Credit Card AR Viewer using Android SDK A newgen banking experience

Guruprasad Kulkarni[1], Manish Sharanappa[2]
[1]*Assistant Professor, Computer Science and Engineering, Guru Nanak Dev Engineering College Bidar-585401, Karnataka*
[2]*PG Student, Computer Science and Engineering, Guru Nanak Dev Engineering College Bidar-585401, Karnataka*

***Abstract*— In this growing world of information technology, it is very important for the innovative ideas to roll especially in the field on banking sector. The banking world has seen a huge revolution between pre and post digital banking. Our project idea aims at creating one of the baking features which might help the users to move further to experience an augmented way of banking. This idea by which a user scans the credit card or debit card after which the recent transactions linked to that savings account is fetched and displayed through an augmented reality effect by drawing the tables and charts using android canvas built in functions. The use of artificial intelligence resources to extract the card number from the card number through the android packages can also be seen. The Internet creates the stage for sustained commitment to develop applications which help the people all over the world. This leads to a greater commitment to this research and development of IT infrastructure.**

***Keywords*—Credit card; Augmented Reality; OpenGL; Android Canvas; AR effect; Finance Technology; Banking; Innovation**

I. INTRODUCTION

The Main objective of the project idea is to create an android app which scans the credit or debit card and fetched the transactions history with the data like transaction type, date, amount, description from the bank service providers which is currently mocked and display the same in the new way i.e., by creating an augmented environment by drawing the necessary pie charts and table above the card. The usage of android SDK tools and machine learning packages help us in achieve the project idea.

The revolution of the world of information through the means of internet has been more than ever before. The internet has brought soul to the IT world. This has set the global possibility for the unparallel level of capability integration. The information can be accessed from any geographic location. In the recent decade there has been a phenomenal improvement in the Banking world. From the conventional way of banking to the modern-day digital banking, there has been a huge evolution in the Finance technology world.

At the same time new technologies gave birth to various sectors. One of them being Augmented reality also popularly known as AR. With this setup, the main agenda for the project is to use this concept of augmentation or virtualization to display the balance and the recent transactions for a savings account. This results in an evolution of FinTech world. The user gets connected with this augmented output through canvas drawings.

This project targets native android mobile and android users through the use cases of android native SDK but can also be made to suit the Web users through its Responsive UI and improved UX design. In the world of Artificial intelligence, we do make use of the Machine Learning android packages to scan the credit or debit card number and then the result from the models is later used for fetching the transaction history and other required information. This revolutionary conceptual idea can help the users to enjoy and feel the banking through the concept the virtualization.

II. REQUIREMENT ANALYSIS

A. Software Requirement Specification

The conventional way of representation was to get the account statement on the passbooks by visiting the banks then to the modern world Mobile or Internet Banking statements, we target a level up revolutionary idea of representing the same on the augmented format.

The usage of core android libraries like OpenGL, Canvas and ML Kit can be seen to accomplish the said requirement. The more about these can be discussed in the further chapters. Under this section, an SRS document containing the details like the purpose of the application, functionality of the application is described. This consists of the scope, purpose, functional and nonfunctional requirements, software and hardware requirements of the project.

B. Purpose of the Project

The main agenda of this idea is to build a credit card or debit card transactions scanner application which is focused on displaying the transactions history and the account balance by creating an AR effect using the Canvas class and OpenGL core library. This creates the essence of banking in a more augmented way. From the early offline banking to the digital banking era, the Information technology world has come a long way and its contribution to the Banking sector is enormous.

C. Intended Audience

The Primary audience or the consumers of this application are the Debit Card or the Credit card holders who are willing to fetch and check their recent transactions carried out in a savings accounts.

D. Application Functionality

The following are the major functionalities prescribed for the project idea.

- Live Card Scanning and Number extraction using Android ML Kit
- Display the Transaction History and Account Balance though Canvas drawings.
- Spends analyzer through the categorical pie-chart representation.

E. Application Features

The following are the application features which are mainly focused for the project idea.

- Portable: The proposed application can be consumed at any time on any environment irrespective of system software or hardware specification.
- Ready to Use: No extra configuration, plugin or installation is required.
- User friendly: User can easily operate the application as it is developed under the user experience consideration as the primary intent.
- Native-Functionality: The proposed application only uses native resources and do live reload.
- Performance: User can extract the card transaction details at faster response time.

