

# ANDROID-BASED MOBILE APPLICATION DEVELOPMENT EXPENSIFY: PERSONAL EXPENSE MANAGER

Rizki Fajar Maulian<sup>1</sup>, Ronny Juwono<sup>2</sup>  
Faculty of Computing  
President University  
Bekasi Indonesia  
rizkifajarmaulian@student.president.ac.id

**Abstract**— Personal financial management is no less important to achieve personal prosperity. By planning, managing, and controlling their finances wisely and carefully, each individual can consider their current financial needs and conditions, to determine future financial goals. By having financial planning, each individual can manage their financial resources optimally to achieve the desired financial goals. However, there are still many individuals who are still not able to manage their finances properly. In addition to a lack of knowledge in managing personal finances, consumptive behaviour is also a contributing factor. One effort that can be done to overcome this is by making financial records, by recording daily expenses and income, individuals can more easily control and manage their finances. This study aims to develop an Android-based personal expense management application using the Kotlin programming language. This application has features that can assist users in managing their finances that can be used anywhere and anytime.

**Keywords:** *financial planning, financial management, Android*

## I. INTRODUCTION

Nowadays, many individuals have not been able to plan and control the use of money to achieve personal goals, especially among the younger generation. Planning for personal finances needs to be done to be able to prosper in the future and be able to overcome unwanted financial problems. Financial management is an activity that includes planning, analyzing, and controlling finances to achieve a goal in maximizing the use of money [1].

However, there are still many individuals who have not been able to manage their finances wisely and carefully [2]. In addition, many individuals are often lazy to records their daily expenses because they have to do the recording manually on paper, notebooks or in spreadsheet programs such as Excel. This is because manual financial records do not have well-administered financial management [3]. First, the recording of transactions to a notebook and the calculations are done manually or using a calculator, making it time-consuming and difficult to monitor financial flows. Second, manual recording is prone to miscalculations that can affect future financial planning. Third, manual recording requires repeated checks and more thoroughness to produce precise and accurate financial reports [4].

To solve these problems, we need a system that can help and make it easier to manage personal finances. With this system, users can easily learn to manage finances independently through their Android-based smartphones. Users can manage finances by recording detailed income and expense transactions based on categories, such as food and beverages, entertainment, transportation and other categories. Users can add certain categories according to their needs. This system can also display financial statement statistics in graphical form for weekly, monthly, or yearly periods. Users can also view reports by category. With this system, users can also limit their spending budget according to the desired category. This system is expected to be able to assist and facilitate users in managing their finances wisely so that they can realize their personal financial goals and become prosperous individuals.

Based on the background described above, it can be formulated into several problem formulations. First, how to design an Android-based personal financial management application. Second, how to manage personal finances independently. Third, how to record income and expenditure transactions in detail by category. Fourth, how to monitor spending based on a predetermined spending budget limit. Fifth, how to monitor and analyze expenditure and income statistics with graphs and diagrams. In addition, the main purpose of this research is to solve some of the problems. First, to develop an Android-based personal financial management application. Second, to help record income and expenditure transactions in detail by category of transactions. Third, to help analyze, plan, and control personal finances independently using a smartphone device.

This research focuses on developing applications that can help users to manage, analyze, plan, and control personal finances. The direct application can be used without logging in or registering first into the application. Financial analysis will be carried out in the background. The financial summary will be displayed based on the data that has been recorded by the user into the application. The history of recorded transactions can also be viewed and filtered by category and time period. From the background described above, there are several limitations to the problem. First, data can only be stored locally, so transaction data cannot be synchronized to other devices. Second, the user must export the database first so that the recorded data can be displayed on other devices. Third, the account has not been able to record automatic transactions by synchronizing the user's credit card or bank account. Fourth, the system built can only be run on Android with a minimum operating system of API 21 or Android 5.0 Lollipop and above.

