WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Expense Monitoring Using Mobile Application

[1]Dr. P Srinivasa Rao, [2]Sudan Neupane, [3]Shivraj Kumar Chhetri, [4]Biplav Sharma, [5]Mahim Pyakurel, [6]Sunil Kumar Singh

[1](Guide), [2](20BTRIS055), [3](20BTRIS041), [4](20BTRIS060), [5](20BTRIS048), (20BTRIS043)
Department of Information Science and Engineering,
Faculty of Engineering and Technology, Jain (Deemed to be) University, Bangalore, India
20btris055@jainuniversity.ac.in, 20btris041@jainuniversity.ac.in, 20btris060@jainuniversity.ac.in, 20btris048@jainuniversity.ac.in, 20btris043@jainuniversity.ac.in, srinuit2006@gmail.com
DOI: https://doi.org/10.5281/zenodo.10024003

ABSTRACT –

The introduction highlights the transformative impact of UPI (Unified Payments Interface) on India's digital payments landscape, emphasizing its role in driving financial inclusion, facilitating real-time transactions, and promoting economic growth. The wide adoption of UPI is evident by its soaring transaction volumes. In response to the challenges of tracking daily expenses, particularly for students, a project was undertaken to develop an application that consolidates transactions from various sources into a single thread. This application aims to provide users with a comprehensive view of their daily expenditures. The project utilizes natural language processing and pattern recognition techniques to extract relevant details from SMS messages, such as transaction amounts, dates, and merchants. This information is then organized and presented in a user-friendly format, allowing users to easily track their spending patterns and identify areas for potential savings.

The implementation of the project involved data collection, preprocessing, feature extraction, classification, visualization, and testing and refinement. The application has the potential to significantly improve financial management for individuals and businesses by providing a comprehensive view of daily expenses.

Keywords: UPI, digital payments, financial inclusion, transaction consolidation, financial management

1. Introduction

Development of UPI has led to such a big improvement in the transaction gateway of India. UPI has revolutionized the way Indians transact by making payments instant, real-time, and interoperable across various banking platforms and third-party applications. It has played a pivotal role in driving financial inclusion, empowering even the most marginalized segments of society to access financial services and build a secure financial future. UPI has also emerged as a catalyst for economic growth and development, facilitating a transition towards a transparent and accountable economy by reducing the dependence on cash transactions. The wide acceptance of UPI is evident from its staggering growth trajectory, with monthly volumes exceeding 9 billion transactions for the first time in May 2023, a 58% YoY volume growth. The volume growth of UPI peer-to-merchant (P2M) payments continues to trend significantly higher than that of peer-to-peer (P2P) payments, and for May 2023, P2M payments constituted 57% of total monthly UPI transaction volume. UPI's success is a testament to its seamless user experience, robust security features, and widespread acceptance by merchants. It has transformed the way Indians transact, ushering in a new era of digital payments and economic growth[1].

In case if you are wondering how are the popular banks of India are catching up with this technology, then. Banks in India are adopting UPI in a variety of ways, including partnering with third-party PSPs, developing their own UPI applications, and integrating UPI into their existing mobile banking applications. Some banks are also innovating in the UPI space by introducing new features and services, such as UPI-based credit card payments and UPI-based solutions for businesses. They are the core drivers of the UPI technology and the companies who are directly helping small to medium size businesses to grow and flourish.

However, every new technology has its own caveats. Even though today's modern technology made it very possible to keep track of day-to-day expenses and faster transactions. For students like us, we find it very hard to keep track of the transactions as we are just notified about the transaction at the time of the transaction and could only identify if our balance goes down a certain limit. Whenever, I personally use to spend the entire day using UPI, I always use to find myself cornered by this feeling that how much money I have left in my bank account. Furthermore, I always thought, What if there was an application that could show all of my transactions that happened within that day in a single thread, combining the transactions from different banks and different UPI's. That's where our project comes into action. We've built an application to combine all the transaction happening within a certain phone and create a thread for all of those transactions. So, that the user can see all of the money that they spend during that day or a certain day which they choose to see.