F. System requirements

- Software Requirements:

1. Operating System: Android Operating System
2. ML Libraries: Google ML Kit
3. Android Studio
4. Nginx Server
5.Buildin libraries

- Hardware Requirements:

1. Processor with min. 1.2GHz Clock Speed
2. Min 512 MB RAM
3. 8GB Storage Space
4. Camera with Min 12 MP

III. SYSTEM DESIGN

The architecture for the proposed system requires a highly performing mobile device with a minimum of 1.4 GHz clock speed. This device runs on the Linux based Android operating system which gives the facility to establish the connection with the hardware resources. The greater the performance of the CPU faster the processing speed. The design is majorly divided into following layers:

1. Camera Preview
2. ML Kit Preprocessor
3. Pattern Matching
4. API Layer
5. Canvas Drawings.

The Layers of our system architecture is as shown in Fig. 1

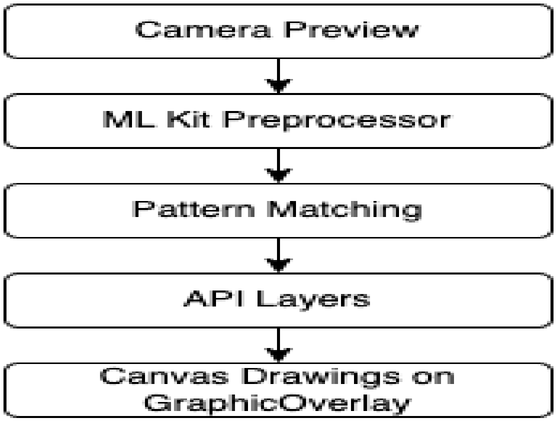

Fig 1: System architecture

A. Camera Preview

This is the first layer of the application architecture where the camera service is started, and the camera resource starts to capture the stream. This stream is then transferred to the next layers for the further preprocessing of data. After the complete processing of data, the output is also rendered on this preview screen.

B. ML Kit Preprocessor Layer

This The stream received from the camera preview is now processed for the extraction of the text. This ML Kit is provided by the google platform for the extraction of text from the image. Once the text is extracted from the image the same is sent for the pattern matching. The flow is as shown in Fig. 2.

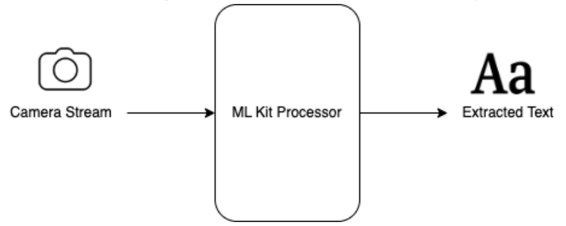

Fig. 2: ML Kit data flow

C. Pattern matching

The consumer of the preprocessor is the pattern matching unit where the text extracted from the image is matched for the credit or debit card of various providers format. If the match is found with all the described rules, then the next tasks are initiated.

D. API Layer

The credit card no is used to make an API call to fetch the transaction data like transaction date, transaction type, amount, description etc. based on the criteria the data is fetched from the backend server of the bank service providers. This layer can be mocked for the testing purpose.

E. Canvas Drawings:

Once the transactions data is received the table, pie chart is drawn on the camera preview screen using the Canvas class. This creates an 2D augmented effect for the user to experience.

IV. DATA FLOW

The flow of data through the multiple layers is as discussed below:

1. User opens the android application on the supported device.
2. The device then checks for Camera permission. If not granted the app is exited else, it is processed to the next steps.
3. The camera preview then starts streaming the camera content, the user has to then scan the debit or credit card.
4. The stream is then sent to the ML Kit preprocessor for extraction of text.
5. The data preprocessed is then matched with the credit card number pattern using pattern matching methods.
6. If the matched text is found to be a credit card or debit card number, then a API call is made to the bank servers to get the recent transactions for the savings account linked with.
7. The canvas drawings are drawn like pie chart, table etc.