## II. METHOD

The methodology chosen by the author to develop this application is the Rapid Application Development (RAD) method. This development methodology prioritizes rapid prototyping and receiving rapid feedback from users so that changing system requirements can be implemented into the system quickly compared to traditional application development and testing cycles that are too tied to a predetermined development schedule. In this methodology, the system requirements are divided into small components, so that changes can be made during the development process [5]. That way, developers can quickly create multiple iterations and software updates without the need to start a development schedule from scratch. The RAD methodology has 4 stages, as shown in Figure

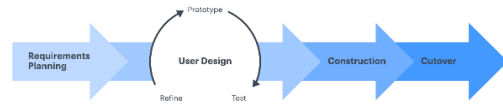


Figure 1 Rapid Application Development

In the early stages of this development method, the development team, software users, and other team members communicate to indicate the objectives of the development project, define software system requirements, and identify current and potential problems that will be encountered during software development. In the second stage, users work hand in hand with developers to ensure their needs are met at every step in the software design process, where users can test each product prototype at each stage to ensure the application meets their expectations. In the third stage, developers, programmers, and testers work together to ensure that everything works smoothly and that the end result meets the requirements of the software being designed. In the last stage of the RAD methodology, the software will be tested starting from its functionality and performance using use case tests to ensure that the developed software works as expected.

## III. DESIGN AND IMPLEMENTATION SYSTEM

### A. System Overview

Expensify: Personal Expense Manager was developed to help analyze, plan, and control personal finances independently using an Android-based smartphone device. The main feature of this application is to record the user's financial income and expenditure transactions. Based on the transaction data that has been recorded, the system will analyze and then display financial summaries, reports and graphs to the user. Users can add additional categories according to their needs. Another additional feature is that users can increase their budget for a certain period.

### B. Functional Requirements

Functional requirements define what the system should be able to do. In this study, functional requirements are explored based on literature studies on Financial Planning that have been reviewed. Each functional requirement will be assigned the code SRS\_EPY\_F\_X. SRS stands for System Requirement Specifications, EPY stands for Expensify, F stands for functional requirements and X stands for the number of requirements definitions. Functional requirements in Expensify can be seen in Table 1 below.

Table 1 Functional Requirements

No	Code	Description
1	SRS_EPY_F_01	Users can see a summary of their financial information.
2	SRS_EPY_F_02	Users can record details of expense transactions.
3	SRS_EPY_F_03	Users can record details of income transactions.
4	SRS_EPY_F_04	Users can view details of transactions that have been recorded.
5	SRS_EPY_F_05	Users can change the details of transactions that have been recorded.
6	SRS_EPY_F_06	Users can delete the details of transactions that have been recorded.
7	SRS_EPY_F_07	Users can view a list of financial history based on a certain period.
8	SRS_EPY_F_08	Users can add transaction categories.
9	SRS_EPY_F_09	Users can change transaction categories.
10	SRS_EPY_F_10	Users can delete transaction categories.
11	SRS_EPY_F_11	Users can create budgets.
12	SRS_EPY_F_12	Users can see the details of the budget that has been made.
13	STS_EPY_F_13	Users can change the budget that has been created.
14	SRS_EPY_F_14	Users can delete the budget that has been created.

C. Non-Functional Requirements

Non-functional requirements are requirements that are not directly related to the user but provide quality to the system and user satisfaction. Non-functional requirements on Expensify can be seen in Table 2.

Table 2 Non-Functional Requirements

No	Parameter	Description
1	Usability	The interface design is easy to use by users based on effectiveness, efficiency, and satisfaction [6].

2	Compatibility	The app must run on Android hardware with a minimum of API 21 or Android 5.0 (Lollipop).
---	---------------	--

1. Use Case Diagram

Use Case Diagram is a diagram that is used to describe a series of actions (use cases) that the system must or can do (subjects) and visualize the interactions of these various things with users (actors). Use Case Diagrams represent the core parts of a system and the workflow between them. With the help of use case diagrams, we can find out the conditions before and after interacting with the user (actor). The use case diagram of Expensify can be seen in Figure 2.