2. Literature Review

To reasonably think, there were less technological advancements made for a transaction calculator app. In retrospect, the bigger companies like Paytm, GooglePay, PhonePe and other apps provide the feature of combining the transactions that took place within the app. Now, if you take a closer look. The problem lies in the sentence itself, "Within the App". We can only see the transaction taken place in that particular application not on others. So, in order to build an application that combines all the transactions from different application and different banks, we went out to find the technologies and research papers that could help us in building the application we desire.

The very basic thing we need to do before we move on to our main program was to take permissions. However, we didn't know how to do this. So, we referred to the Android's Developer Website to learn more about Manifest[2]. 'Manifest' as the name sounds is the process through which he let the program know that we want to access some function of the phone. We searched different articles around the internet to give a practical demonstration of this feature and we landed on a tutorial of "Read SMS Messages in the Android" from Lindevs, in their website[3]. This allowed us to briefly understand the technology. We referred to Programmer.co's website where we learned on how we can actually use the 'Manifest.permission.READ_SMS' code in the JAVA to help us more to accumulate the data from the SMS.

The main Logic behind our program is separation and Combination. These both go hand in hand to help us to learn more about on how we cab separate the information we want and how can we gather and combine the information that we want. Since, we were building everything in Android Studio, it was obvious for to visit the Android, Developer's site to learn more about our basic logic which is text classification and text combination[4]. As you will learn in the coming sections of this report, that this logic of text separation and Combination is the main secret behind the Application.

3. Limitations of Current Work

The main reason for creating this application was to allow users to seamlessly view all of their transactions for a given day. The technological limitation of this technology is that no other application can currently display all of the transactions that take place on a user's phone within a single app and thread. As mentioned earlier, there are other applications such as Paytm, PhonePe, and Google Pay, but they do not allow users to view all of their phone's transactions as a whole. Instead, they only display the transactions that took place within their own app. This creates a number of problems for users who have many small transactions throughout the day and use multiple UPI and bank accounts.

We all have the misconception that our transactions actually take place through a UPI app. However, UPI apps are simply mediators between two banks[5]. When you send money to someone, the UPI app verifies the recipient's authenticity and informs your bank that the transaction is valid and secure. Your bank then initiates the transaction.

Relying on a UPI app that is only a mediator between your bank and the recipient seems misleading. We should be interested in the total amount of transactions that take place within banks. This is where UPI apps fail, because they only show transactions that took place within their own app. If you use multiple UPI apps, you can only see the transactions that took place in the app that you used for a particular transaction.

For example, if you use Paytm for a particular offer on a particular transaction, you must refer to that specific app to learn more about the transaction. Imagine different UPI apps offering different discounts and offers for your different transactions.

One of the major ways to keep track of your daily transactions is to monitor the messages you receive from your bank[6]. When you first opened an online account with your bank, they may have enrolled you in a service called SMS banking. This service sends you text messages to inform you of any transactions that have taken place in your account that day, including credits and debits. So, our approach is to remove the limitations, that are set by the UPI accounts so far.

4. Problem Definition

As regular UPI users, we often ignore the fact that our banks send us regular transactional SMS to keep track of our daily spending. This can be problematic if we use multiple banks for our transactions, as we may lose track of our overall spending. For example, I keep a large portion of my money in a separate bank account for savings, and I use a separate bank account for smaller transactions. While I have clearly defined my spending and saving habits, this distinction becomes blurred when I want to spend a little more money and transfer it from my savings account to my spending account, using a particular UPI app.

Imagine having multiple bank accounts that send you transactional SMS every time you make a transaction through a UPI app. You would then need to sift through a large number of SMS messages to track your daily spending. This is a point where people usually give up and simply use their UPI apps to find out what transactions took place during the day. However, this approach is only effective if you have a limited number of transactions, ranging from zero to 15 or 20. If you have more than 50 transactions that took place between different apps, it is very difficult to track them all. In fact, it is even harder to go through all of those transactions in the SMS section.