The flowchart for the system can be seen in the Fig. 3

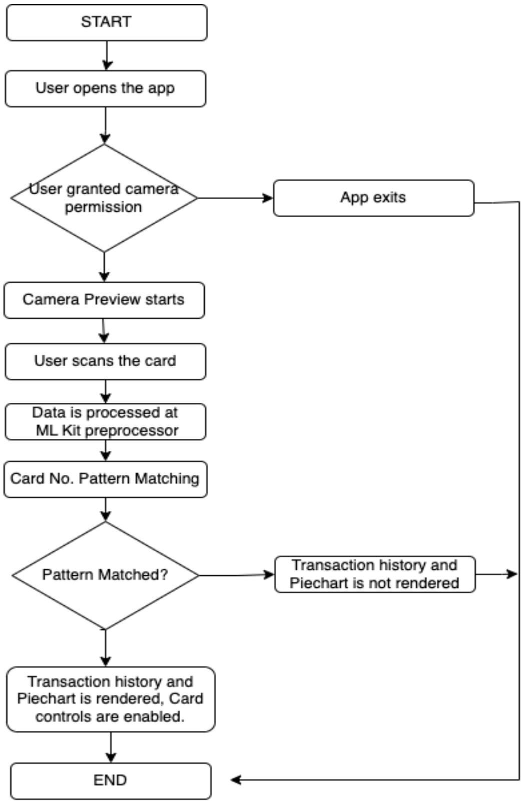
START
User opens the app
User granted camera permission
App exits
Camera Preview starts
User scans the card
Data is processed at ML Kit preprocessor
Card No. Pattern Matching
Pattern Matched?
Transaction history and Piechart is not rendered
Transaction history and Piechart is rendered, Card controls are enabled.
END

Fig 3 Data Flow Chart

V. USER INTERFACE FLOW

The user interface here is referred to the android layouts which is represented by the .XML files. These layouts are required to render the canvas elements. The User interfaces are created majorly for the elements mentioned below:

1. Transactions History Layout
2. Pie-chart Layout
3. Card Controls Layout
4. Camera Preview Layout

The main interface consists of the camera preview space where the camera stream is presented to the user. The flow of data between layouts is as shown in Fig 4

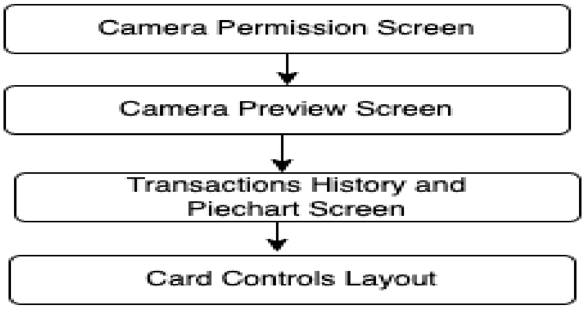

Fig 4: Layout transitions

Once the camera preview layout is loaded, The hardware resources are consumed, and it waits for the user to scan credit or debit card. Camera Preview Layout is the preview layout to stream the camera. The user gets the camera preview from where the card can be scanned. The Graphic overlay is used to render the transaction history layout and the pie-chart with legends layout.

Pie-chart Layout consists of the xml tags under the base of which the pie-chart is drawn using android canvas class. The pie chart helps the user in understanding the spends by category. Pie-chart is drawn with a central angle of 360 degrees. The score is calculated by the summing up all the categories score and the degree for a pie in the chart is drawn. The Graphic Overlay is the component base which establishes the environment. The add On Layout Change Listener method listens to the canvas methods which is triggered whenever there is a change or update in the drawings

VI. TESTING

The application needs to be tested at multiple level of development and post development. Unit level of testing should make sure that each android component is rendered properly on successive flow of events like onClick, onView, onScroll etc. It's the developer's responsibility to check each component with check the props sent and the state initialized at the unit level. The one major testing being the user Interface testing. In this we will test whether the layouts designed for the application is rendered as expected or not. The layouts are designed using xml files.

A. Unit and Integration Testing

The API call at the backend is the key functionality of the backend server. The data is supposed to be received from the bank service providers. The data from the API call is mocked for the time being of the project implementation. The working of backend is necessary for the program execution. It should be tested by with various possible scenarios. The transaction history layout and pie-chart layout will have different working scenarios.

a. The transactions history and pie-chart should be rendered if the card scanned is correct and is linked with a savings account. This is the happy path scenario for the project.
b. The transactions history and pie-chart should not be rendered if the card scanned is incorrect or is not linked with any savings account.
c. If there are no transactions for a savings account, then show no recent transactions message.
d. Performance testing should be done to check the integrity and the capacity to handle the multiple API calls.
e. The server should return 400 if any request param is missing in the request body.
f. The server should return null in the standard input if the server is down.

