International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 03 | March - 2022 ISSN: 2582-3930

Plutus: Money Management Application for Smart Phones

Sharath D[1], Raghavendra R[2]

[1]Student, School of CS & IT Jain (Deemed to be University), Bengaluru, India
[2]Professor, School of CS & IT Jain (Deemed to be University), Bengaluru, India

---

Abstract - Money management in the current world has become a highly stressful profession, and the introduction of online payments such as UPIs, Net Banking, and other similar services has made the task of keeping track of the money we save even more difficult.

We reviewed the various approaches for constructing a more secure and efficient mobile application that collects and calculates inputs faster with a shorter reaction time than typical apps in this study.

***Key Words*:** Mobile application; money manager; application development; manage expenses; user interface development; gamification

1. INTRODUCTION

Technological advancements have
accelerated in the modern era, particularly in mobile technologies.-As a result, as the number of mobile users grows, so does the number of mobile applications available to them. There are many software developers who wish to create a new mobile app to meet the needs of customers, and these apps are currently in great demand in the technology industry.
A mobile application is a piece of software designed to run on mobile phones and tablets. [1]

The developed smartphone application will assist ordinary people in managing their finances and learning how to conserve money.
The application's implementation includes a money manager that customers may use to save money and assist them in managing their finances.
It became easier for consumers to track money on a monthly or daily basis as a result of this.
Inability to comprehend financial ideas can exacerbate a person's financial situation and lead to poor financial decisions.

Gamification has a function to play in people's engagement and learning.
According to Zainuddin, Shujahat, Haruna, and Chu [2], gamification is capable of promoting motivation and engagement, improving user participation and instructional interactivity, and leading to knowledge acquisition [3].
People can provide motivation to save money and learn to manage their finances.

The primary goal of this project is to create a smart mobile money manager for regular people.

The other goal is to see how effective a money management app is in teaching people how to handle their money.

2. Literature Review

Recording each transaction and managing day-to-day expenses and incomes based on user requirements very tedious process. While recording the transactions there are chances to show only recently recorded transactions. Keeping records of small expenses and incomes are very furious as transacting them through UPIs and another transaction like IMPS gives us only the transaction ID and there are no possible remarks for the recorded transaction in bank passbook and even the UPI apps transaction history may contain only the names of the transacted person's name only and there may be no more available information for this, like what kind of shop it is or for what the expense or income had been recorded.

Setting up Log-in credentials and signing up to different companies are a threat to data. The basic ideology of this application is to be as user-friendly as possible avoiding the basic constraints like signing up for an account, filling up personal details, etc…
The main ideology is to avoid unwanted information and keep the application user interface minimalistic as possible. The application requires only a username and the available savings balance of the user.
The further inputs are given by the user as the date of the transaction, the amount of the transaction, and comments (purpose) of the transaction. The application processes the data obtained and outputs the required operation by adding the income to the current savings balance or deducting the expense from the savings.
The application also sets reminders in a form of notification for future expenses and incomes finally it provides a strategic view of the total expense and incomes for the month in form of a bar graph and pie-chart.

Module Design

This system is basically composed of 7 modules as mentioned below.

1. **Add Income.**
2. **Add Expense.**
3. **Add Due-Dates.**
4. **Add Income-Dates.**
5. **Edit Income.**
6. **Edit Expense.**

7. **View profile.**

1. Add Income
Make an entry to the application to add income. The user can use this function to add an entry to increase the amount of the savings, the input is in form of numerical, and also comments sections are used to save the transaction in a name that the user finds easy to remember.

2. Add Expense
Make an entry to the application to add expense. The user can use this function to add an entry to deduct the amount from the savings, the input is in form of numerical, and also comments sections are used to save the transaction in a name that the user finds easy to remember.

3. Add Due-Dates
The user can set a reminder for the upcoming expense on the calendar the application will let the user know about the expense a day prior and keeps on reminding the user that he is due for a bill or so.

4. Add Income-Dates
The user can set a reminder for the upcoming income on the calendar, the application will let the user know about the income that is being expected a day prior and keeps on reminding the user that he is expecting an income.