Although no research has been conducted on how many people check their daily transactions in India, Investopedia recently reported that a Lexington Law survey found that only 36% of Americans check their transactions daily. Another 30% check their transactions weekly, and the remainder check their transactions once their bank informs them that their balance is low[7].

Our application aims to increase the number of people who check their transactions daily by providing an easier way to view all of their transactions for a given day. We believe that this approach will create a more financially aware society and nation.

5. Importance of Expense Monitor

5.1 Easier Access of Daily Transactions.

As suggested by the Investopedia article, we need to learn more about our daily transactions in order to improve our financial stability. Since the core objective of our application is to help you keep track of your daily transactions, I believe that it can be a valuable tool for achieving financial stability.

5.2 Remainder for Your Expenses

One of the key features of our application is that it reminds you to check your daily transactions at a time of your choosing. You can even set custom notification reminders so that you can easily review any transaction that took place within a specific day or week. No more missing out on transaction details, and no more slacking off or procrastinating to learn more about the things that happened within the day with your money.

5.3 No more Dependency on a singular UPI app

Now you don't need to depend on one single UPI app. You can use multiple UPI apps for their offers and still get to know all about your transaction details within a single click of a button.

5.4 No more Dependency on a singular Bank Account

Depending upon your use cases, you can use multiple bank accounts for your transactions. You can now easily separate those types of bank accounts which you use to store heavy amounts of money and keep track of smaller transactions from different bank accounts if you wish to do so. Our application will keep your transactions safe in a single thread so that you can refer back and use the details within your transactions to make effective decisions.

5.5 Effort Saving

Imagine being able to learn more about your transactions from a day ago or a particular week. With our application, you can simply click on a calendar icon and press a button to view all of your transactions for that day or week. This is a huge time and effort saver that we are building into our application.

5.6 Minimum Maintenance

Don't worry about the constant updates, features, or maintenance required by other applications. Our standalone application uses SMS classification to calculate all of your transactions. This means that you don't need to create or maintain anything within the application to use it effectively. Since our application is only built for this small but effective approach, you can install it and start using it right away.

5.7 Mobile Integration

You don't need to go to your laptop to learn more about your transactions. You can just press a button and every transaction that happened during the day will be available on your phone. This means that it is highly mobile integrated and can be used anywhere at any time. Just install the app, allow it to access your SMS, press a button, and you are good to go.

6. Methodology

We explored various sophisticated methods for building our prototype but eventually concluded that a more straightforward approach using common sense would be more feasible and effective. While our project involved extensive research on transaction processing techniques, we discovered that simply classifying the SMS messages sent by different banks would enable us to build our system efficiently. Therefore, armed with the knowledge gained from our literature review and problem statement, we began constructing a basic outline of our system to visualize our application's goals.

Lets us learn about each of our components in detail, starting with Manifest or the Permission Granting Feature of Android Studio which basically is the building block of our Application.

In Android Studio, the manifest file is a critical component of the Android application development process. It serves as the fundamental configuration file for an Android app, providing essential information about the app to the Android system and various development tools. The manifest file plays a crucial role in defining the app's structure, components, permissions, and overall behavior. To Better understand about how Manifest works, we can use the “A Clean way to handle permissions in android article” by Fidel Montesino. This was from this article that we learned how we can guide our application in a better way to access the requiring permissions[8].

Our main goal was to build a system which is designed to be robust[9], compatible, and easy to install and remove. With these objectives in mind, we started building a basic framework within Android Studio following the flowchart provided below.