VII. INTERPRETATION OF RESULTS

Headings As reported in the earlier chapters, The design and implementation of the proposed application. The process from requirement analysis to the testing part, we conclude here to the result part. The output can be discussed under the test environment along with sample outputs with the help of screenshots captured.

Recent Transactions Table consists of the transaction data which is received from the bank service providers. For the project implementation this API response is mocked. These transaction data include transaction date, transaction type, transaction amount, Transaction description. The application developed to create the advanced form of transactions viewing on the android app is as shown in Fig 5.

VIII. CONCLUSION

As shown in the research study, this application can be a very essential tool for the bank customers who can check their recent activity, balance and more details like card controls on the go with a more realistic approach. This is achieved by creating a augmented effect by drawing the transactions history and pie chart where the details on recent activity is displayed above the scan card. This application is a revolution benchmark set in the field of banking and finance technology world. This can also be a standard example to inspire the researchers or learners who aspire to develop more augmented or virtual apps in the future days.

A. Future enhancements

The following are the some of the enhancements that can be made which can bring some more features to the application

- Currently the view is available for the English language. More regional and international languages can be implemented to let everyone do banking in their favorite and comfortable language.
- Currently the scanning of card takes 7-8 secs to extract the card number which can be tweaked to get a more improvised and efficient time scale.
- This app is limited to very few card issuers as we restrict certain patterns as valid.
- Add support for more themes so that the user can have a good feel on the app environment.

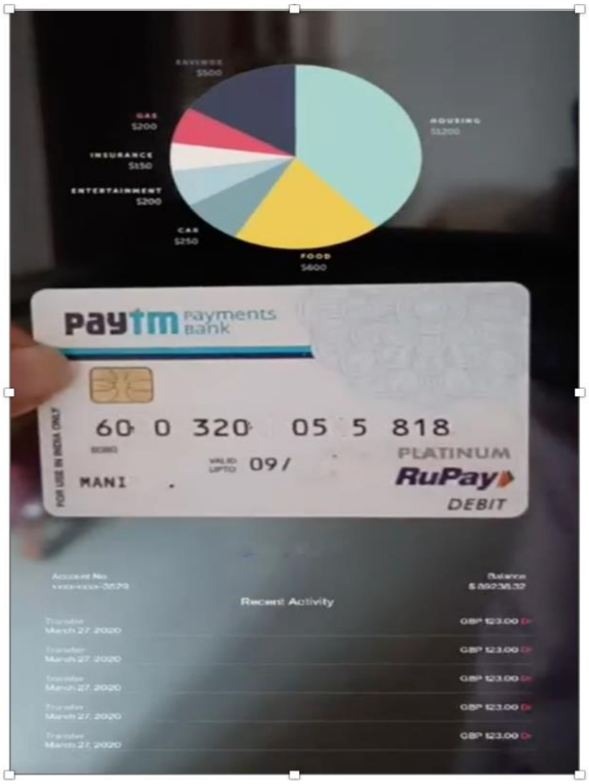

Fig 5. Complete output preview

ACKNOWLEDGMENT

The satisfaction and euphoria that accompany the successful completion of any work would be incomplete without mentioning the people who have made it possible, I would like to thank Dr. Dayanand J, for giving us the guidance and encouragement to complete our project successfully.

It gives us immense pleasure in expressing our heartfelt gratitude to Dr. Dhananjay Maktedar for boosting our confidence and standing with us in our tough times.

REFERENCES

[1] Augmented Reality: An Emerging Paradigm by Hemant Sharma

[2] OCR to Read Credit/Debit Card Details to Autofill Forms on Payment Portals by MIT College of Engineering, Pune

[3] Credit Card Transaction Based on Face Recognition Technology by SRM College of Engineering, Chennai

[4] Augmented Reality and Virtual Reality in banking industry Research
Paper By Shree Ram

[5] Online Credit card Transaction using Fingerprint Recognition bu Uma Maheshwari

[6] Banking in Metaverse – Opportunities and Challenges by Raghavendra Prasad

[7] Advances in Banking technology by Vadlamani Ravi

[8] Growth and Emrging Fintech Prospects by Jai Sharma

[9] Banking Governance for Innovative Ideas by Rajesh Krishna

[10] Innovative Strategies or Implementing FinTech in Banking by Hari Gopal Das