5. Edit Income
If the user enters a wrong entry due to a mistake. The user can alter the entry by just editing it and also it provides an option to delete an income.

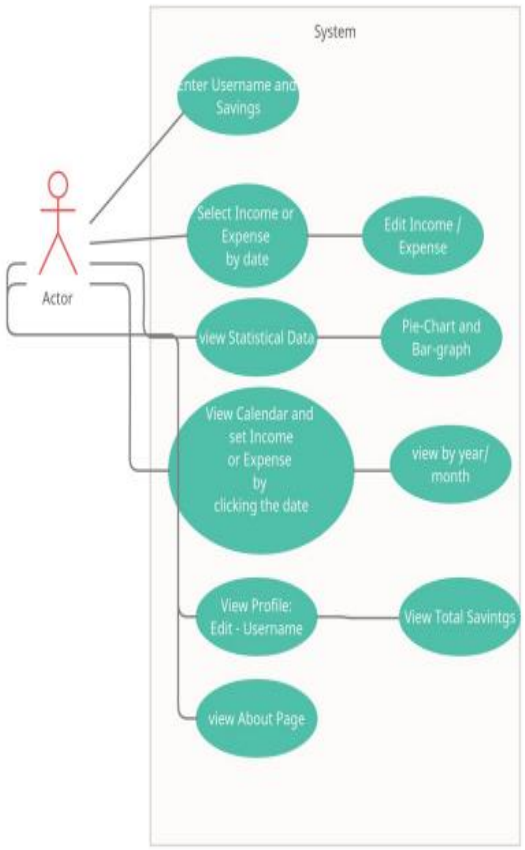
6. Edit expense
If the user enters a wrong entry due to a mistake. The user can alter the entry by just editing it and also it provides an option to delete an expense.

7. View Profile
The user can view the profile to find the total savings in his balance and also the recorded transaction. He can overview the total activities (income and expense) in this module.

Procedural Design

The Plutus application is being designed for common people who have day-to-day expenses it can be in a lesser amount to thousands.

- The user needs to launch the application. The application will request the username and total savings as of the current date.
- The application will take the customer to the main page. On this page, the user will be given a calendar interface.
- The user can select the date whether it's current date or previous date or future date and add an entry for an expense or income.
- After selecting the date the application will prompt the user to select whether it is an expense or income or add a custom entry.
- The User needs to select either one of the options.
- If the user had selected income, the application will display the options to save the transaction i.e. Income, Loan, Miscellaneous, Salary, Rent.
- The User then needs to select the category and enter the amount.
- The changes in the savings balance will be altered and we can see a transaction had been committed on the selected date.
- The user can select an expense similarly and the application will prompt the user with different expense options the user needs to select one and the application will ask for the amount of expense and the user needs to comment on the section for what the expense is.
- The user can select a custom entry it is a combination of both expense and income, the user needs to select his own category of the income or expense. And the entry will be added and the effect will be displayed on the total savings.
- The user can set notifications for the future transaction be it an expense or income. The application will notify the user that on the entered date certain expense or income is being expected. The main ideology of this module is to make sure that the user does not forget the expense or income committed prior to the particular date.
- The application will provide a strategical view of the total expense and income carried out in the entire month, week and day. It represents them in a bar graph and pie-chart model.
- The user can go to the profile and see the overview expense and income modules. The application provides to show all the transactions separately. The user can alter or delete the entered transaction in the module.



3. Report Design

The structure of the report that will be generated by the system. The system basically collaborates with the different modules to render the final outcome such as income, expense, custom entry, edited income, edited expense, deleted income, deleted expense.

Inputs given to generate the report

- Enter total savings.
- Enter income.
- Enter expense.
- Enter future date for due-dates.
- Enter future date for incomes.
- Edit Income
- Edit expense

Output fields in the generated report

- Expense Category.
- Income Category.
- Due date reminder.
- Income date reminder.
- Calculate total savings.
- Render bar graph for the income and expense for the month.
- Render pie chart for the income and expense for the month.