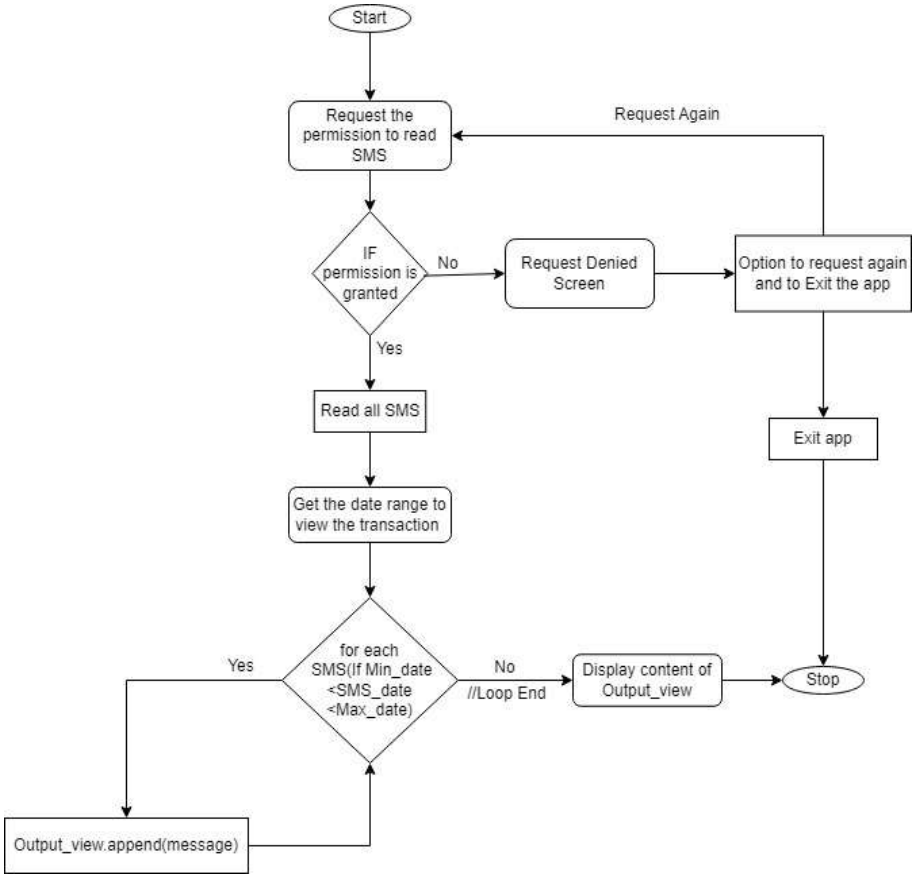

Fig 6.1: Flow chart of the Expense Monitoring Application

Here's a brief algorithm on how our application works: -

- Step 1: Start

The application's initialization is the crucial first step, as it sets the stage for all subsequent activities. This involves loading the necessary libraries, initializing data structures, and establishing connections to essential services.

- Step 2: Request Permission to Read SMS

Accessing SMS messages stored on the user's phone requires explicit permission. This step ensures that the application respects the user's privacy and only accesses data with their consent. Without permission, the application cannot proceed with its primary function of reading transaction SMS.

- Step 3: Check Permission

Verifying user permission is essential to prevent unauthorized access to SMS data. This step checks whether the user has granted permission during a previous app launch or if it's their first time using the app. If permission is granted, the application can proceed to read SMS messages; otherwise, it needs to inform the user and handle the situation accordingly.

- Step 4: Permission Granted

If the user has granted permission, the application can proceed to read all SMS messages stored on the phone. This step retrieves the SMS data that will be used to extract transaction details.

- Step 5: Permission Denied

If the user has denied permission, the application cannot function without the ability to read SMS. This step informs the user that the app requires permission to access SMS data for its primary purpose. It also provides options for requesting permission again or exiting the app if the user is not willing to grant permission.

- Step 6: Read All SMS

Reading all SMS messages allows the application to gather all relevant data for transaction extraction. This step ensures that no transaction SMS is missed, providing a comprehensive view of the user's financial activities.

- Step 7: Get Date Range

Prompting the user for a date range allows for focused transaction analysis. This step enables users to filter transactions based on their desired timeframe, making it easier to track spending patterns or identify specific transactions.