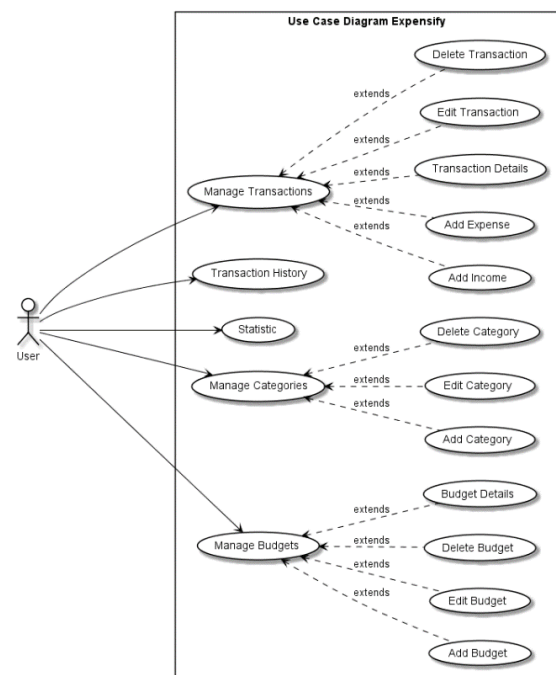


Figure 2 Use Case Diagram Expensify

D. Design and Implementation

1. System Design

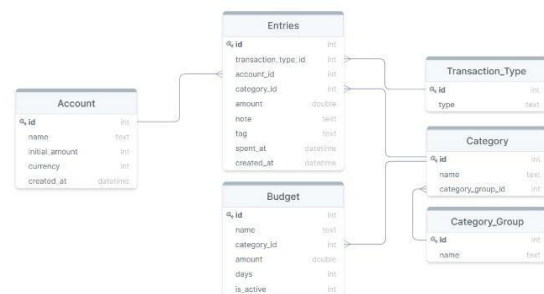


Figure 3 ERD Expensify

Figure 3. is the ERD design of the Expensify application. Entity Relationship Diagram or ERD is a structural diagram used to design a database. The components of the ERD consist of entities, entity attributes, primary keys, foreign keys, and the relationship itself [7].

## 2. User Interface Design

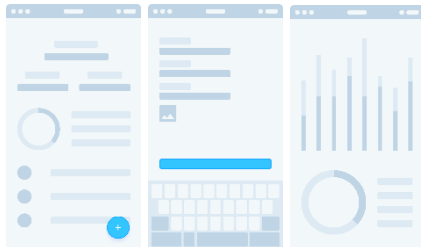


Figure 4 Expensify’s User Interface Design

In Figure 4. above are some samples of the low-fidelity design of the Expensify interface. Interface design or User Interface Design is one process that is no less important when designing a system. The interface serves to bridge between the user and the system.

## E. System Implementation

### User Interface Implementation

The process of implementing the program code is divided into 2 parts. First, the database and system algorithms are implemented using Kotlin programming language recommended by Google for Android application development. Second, the interface is implemented using XML markup language.

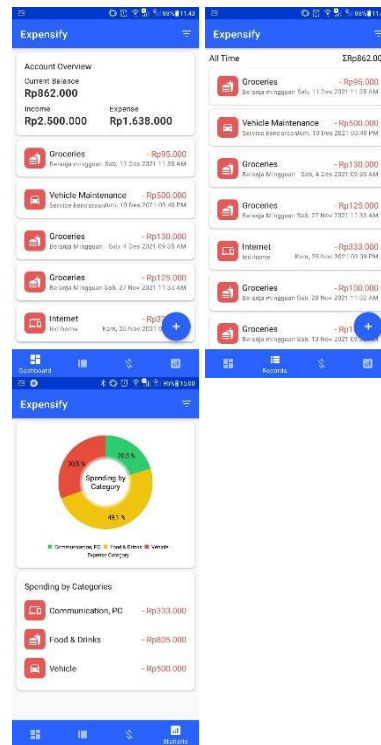


Figure 5 User Interface Implementation

In Figure 5 above, it can be seen the results of the implementation of several interfaces on the Expensify Application. The Expensify app has 10 interfaces. The ten interfaces are Dashboard page, Add Edit Transaction page, Transaction Details page, Transaction List page, Categories page, Categories page, Add Edit Category page, Budgets page, Add Edit Budget page, Budget Details page, and Statistics page.

## IV. RESULT

### A. Functionality Testing

Functional testing is intended to validate the functional requirements that are made to run properly and are per the requirements that have been set in the requirements analysis. To perform functional testing, the black box testing method is used to test whether the input and output provided are appropriate or not. The results of functional testing of the Expensify application can be seen in Table 3.

