project Research, Yogyakarta: Yogyakarta Gadjah Mada University Press, 1992.

[7] L. A. a. G. L. (FIMI), "User Centred Design for EASYREACH," ISO 9241- 210: 2010, pp. 1-45, 2010.

[8] ISO, "Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems," Journal of Ergonomics, vol. 30, no. 1, 2010.

[9] P. S. John Feather, International Encyclopedia of Information and Library Science, Library Science (Routledge), 2003.

[10] A. H. Bernard Renaldy Suteja, "PERANCANGAN USER INTERFACE E-LEARNING BERBASIS WEB," Seminar Nasional Informatika 2008 (semnasIF 2008) UPN ”Veteran” Yogyakarta, vol. 1, no. 5, 2008.

[11] T. D. S. A. H. P. Muhammad Nauval El Ghiffary, "Analisis Komponen Desain Layout, Warna, dan Kontrol pada Antarmuka Pengguna Aplikasi:· Mobile Berdasarkan Kemudahan Penggunaan (Studi Kasus: Aplikasi Olride).," Jurnal Teknik ITS, 2018.

[12] K. Teguh Martono, "Augmented Reality sebagai Metafora Baru dalam Teknologi Interaksi Manusia dan Komputer.," Jurnal Sisten Komputer Diponegoro University,

vol. 1, no. 2, 2011.

[13] L. R. Karie Terrell Hanna, "Google Firebase," TechTarget | Mobile Computing, May 2023 . [Online]. Available: https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase.

[14] P. o. t. R. o. I. &. T. H. o. R. o. t. R. o. Indonesia, Indonesia Library Law No. 43 year 2007 or Undang Undang Republik Indonesia No. 43 tahun 2007 tentang Kepustakaan, Jakarta: Ministry of State Secretariat of the Republic of Indonesia, 2007.

[15] B. O. Lubis, "Peranan Pengetahuan Desain Komunikasi Visual dalam Pengajaran Matakuliah Interaksi Manusia dan Kamputer," in Seminar Nasional Aplikasi Sains & Teknologi (SNAST), Yogyakarta, 2014.

[16] " Android Studio Developers Documentation," 17 May 2023. [Online]. Available: https://developer.android.com/studio/intro. [Accessed 20 May 2023].

[17] S. Basuki, Sejarah Perpustakaan Nasional Republik Indonesia / Penyusun : Sulistyo Basuki, Jakarta: Perpustakaan Nasional RI, 2016.

[18] " What is Class Diagram," N/A. [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/. [Accessed 21 May 2023].