- Step 8: Extract Transactions

Extracting transaction information from SMS messages is the core function of the application. This step utilizes natural language processing and pattern recognition techniques to identify and extract relevant details from SMS messages, such as transaction amounts, dates, and merchants.

- Step 9: Display Transactions

Presenting extracted transaction details in a user-friendly format is crucial for effective communication. This step involves organizing extracted transaction data into a clear and concise format, making it easy for users to understand and analyze their financial activities.

- Step 10: Stop

The process ends with the application awaiting further user interactions or exiting as instructed. This step ensures that the application remains responsive and handles user input appropriately. It also allows the application to gracefully exit when the user is finished using it.

Let us check out some UML Diagram Implementations of our Algorithm: -

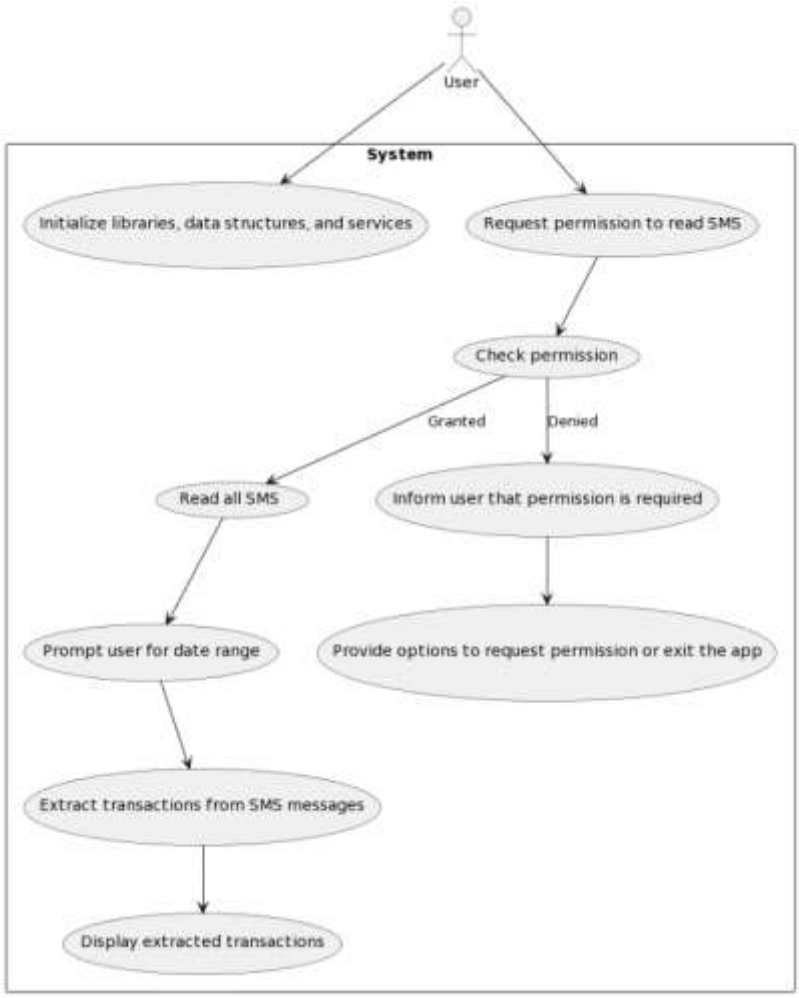

Figure 6.2: Use-Case Diagram

The provided Use Case diagram illustrates the interactions between a user and a system designed for SMS transaction processing. The "User" initiates actions, such as requesting permission to read SMS messages, and the "System" encapsulates key functionalities. The system checks and grants

permission, reads SMS messages, prompts the user for a date range, extracts transactions, and displays the results. In the event of denied permission, the system informs the user and provides options to either request permission again or exit the application. Overall, the diagram visually conveys the high-level user-system interactions and the various services offered by the system in handling SMS transactions.