Implementation Approaches

The implementation of this application faced multiple challenges and hurdles during the initial stages of the system designing and implementation.
Implementing various technologies and concepts like tabung application, expense manager helped to understand the user requirements. Testing is done at multiple frames. Testing is extensively done at various stages of the development process and testing life cycle, making sure quality application is delivered that helps customers get the required application to maintain their transactions.

The initial prototype was implemented with mere requirements from the clients which in turn took multiple turns to attain a stable application. Requirements kept on changing since the project is huge and has high impact on real-time use, all requirements were kept at its priority levels and addressed consecutively over the time. Multiple teams across multiple locations worked on specific modules and features of this applications. Each feature implementation went into content review where the analysed requirement is reviewed, then Solution Review is done to review the proposed solution for the feature, Code Review basically happens to check the designed code standards and feasibility, Then the code is adopted and testing begins.

Unit Testing

Unit testing was basically performed by the developer during and after the feature or UI element implementation and appropriate action was taken. This process or phase involved identifying all the units that were being developed and making sure they are displayed and behaves as expected under different user interaction inputs. Since this application contains three main modules and multiple sub modules, so many elements, forms and other objects being used to build the interface and function which in turn resulted in big unit testing curve.
Example 1: Logon button – Logon button was tested with all the possible positive and negative testing scenario to check if it works well all the time.
Example 2: Sign up button – Sign up button was tested with all possible validations such has email, integer and variable validations.

Integration Testing

This is the testing phase where all the developed individual components of the application are glued together to see if it functionally works fine. This system is made up of modules like Seller and Customer who needs to talk to each other in getting the services done from each. And this basically involves interaction that usually occurs while integrating two different things which may arise due to change of code or functional behaviour and this was tested. This testing was done by the tester.
Example 1: Accessing furniture details is done by the cart module from the customer, so these two interface interactions were tested to check nothing was interrupted.

Example 2: upload module and cart modules are coupled together each accesses information from the other while processing the furniture from the seller. And this was tested with system testing test case to check nothing was broken or impacted in thefunctionality.

4. Conclusion

This application provides a computerized version of the Money Management System for individuals facing problems with being on date for their bill payments so there is no due date in his/her calendar. Not only that but also help the individual to oversee her future expenses, income and what will be the overall savings be for this particular month.
Helps to cut off unwanted expenses in his/her list and add that to the savings list to increase the volume in savings for their benefit.

5. Future Scope

In the future upgrade, there will be an AI monitoring the income and expenses and also provide efficient ways to increase the savings in the user's life. Providing the direction to save up the incomes and also give an efficient way to spend the incomes so that the expense is less which gradually increase the Savings it will benefit the users to take off tension from the money management crisis.
The application will allow the user to download the transaction history in many formats of choice like the excel format, word document format, pdf format and others.

REFERENCES

[1] Amrit Poudel, "Mobile Application Development for Android Operating System," 2013.
[2] D. Anggi et al., "ScienceDirect Mobile Mobile Financial Financial Management Management Application Application using Google Cloud Vision using Google Cloud Vision API," Procedia Comput. Sci., vol. 157, pp. 596–604, 2019.
[3] Z. Zainuddin, M. Shujahat, H. Haruna, and S. K. W. Chu, "The role of gamified e-quizzes on student learning and engagement: An interactive gamification solution for a formative assessment system," Comput. Educ., vol. 145, p. 103729, 2020
[4] P. K. Wamuyu, "Promoting savings among low income earners in Kenya through mobile money," 2016 IST-Africa Conf. IST-Africa 2016, pp. 1–11, 2016.
[5] S. C. Yang, T. L. Lee, and T. T. Feng, "User experience of mobile application's interface: Measurement development," ACM Int. Conf. Proceeding Ser., 2018.
[6] A. A. Aguileta, "Estudio de calidad y eficiencia de un enfoque de desarrollo software secuencial con programadores solos y en pareja
A study on quality and efficiency of a waterfall-like software development process applied by pair and solo programrs," vol. 27, pp. 304–318, 2019.