Table 3 Functional Requirements Test Results		
No	Code	Description
1	SRS_EPY_F_01	Valid
2	SRS_EPY_F_02	Valid
3	SRS_EPY_F_03	Valid

4	SRS_EPY_F_04	Valid
5	SRS_EPY_F_05	Valid
6	SRS_EPY_F_06	Valid
7	SRS_EPY_F_07	Valid
8	SRS_EPY_F_08	Valid
9	SRS_EPY_F_09	Valid
10	SRS_EPY_F_10	Valid
11	SRS_EPY_F_11	Valid
12	SRS_EPY_F_12	Valid
13	STS_EPY_F_13	Valid
14	SRS_EPY_F_14	Valid

#### B. Compatibility Testing

Compatibility testing is intended to determine whether the Expensify application can work in different environments, both in terms of software and hardware. Tests are carried out by running the Expensify application on devices that have an android operating system with different versions of API levels. The test device must be able to complete 14 test cases that refer to the use case. The results of this test, the application runs well and can complete all test cases on Android with API level 21 and above, while on API level 21 and below the application cannot be launched.

#### V. CONCLUSION

From the results that have been obtained based on the stages of needs analysis, design, implementation, and testing that have been completed previously, several conclusions can be obtained. First, the needs analysis on the Expensify application is based on the study of Personal Financial Planning and Consumptive Behavior in the Era of Globalization resulting in 14 functional requirements which are divided into 3 groups, Manage Transactions, Manage Categories, and Manage Budgets. In addition, this analysis also produces 2 non-functional requirements in terms of usability and compatibility. Second, the system implementation is implemented on a mobile application platform with

the Android operating system using the MVVM (Model-View-ViewModel) architecture and Jetpack Components. Fourth, the tests carried out by testing the functionality show that the functionality requirement features have a 100% success rate as expected. The features in Manage Transaction can add new transactions, view detailed transaction details, modify transactions, and delete transactions. Manage Categories features can add new transaction categories, change transaction categories, and delete transaction categories. Features in Manage Budgets can add new budgets, view detailed budget details, modify budgets, and delete budgets. While in system compatibility testing, the Expensify application can only run on devices running the Android operating system level API level 21 (Android 5.0 Lollipop) and above. On devices with API level 21 and below, the Expensify app cannot be installed.

#### REFERENCES

- [1] S. Husnan, *Manajemen Keuangan*, 2 ed., Tangerang Selatan: Universitas Terbuka, 2014
- [2] Otoritas Jasa Keuangan Indonesia, "Survei Nasional Literasi dan Inklusi Keuangan 2019," 1 December 2020. [Online]. Available: <https://www.ojk.go.id/id/berita-dan-kegiatan/publikasi/Pages/Survei-Nasional-Literasi-dan-Inklusi-Kuangan-2019.aspx>. [Accessed 31 October 2021].
- [3] S. Rahmayuni, "Analisi Komparasi Proses Laporan Keuangan Secara Manual dan Menggunakan Aplikasi Komputer Akuntansi," SNITT- Politeknik Negeri Balikpapan, pp. 148-152, 2017.
- [4] R. Julian, "Perbandingan Pencatatan Penjualan Secara Manual Dengan Aplikasi Penjualan," Zahir, 20th July 2017. [Online]. Available: <https://zahiraccounting.com/id/blog/perbandingan-pencatatan-penjualan-secara-manual-dengan-aplikasi-penjualan/>. [Accessed 28 October 2021].
- [5] C. V. Geambasu, I Jianu, I Jianu, A Gavrila "Influence Factors for the Choice of a Software Development Methodology," *Journal of Accounting and Management Information Systems*, vol. 10, no. 4, pp. 479-494, 2011.
- [6] J. Sauro and J. Lewis, *Quantifying the User Experience*, Cambridge: Elsevier, 2016.
- [7] M. A. Mohammed, D. A. Muhammed and J. M. Abdullah, "Practical Approaches of Transforming ER Diagram into Tables," *International Journal of Multidisciplinary and Scientific Emerging Research*, vol. 4, no. 2, pp. 1106-1110, 2015.