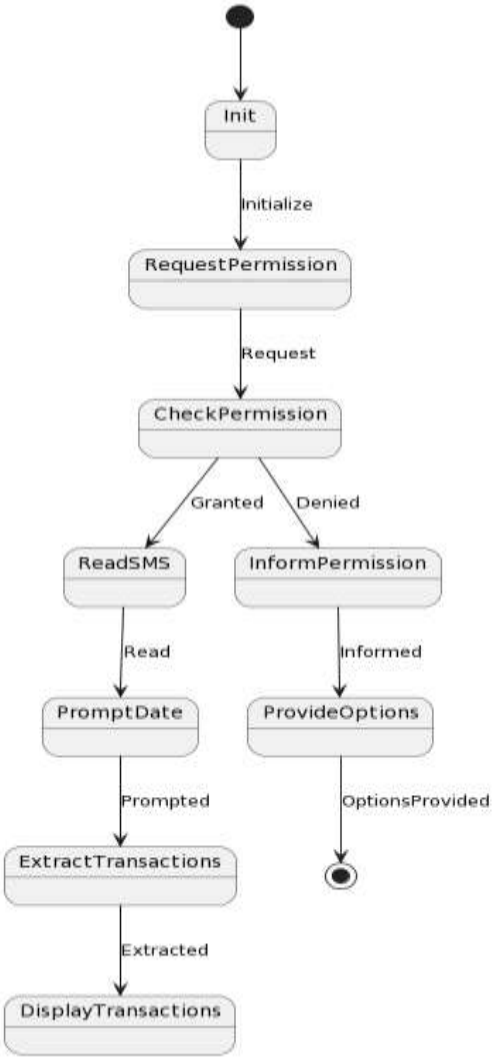

Figure 6.3: Activity Diagram

The UML diagram represents the workflow of a mobile application that extracts transactions from SMS messages. The process begins with the initialization of necessary libraries, data structures, and services. Following this, the application requests permission to read SMS messages from the user's device. If permission is granted, the application proceeds to read all SMS messages, prompts the user for a date range, extracts transactions from SMS messages within the specified date range, and displays the extracted transactions. Alternatively, if permission is denied, the application informs the user that permission is required and provides options to request permission again or exit the application. The process ends when the user has reviewed the extracted transactions or exits the application.

This explanation provides a comprehensive overview of the application's workflow, highlighting the key steps involved in extracting transactions from SMS messages. It also addresses the handling of user permission, ensuring that the application operates with the user's consent.

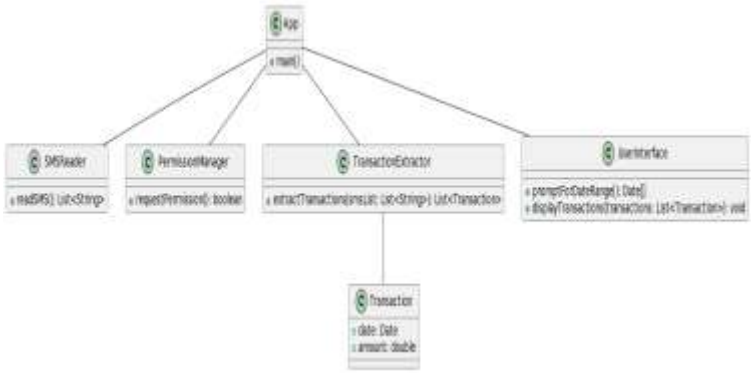

Figure 6.4: Class Diagram

The UML class diagram illustrates the overall structure and interactions between the core components of an application designed to extract and display transactions from SMS messages. The App class serves as the central orchestrator, coordinating the activities of the SMS Reader, Permission Manager, Transaction Extractor, and User Interface classes. The SMS Reader class acts as the data acquisition module, responsible for retrieving SMS messages from the user's device. It provides the App class with a structured representation of the SMS messages, enabling further processing. The Permission Manager class oversees the crucial aspect of user consent, ensuring that the application adheres to privacy guidelines. It checks for and requests permission to access SMS messages, informing the App class of the user's decision.

The Transaction Extractor class forms the heart of the transaction analysis process. Employing natural language processing and pattern recognition techniques, it meticulously extracts relevant transaction details from the SMS messages. These details include transaction amounts, dates, and merchants, which are then encapsulated within Transaction objects. The Transaction class serves as a data container, holding the extracted transaction information. It provides a structured representation of each transaction, facilitating further analysis and presentation. The User Interface class bridges the gap between the application's internal processes and the user's interaction. It receives a list of extracted transactions from the App class and presents them in a clear and organized manner, allowing the user to easily review and comprehend their financial activities.

In summary, the UML class diagram portrays a well-structured system that seamlessly integrates data acquisition, permission management, transaction extraction, and user interface elements, enabling the application to effectively extract and present transaction information from SMS messages.

7. Hardware and Software Used

Primarily, we have built our application to be tested with Android, because it is not easy for an untested application to onboard on IOS. Similarly, there were two hardware's used on the testing. Each of these hardware plays an important on building and testing. One is responsible for building and another testing. On the other hand, we are using the Android Studio as a Complete Builder for our Application and the complete Specifications of the both the hardware and the software has been listed below:

For Hardware:

Device Name	Specifications	Purpose
Acer Nitro 5	Ryzen 5 4600H 6 cores 12 Threads 16 GB RAM	Application Building for Android as Integrated Development Environment (IDE)
One Plus 9R	snapdragon 870 octacore 8GB RAM	Testing and Bug Identification

Table 7.1

For Software:

Software Name	Specifications	Purpose
Android Studio	64-Bit Program, Android Studio Giraffe | 2022.3.1 Patch 2 (September 2023)	Used as an IDE

Table 7.2

8. Previously Proposed Achievements

8.1 SOFTWARE APPLICATIONS-

Our research involved investigating the software and hardware requirements for our project. While there have been previous accomplishments in this field, they were not comprehensive enough to address the scope of our project. Through a report, we gained valuable insights into SMS reading and

writing using Android Studio[11]. Additionally, an existing GitHub developer's application named "Organizer" available on the Play Store served as an inspiration for our project. However, the developer's transaction processing and sorting methods were not adequate for modern user needs. Consequently, we had to develop our own solutions. These were the primary resources that guided our project development.

Here is our code snippet created within Android Studio:

Figure 8.1: Code Snippet

This particular code snippet displays on how read sms() function is used within our project to read the sms from the machine and to display the information when it is asked by the user. This is usually done by pressing a button within our application named Read SMS.

8.2 Designed Application

As mentioned earlier, our application is a simple version 1.0 build that currently focuses on collecting transactions from various banks within a specified date range. While features like data accumulation and date sorting are still under development, we can provide you with details or pictures of the application's interface on a mobile system.

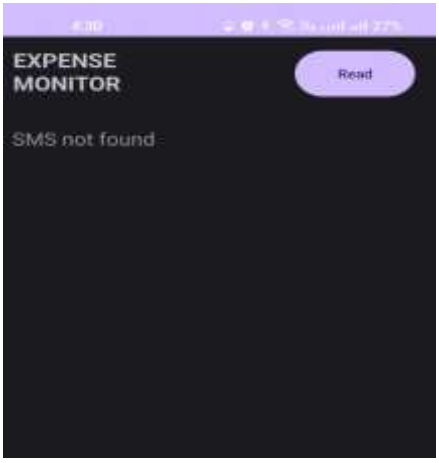

Fig 8.2.1: Application Interface before clicking Read

The first picture showcases the main interface that appears when you launch the application. Upon clicking the "Read SMS" button, the application will request permission to access your SMS inbox.

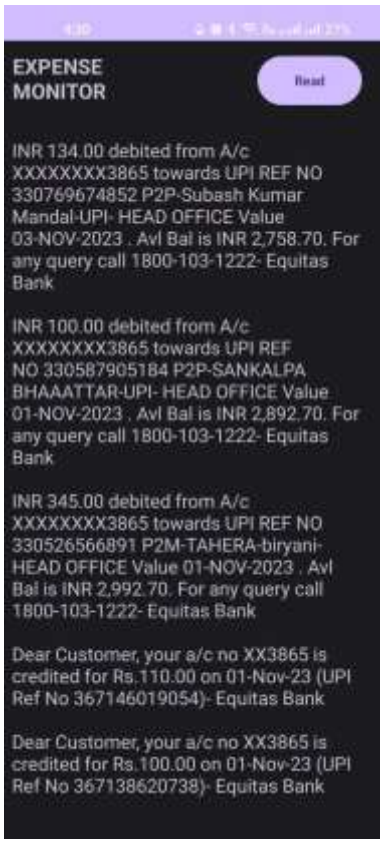

Fig 8.2.2: Application Interface showing the thread of Transactions after you click the Read button

Once permission is granted, the application will read all SMS messages and sort them based on the bank or the transaction date. The final output is a comprehensive list of transactions that have occurred through various banks and UPI apps.

9. Conclusion

Our project aims to empower our users to monitor and manage their daily transactions. We want to clarify that UPI is not the only way to track transactions, and that their bank provides the main record of their financial activities. Our application is designed to be user-friendly and cost-effective, requiring only a simple click of a button to access its features.

By using classification algorithms and sorting mechanisms, we were able to sort over 90% of messages that contain transaction details. In the future, we plan to add more features, such as date sorting, to help our users organize their transaction history more efficiently

References

[1] Prasad, H. (2023, July 9). The UPI Effect: How UPI is reshaping India's payment industry. Inc42 Media. https://inc42.com/resources/the-upi-effect-how-upi-is-reshaping-indias-payment-industry/

[2] App manifest overview. (n.d.). Android Developers. https://developer.android.com/guide/topics/manifest/manifestintro#:~:text=The%20manifest%20file%20describes%20essential,operating%20system%2C%20and%20Google%20Play.

[3] Read SMS messages in Android. (2023, January 19). Lindevs. Https://lindevs.com/read-sms-messages-in-android

[4] Text Classification. (2022). developer.android.com. https://developer.android.com/reference/android/view/textclassifier/TextClassification

[5] What is UPI (Unified Payments Interface) and How it Works? (2023, September 23). Razorpay Blog. https://razorpay.com/blog/what-is-upi-and-how-it-works/

[6] Adagunodo, E. R., Awodele, O., & Ajayi, O. B. (2007). SMS Banking Services: a 21st century innovation in banking technology. Issues in Informing Science and Information Technology, 4, 227–234. https://doi.org/10.28945/945

[7] Lake, R. (2021, September 30). How often should you monitor your checking account? Investopedia. https://www.investopedia.com/how-often-should-you-monitor-your-checking-account-4798537

[8] Montesino, F. (2021, December 24). A clean way to deal with permissions in Android | by Fidel Montesino | Kin + Carta Created. Medium. https://medium.com/kinandcartacreated/finally-a-clean-way-to-deal-with-permissions-in-android-539786a7846

[9] Adel, A. (2021, February 1). Building robust Android Apps — Step by step - EGDroid - Medium. Medium. https://medium.com/egdroid/tips-and-tricks-for-building-a-robust-android-app-a76bcad55eda

[10] Android Studio Giraffe | 2022.3.1. (n.d.). Android Developers. https://developer.android.com/studio/releases

[11] Pise, S., Moundekar, R., Meshram, R., Mohadikar, S., Durugwar, R., & Banabakode, D. (2019). Implementation of Handling Android Application using SMS. International Research Journal of Engineering and Technology (IRJET), 06(03), 2395–0056. https://www.irjet.net/archives/V6/i3/IRJET-V6I3360.pdf